Classifying Mushroom Edibility

New Mexico

Supercomputing Challenge

Final Report

April 6, 2022

Team 30

La Cueva High School

Team Members

Yana Outkin

Teacher

Yolando Lozano

Classifying Mushroom Edibility

Mushroom hunting is a beloved hobby, especially in Europe. Moreover, there are numerous health benefits due to the antioxidants found in certain mushrooms. But despite the existence of numerous datasets, many apps still misdiagnose common poisonous mushrooms as safe. This problem has negatively impacted the prospects of mushroom-hunting, where hikers are discouraged from picking mushrooms that are unfamiliar to them.

This project uses two datasets, a characteristic dataset from the University of California-Irvine, and an image dataset from WildFoodUK, to computationally predict the edibility of a mushroom given an image. My goal is to provide a framework for utilizing and connecting multiple datasets, in hopes of increasing the accuracy in identifying edibility.

Executive Summary

In hopes of increasing mushroom-hunting prospects, several mobile apps have been designed to identify unknown mushrooms, even going as far as to plot the exact location of where the mushroom was found. But despite the existence of numerous datasets, many apps still misdiagnose common poisonous mushrooms as safe (Shields, 2017). This problem has negatively impacted the prospects of mushroom-hunting, where hikers are further discouraged from picking mushrooms that are unfamiliar to them.

The main reason for the error is that many studies fail to utilize multiple datasets in their research, focusing mainly on a dataset made purely of images or purely of characteristics. For example, with an accuracy of 55%, the following research, Deep Shrooms, utilized a dataset consisting purely of images from Mushroom World (an online database) (Koivisto et al., 2017). Similarly, a study analyzing the speed of the Naive Bayes and Decision Tree algorithms used only the characteristic dataset (Al-Mejibli I., and Abd D. 2017).

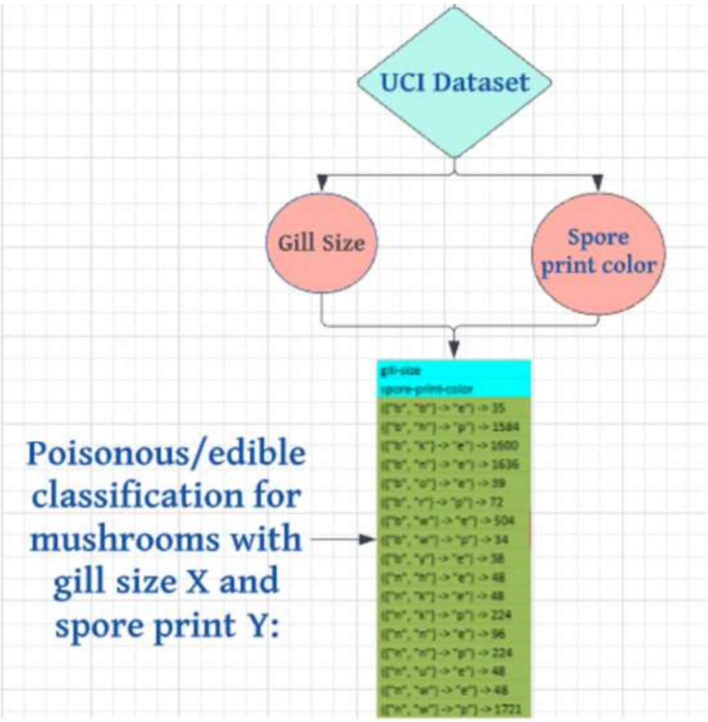
My research will focus on optimizing the correlation between multiple datasets to increase the accuracy of computationally identifying the edibility of a mushroom. Furthermore, to make this project more user-friendly, I wanted to have the user input an image, rather than a long list of characteristics, as the input data. This organization will make this project readily available to hikers with little knowledge of specific mushroom characteristics (such as veil, ring type, etc) and to hikers who are currently in the mountains. The ease with which mushroom hunters can upload an image to get the classification will save the time in determining edibility in comparison to using a field guide, be more accurate if the field guides have only a couple of sparse images documenting the mushroom, increase the amount of hikers, and will also help with maintaining forest biodiversity when fewer inedible mushrooms are picked.

For my project, I used the Wolfram Language, a flexible language with in-built machine learning functions. I utilized the machine learning functions to extract relevant features from mushroom images, after I created the training datasets. With this programming language, rather than focusing on syntax and specific algorithm structure as I would have needed to do with Java or Python, I was able to focus more on data analysis and finding trends in the datasets.

I used two main datasets. The dataset from the University of California Irvine (UCI) documents specific mushroom characteristics and the corresponding edibility of the Agaricus and Lepiota mushroom families. The Agaricaceae mushroom family consists of the most commonly known mushrooms, found mainly across Northern Europe. However, these mushrooms are also found worldwide on a variety of terrains—ranging from dense forests to prairies. They are well-known decomposers, decomposing mulch, leaf litter, and wood. There are also documented online image databases and field guides (Koivisto et al, 2017).

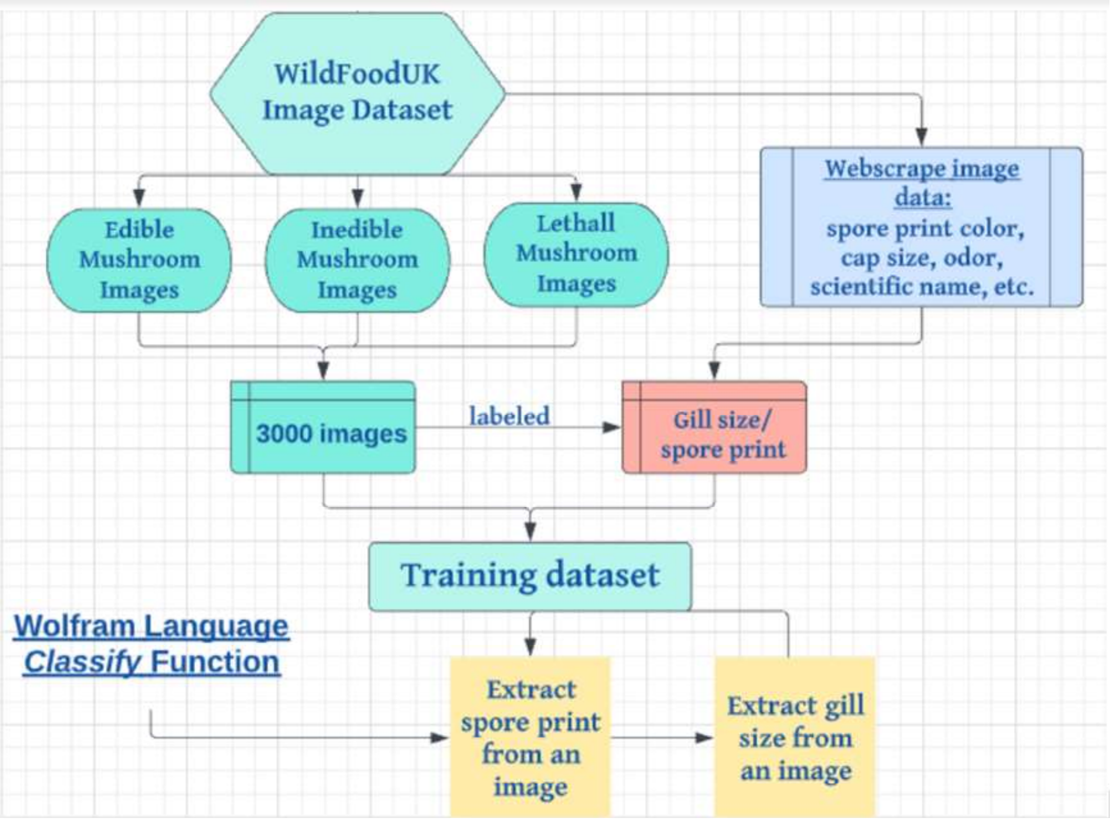
The goal of my project is to take a picture of a mushroom, extract the significant characteristics, and predict the edibility from those characteristics.

My first step is finding the significant characteristics that predict edibility, using the UCI dataset.


UCI Dataset
Gill Size
Spore print color
Poisonous/edible classification for mushrooms with gill size X and spore print Y:

```
gill-size
spore-print-color
({"b", "b"} -> "e") -> 35
({"b", "h"} -> "p") -> 1584
({"b", "k"} -> "e") -> 1600
({"b", "n"} -> "e") -> 1636
({"b", "o"} -> "e") -> 39
({"b", "r"} -> "p") -> 72
({"b", "w"} -> "e") -> 504
({"b", "w"} -> "p") -> 34
({"b", "y"} -> "e") -> 38
({"n", "h"} -> "e") -> 48
({"n", "k"} -> "e") -> 48
({"n", "k"} -> "p") -> 224
({"n", "n"} -> "e") -> 96
({"n", "n"} -> "p") -> 224
({"n", "u"} -> "e") -> 48
({"n", "w"} -> "e") -> 48
({"n", "w"} -> "p") -> 1721
```

Afterwards, I created training images datasets that extract those relevant characteristics from an image, using the Wolfram Language’s *Classify* function.



After I created the *Classify* functions to extract the relevant characteristics from an image, I used the UCI dataset to predict edibility.

Enjoy! My paper documents the specific steps I took to link the datasets and then analyze the results. I learned a lot about working with large amounts of data, as well as the limitations of connecting multiple datasets.

Datasets

UCI Dataset

The University of California Irvine dataset is a multivariate, categorical characteristic dataset (Dua and Graph 2019). It contains approximately 8000 mushroom entries, each with 22 characteristics (cap color, gill size, odor, habitat, etc) and an edibility classification (“e” or “p”). Each characteristic contains different attributes, such as cap color “white”, “brown”, “orange”, etc.

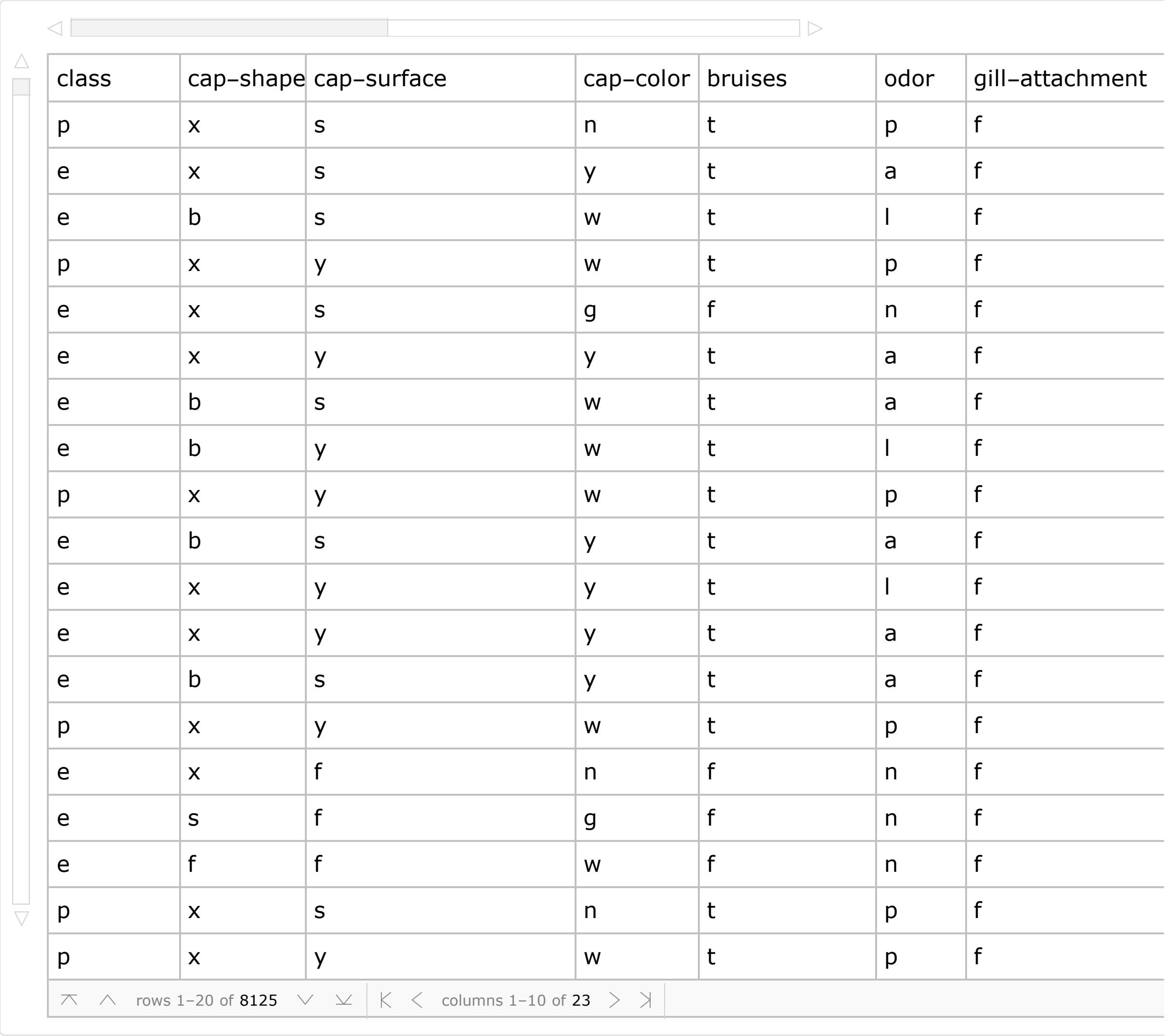
https://archive.ics.uci.edu/ml/datasets/Mushroom

```
mushroomCharacteristics = Import["cvs file"];
```

(Debug) In[ ]:=

```
Dataset@mushroomCharacteristics
```

(Debug) Out[ ]=



class	cap–shape	cap–surface	cap–color	bruises	odor	gill–attachment
p	x	s	n	t	p	f
e	x	s	y	t	a	f
e	b	s	w	t	l	f
p	x	y	w	t	p	f
e	x	s	g	f	n	f
e	x	y	y	t	a	f
e	b	s	w	t	a	f
e	b	y	w	t	l	f
p	x	y	w	t	p	f
e	b	s	y	t	a	f
e	x	y	y	t	l	f
e	x	y	y	t	a	f
e	b	s	y	t	a	f
p	x	y	w	t	p	f
e	x	f	n	f	n	f
e	s	f	g	f	n	f
e	f	f	w	f	n	f
p	x	s	n	t	p	f
p	x	y	w	t	p	f

rows 1–20 of 8125 columns 1–10 of 23

Webscraping: Image Dataset

WildFoodUK contains an extensive database with mushrooms found in the UK, sorted by edible, inedible, and lethally poisonous mushrooms. Each mushroom contains consistent descriptions of aspects such as cap width, cap color, odor, and spore-print-color.

Originally, I was planning on using Mushroom World to create my image dataset because it contains an extensive community-uploaded database with multiple images for each mushroom, along with explanations regarding notable characteristics.

However, I found an even better website. WildFoodUK contains a more extensive database with mushrooms found in the UK, sorted by edible, inedible, and lethally poisonous mushrooms. In addition, in contrast to Mushroom World, each mushroom contains consistent descriptions of aspects such as cap width, cap color, odor, and spore-print-color. This consistency in the organization of the information makes it great for my computational project. ie, after I extract the images and characteristics for 1 mushroom/website, because of the consistency of the layout, I can easily extract the same information from the rest of the mushroom websites. Webscraping is the process of computationally extracting data from the web.

There are 98 entries for edible mushrooms, 33 inedible entries, and 35 lethally inedible mushrooms. For each entry, I webscraped 10 - 15 images, by going to the individual website documenting that specific mushroom.

An obvious limitation with this dataset, however, is that the mushrooms do not all belong to the Agaricus and Lepiota family, as is documented by the UCI. Therefore, there may be some discrepancies in classifying edibility for mushrooms not in the Agaricus or Lepiota family.

Getting the Hyperlinks

Importing the hyperlinks to all the mushrooms listed on the “edible” section of WildFoodUK:

```
(Debug) In[◦]:=
hyperlinksEdible = Import[
   "https://www.wildfooduk.com/mushroom-guide/?mushroom_type=edible", "Hyperlinks"];
```

Each mushroom has 3 hyperlinks, removing duplicates:

```
(Debug) In[◦]:=
goodhyperlinksEdible = First /@ Partition[hyperlinksEdible[[39 ;; -14]], 3];
```

Double checking that there are indeed 98 entries:

```
(Debug) In[◦]:= Length@goodhyperlinksEdible
(Debug) Out[◦]= 98
```

And that each hyperlink is distinct:

```
(Debug) In[◦]:= CountDistinct@goodhyperlinksEdible
(Debug) Out[◦]= 98
```

Inedible hyperlinks:

```
(Debug) In[◦]:=
hyperlinksPoisonous =
  Import["https://www.wildfooduk.com/mushroom-guide/?mushroom_type=inedible",
   "Hyperlinks"];
```

```
(Debug) In[◦]:=
goodhyperlinksPoisonous = First /@ Partition[(hyperlinksPoisonous[[39 ;; -14]]), 3];
```

Lethally poisonous hyperlinks:

```
(Debug) In[ ]:= hyperlinksLethallyPoisonous =
                  Import["https://www.wildfooduk.com/mushroom-guide/?mushroom_type=poisonous",
                   "Hyperlinks"];
```

```
(Debug) In[ ]:= goodhyperlinksLethallyPoisonous =
                  First /@ Partition[(hyperlinksLethallyPoisonous[[39 ;; -14]]), 3];
```

Importing images from each of the hyperlinks:

```
edibleMushroomImages =
  Table[(Import[goodhyperlinksEdible[[n]], "Images"])[[10 ;; -10]],
   {n, Length@goodhyperlinksEdible}];
```

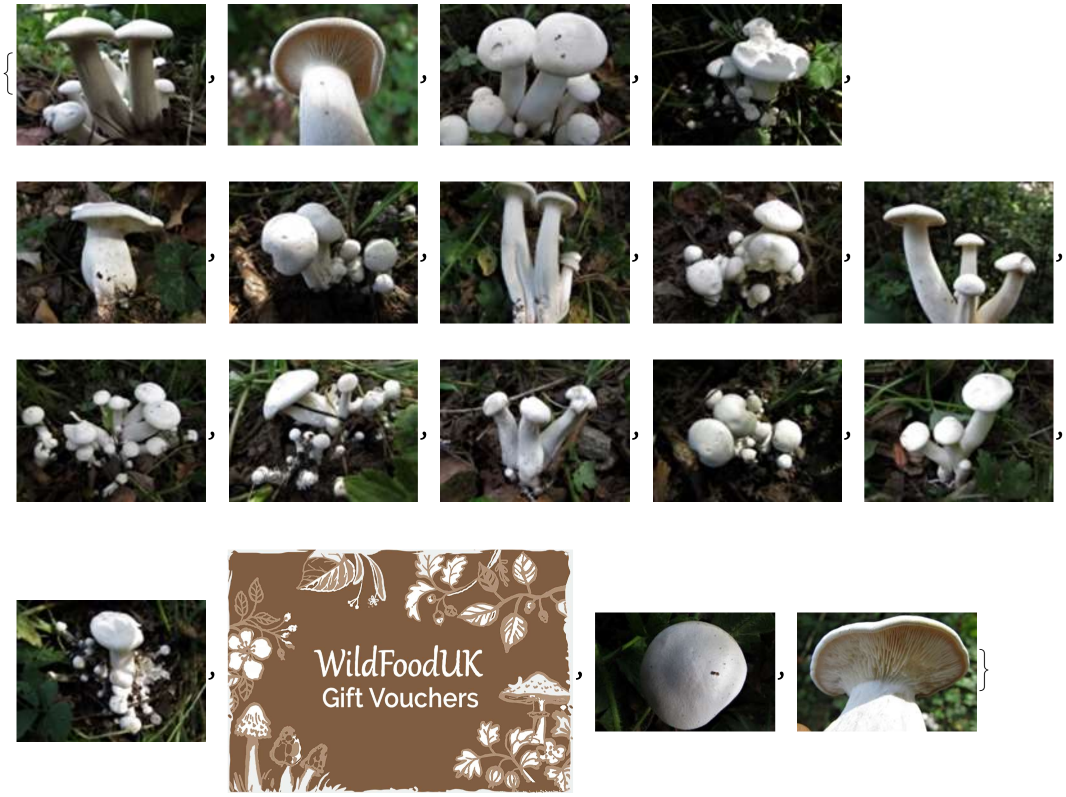
```
poisonousMushroomImages =
  Table[(Import[goodhyperlinksPoisonous[[n]], "Images"])[[10 ;; -10]],
   {n, Length@goodhyperlinksPoisonous}];
```

```
lethallyPoisonousMushroomImages =
  Table[(Import[goodhyperlinksLethallyPoisonous[[n]], "Images"])[[10 ;; -10]],
   {n, Length@goodhyperlinksLethallyPoisonous}];
```

Example images from lethally poisonous mushroom #26 - White Domecap:

```
(Debug) In[ ]:= lethallyPoisonousMushrooms[[26]]
```

(Debug) Out[ ]=



In total, there were approximately 4000 training images!

However, as you can see, the training images also contain images other than mushrooms webscraped from the website, such as vouchers, user profile pictures, and mushroom season categories (such as a sun for summer, and a leaf for spring). I was not sure how to filter the images so that there would only be images of mushrooms, instead, I manually deleted the 1000 unneeded images (took about 40 minutes).

The resulting image dataset contains approximately 3000 images.

Random Images - Validating Results

To later test the accuracy, I also compiled a list of random images, 2 for each distinct mushrooms, resulting in 196 random edible mushroom images, and 66 inedible, and 70 lethally poisonous mushroom images.

```
(Debug) In[ ]:= randomEdible = Flatten[#[[3 ;; 4]] & /@ edibleMushroomsBetter];
```

```
(Debug) In[ ]:= randomInedible = Flatten[#[[3 ;; 4]] & /@ inedibleMushroomsBetter];
```

```
(Debug) In[ ]:= randomLethal = Flatten[#[[3 ;; 4]] & /@ lethallyPoisonousMushrooms];
```

```
(Debug) In[ ]:= Length /@ {randomEdible, randomInedible, randomLethal}

                {196, 66, 70}
```

Identifying Significant Characteristics

Which characteristics, based of the UCI dataset, are good predictors of edible/poisonous mushrooms?

To determine the relevant characteristics, I analyzed the difference between the amount of poisonous and edible counts. For example, if 80% of entries for cap-color blue are poisonous, then cap-color blue is a relatively high indicator. On the other hand, if 50% of entries for cap-color red are poisonous, then this characteristic gives no valuable information whatsoever, because cap-color red is equally likely to be poisonous as edible.

Single Variable

I first looked at individual characteristics, and the amount of times those mushrooms were edible or poisonous.

Correlation of cap shape v. edibility

```
(Debug) In[ ]:= capShape =
                 Table[mushroomCharacteristics[[n, 2]] → mushroomCharacteristics[[n, 1]], {n, 8000}][[
                  2 ;; -1]]

(Debug) Out[ ]= {x → p, x → e, b → e, x → p, x → e, ⋯ 7989 ⋯ , x → e, k → p, f → e, k → p, k → p}
                large output    show less    show more    show all    set size limit...
```

Counting the amount of times each attribute of cap shape, “x”, “b”, “s”, “f”, “b”, “k”, and “c”, is poisonous or edible:

```
(Debug) In[ ]:= countsCapShape = Counts[capShape]

(Debug) Out[ ]= <| (x → p) → 1703, (x → e) → 1931, (b → e) → 387, (s → e) → 32, (f → e) → 1585,
                 (f → p) → 1552, (b → p) → 48, (k → e) → 205, (k → p) → 552, (c → p) → 4 |>
```

By sorting keys, easier to understand datasets and bar graphs

```
(Debug) In[ ]:= KeySort[countsCapShape]

(Debug) Out[ ]= <| (b → e) → 387, (b → p) → 48, (c → p) → 4, (f → e) → 1585, (f → p) → 1552,
                 (k → e) → 205, (k → p) → 552, (s → e) → 32, (x → e) → 1931, (x → p) → 1703 |>
```

```
(Debug) In[ ]:= Dataset@KeySort[countsCapShape]
```

(Debug) Out[ ]=

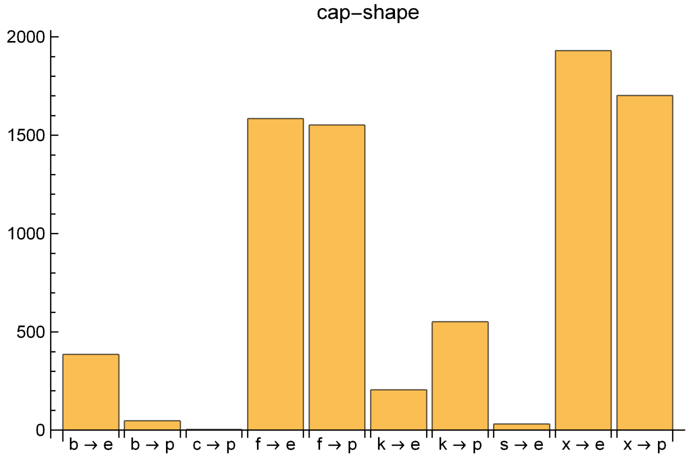
"b" → "e"	387
"b" → "p"	48
"c" → "p"	4
"f" → "e"	1585
"f" → "p"	1552
"k" → "e"	205
"k" → "p"	552
"s" → "e"	32
"x" → "e"	1931
"x" → "p"	1703

Looking at this dataset, one can see that a cap shape of “f”, has an equal amount of poisonous and edible entries associated with it (1585 and 1552 respectively), signaling that cap shape “f” is not a good predictor. This is similar with cap shape “x” (1931 and 1703 respectively for edible and poisonous counts). Overall, cap shape is not a good predictor on its own.

```
(Debug) In[ ]:= BarChart[KeySort[countsCapShape], ChartLabels → Keys[KeySort[countsCapShape]],
                 PlotLabel → mushroomCharacteristics[[1, 2]]]
```

Counts of poisonous/edibility of cap shape in a bar chart:

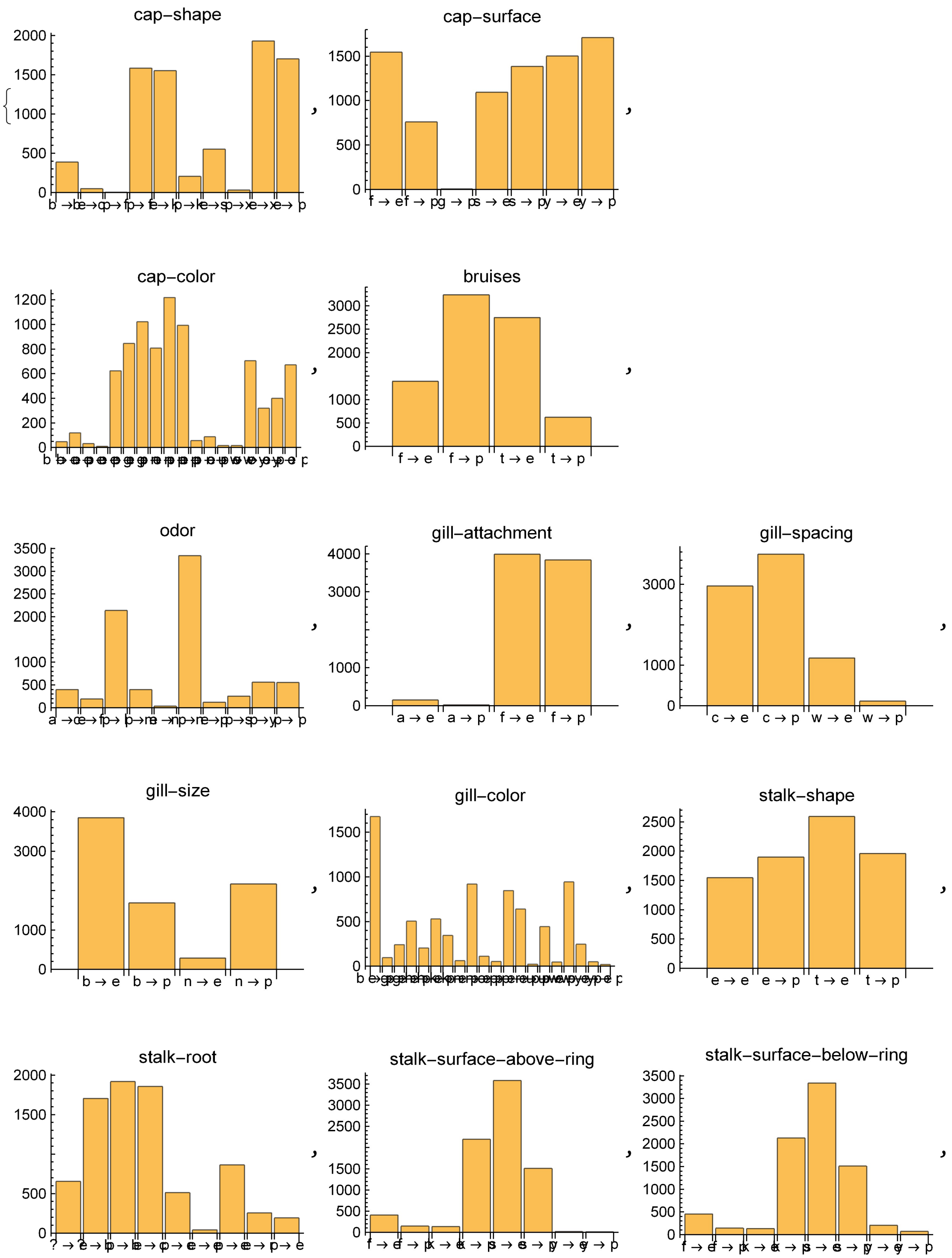
(Debug) Out[ ]=

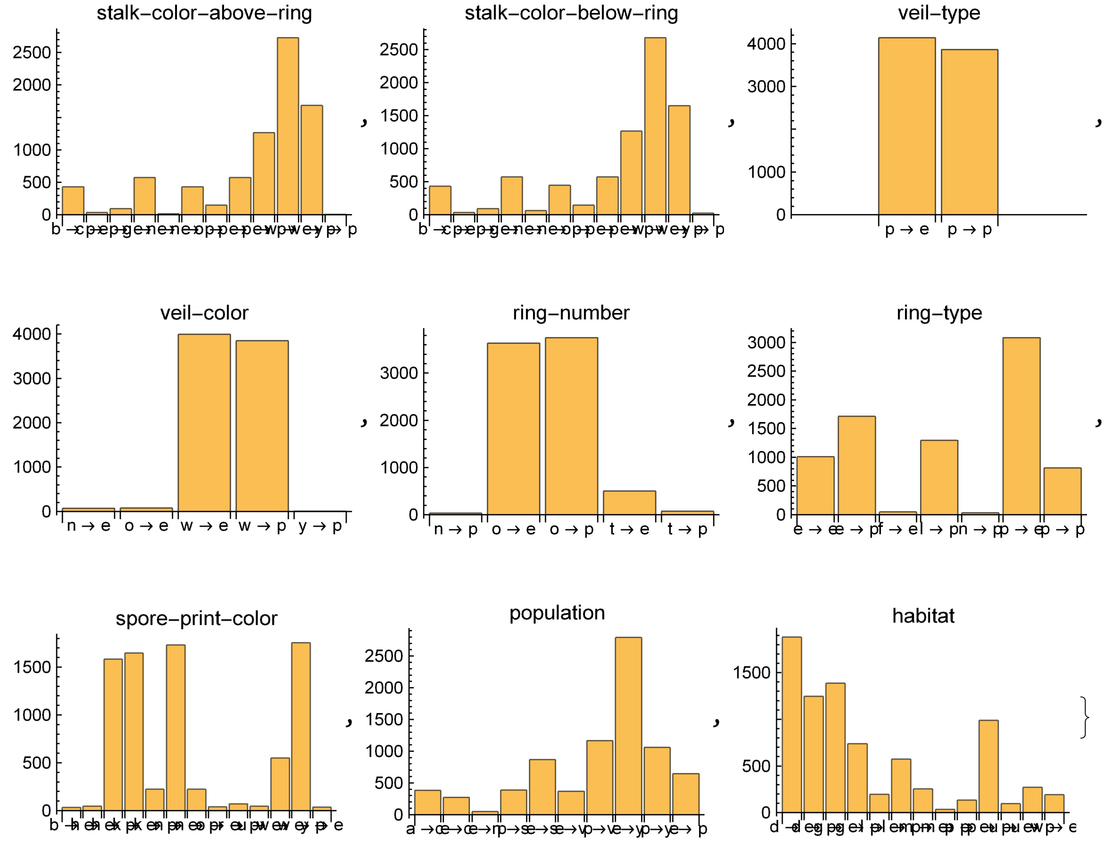


Bar charts for the rest of the 21 characteristics:

```
(Debug) In[ ]:= allBarChartsLabeled = Table[good = Table[mushroomCharacteristics[[n, characteristic]] →
                      mushroomCharacteristics[[n, 1]], {n, 8000}][[2 ;; -1]];
                  countsPE = KeySort[Counts[good]]; BarChart[countsPE, ChartLabels → Keys[countsPE],
                   PlotLabel → mushroomCharacteristics[[1, characteristic]]], {characteristic, 2, 23}]
```

(Debug) Out[•]=

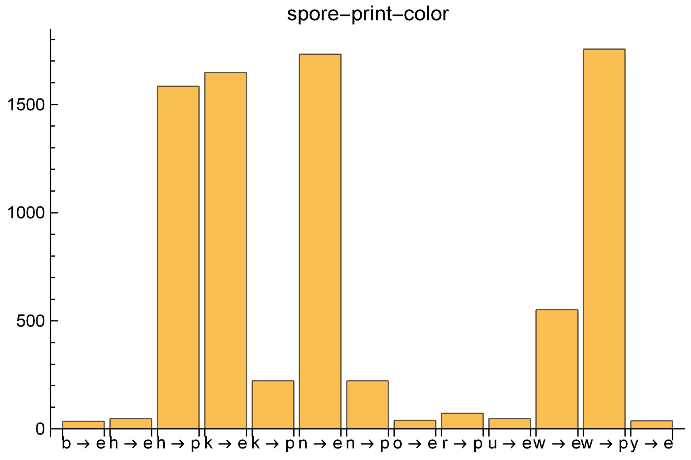

{
cap−shape
2000
1500
1000
500
0
b → e b → p c → p f → e f → p k → e k → p s → e s → p x → e x → p
cap−surface
1500
1000
500
0
f → e f → p g → p s → e s → p y → e y → p
cap−color
1200
1000
800
600
400
200
0
bruises
3000
2500
2000
1000
500
0
f → e
f → p
t → e
t → p
odor
3500
3000
2500
2000
1000
500
0
gill−attachment
4000
3000
1000
0
a → e
a → p
f → e
f → p
gill−spacing
3000
1000
0
c → e
c → p
w → e
w → p
gill−size
4000
3000
1000
0
b → e
b → p
n → e
n → p
gill−color
1500
1000
500
0
stalk−shape
2500
2000
1500
1000
500
0
e → e
e → p
t → e
t → p
stalk−root
2000
1500
500
0
stalk−surface−above−ring
3500
3000
2500
1500
1000
500
0
stalk−surface−below−ring
3500
3000
2500
2000
1500
1000
500
0
,



From first glance, spore print is a good predictor because almost all of the attributes are uniquely poisonous or edible. For example, spore print color “n”, brown, is an indicator for most edible mushrooms.

```
(Debug) In[ ]:= allBarChartsLabeled[[-3]]
```

(Debug) Out[ ]=



Multiple Variables

The combination of multiple characteristics has a higher probabiity being more uniquely poisonous or edible. I repeated the same process for finding good single-variable indicators, but this time with the pairings of multiple variables.

Cap shape (column 2) and cap color (column 3):

```
twoVariables = Table[
  mushroomCharacteristics[[n, {2, 3}]] → mushroomCharacteristics[[n, 1]], {n, 2, 8000}]
```

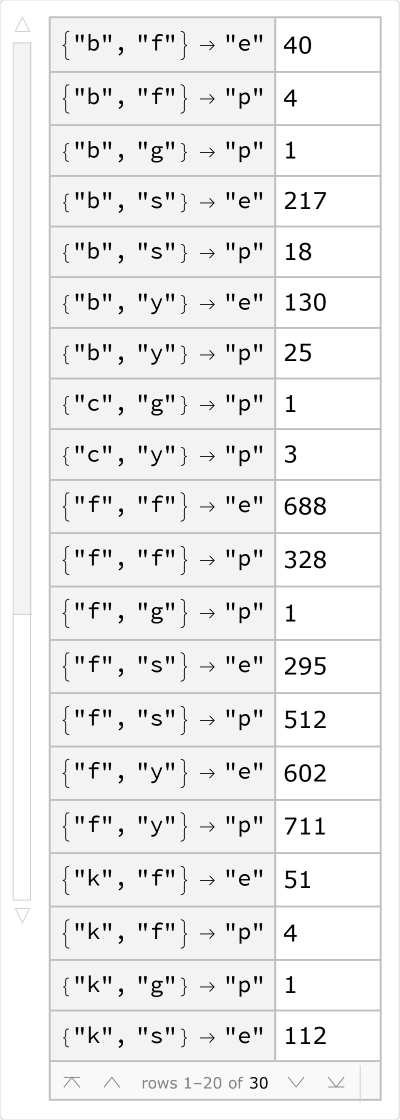
(Debug) Out[•]=

```
{{n, k} → p, {b, n} → e, {b, n} → e, {n, k} → p, {b, n} → e, {b, k} → e,
 {b, k} → e, {b, n} → e, {n, k} → p, {b, k} → e, {b, n} → e, {b, k} → e,
 {b, n} → e, {n, n} → p, {b, k} → e, {n, n} → e, {b, n} → e, {n, k} → p,
 ··· 7964 ···, {n, w} → p, {b, w} → e, {b, w} → e, {b, y} → e, {b, w} → e,
 {n, w} → p, {b, b} → e, {n, w} → p, {n, w} → p, {b, w} → e, {n, w} → p,
 {b, w} → e, {b, o} → e, {n, w} → p, {b, y} → e, {n, w} → p, {n, w} → p}

large output | show less | show more | show all | set size limit...
```

```
(Debug) In[◦]:= Dataset@KeySort[Counts[twoVariables]]
```

(Debug) Out[◦]=



{"b", "f"} → "e"	40
{"b", "f"} → "p"	4
{"b", "g"} → "p"	1
{"b", "s"} → "e"	217
{"b", "s"} → "p"	18
{"b", "y"} → "e"	130
{"b", "y"} → "p"	25
{"c", "g"} → "p"	1
{"c", "y"} → "p"	3
{"f", "f"} → "e"	688
{"f", "f"} → "p"	328
{"f", "g"} → "p"	1
{"f", "s"} → "e"	295
{"f", "s"} → "p"	512
{"f", "y"} → "e"	602
{"f", "y"} → "p"	711
{"k", "f"} → "e"	51
{"k", "f"} → "p"	4
{"k", "g"} → "p"	1
{"k", "s"} → "e"	112

At least for cap shape and cap color, all the combinations are relatively equal in number (in the 100's), and none of the combinations are obviously poisonous or edible. Therefore this combination is not a good predictor of edibility.

My next step is to export all single and double variable charts to Excel, to better identify the relevant characteristics.

Exporting to Excel

By exporting the charts to Excel, I was able to manually highlight the significant attributes that predict edibility.

Single Variable Set up

Exporting cap shape:

```
(Debug) In[ ]:= exportCapShape = Normal[countsCapShape] // TableForm
```

```
(Debug) Out[ ]//TableForm=
(x → p) → 1703
(x → e) → 1931
(b → e) → 387
(s → e) → 32
(f → e) → 1585
(f → p) → 1552
(b → p) → 48
(k → e) → 205
(k → p) → 552
(c → p) → 4
```

```
Export["file location.xls", exportCapShape, "XLS"]
```

Now, export all of the tables.

```
(Debug) In[ ]:= allCharts2 = Table[good = Table[
        mushroomCharacteristics[[n, characteristic]] → mushroomCharacteristics[[n, 1]],
        {n, 8000}][[2 ;; -1]]; countsPE = KeySort[Counts[good]];
     Normal[countsPE], {characteristic, 2, 23}];
```

```
Export["file location.xls", allCharts2 // TableForm, "XLSX", "FieldSeparators" → " "]
```

And then import into Google Spreadsheets, copy and paste transposed, make columns wider, and you’re good to go :)

After exporting the data itself, I labeled each of the 22 columns with the characteristic name.

Export characteristic names:

```
(Debug) In[ ]:= allCharacteristics =
   Table[mushroomCharacteristics[[1, characteristic]], {characteristic, 2, 23}];
```

```
Export["file location.xls",
 allCharacteristics // TableForm, "XLSX", "FieldSeparators" → " "]
```

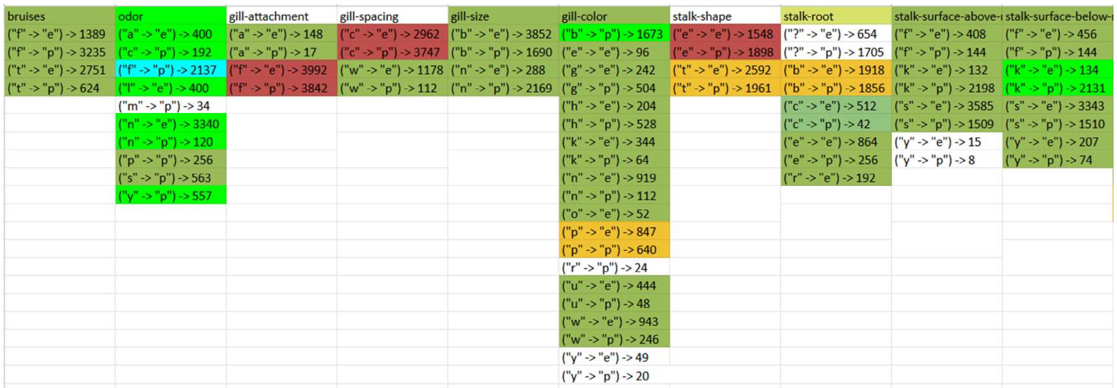
Link to google spreadsheet, with significant characteristics highlighted:

https://docs.google.com/spreadsheets/d/15L_AmXXT386yRl4Mpti_uVwpK8akxEkC9585lMeeWJE/edit?usp=sharing

Results: Significant Characteristics

Here is a sample of the spreadsheet with the individual characteristic data.

Red highlighting is insignificant, yellow could be significant, and green is significant. Arbitrarily, for an attribute to be highlighted as green, it has to be twice as likely to be edible than poisonous, or poisonous than edible.



bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	stalk-surface-above-	stalk-surface-below-
("f" -> "e") -> 1389	("a" -> "e") -> 400	("a" -> "e") -> 148	("c" -> "e") -> 2962	("b" -> "e") -> 3852	("b" -> "p") -> 1673	("e" -> "e") -> 1548	("?" -> "e") -> 654	("f" -> "e") -> 408	("f" -> "e") -> 456
("f" -> "p") -> 3235	("c" -> "p") -> 192	("a" -> "p") -> 17	("c" -> "p") -> 3747	("b" -> "p") -> 1690	("e" -> "e") -> 96	("e" -> "p") -> 1898	("?" -> "p") -> 1705	("f" -> "p") -> 144	("f" -> "p") -> 144
("t" -> "e") -> 2751	("f" -> "p") -> 2137	("f" -> "e") -> 3992	("w" -> "e") -> 1178	("n" -> "e") -> 288	("g" -> "e") -> 242	("t" -> "e") -> 2592	("b" -> "e") -> 1918	("k" -> "e") -> 132	("k" -> "e") -> 134
("t" -> "p") -> 624	("l" -> "e") -> 400	("f" -> "p") -> 3842	("w" -> "p") -> 112	("n" -> "p") -> 2169	("g" -> "p") -> 504	("t" -> "p") -> 1961	("b" -> "p") -> 1856	("k" -> "p") -> 2198	("k" -> "p") -> 2131
	("m" -> "p") -> 34				("h" -> "e") -> 204		("c" -> "e") -> 512	("s" -> "e") -> 3585	("s" -> "e") -> 3343
	("n" -> "e") -> 3340				("h" -> "p") -> 528		("c" -> "p") -> 42	("s" -> "p") -> 1509	("s" -> "p") -> 1510
	("n" -> "p") -> 120				("k" -> "e") -> 344		("e" -> "e") -> 864	("y" -> "e") -> 15	("y" -> "e") -> 207
	("p" -> "p") -> 256				("k" -> "p") -> 64		("e" -> "p") -> 256	("y" -> "p") -> 8	("y" -> "p") -> 74
	("s" -> "p") -> 563				("n" -> "e") -> 919		("r" -> "e") -> 192		
	("y" -> "p") -> 557				("n" -> "p") -> 112				
					("o" -> "e") -> 52				
					("p" -> "e") -> 847				
					("p" -> "p") -> 640				
					("r" -> "p") -> 24				
					("u" -> "e") -> 444				
					("u" -> "p") -> 48				
					("w" -> "e") -> 943				
					("w" -> "p") -> 246				
					("y" -> "e") -> 49				
					("y" -> "p") -> 20				

Relative to other characteristics, bruises, odor, gill size, gill color, stalk surface-above, stalk surface-below, spore-print color, are good predictors of edibility.

The results match well with a similar study, which found that odor, spore-print-color, habitat, gill-size, and cap-color are relevant characteristics (Al-Mejibli I., and Abd D., 2017).

Double Variable Excel Characteristics

The correlation of 2 variables produces a greater variety of unique identifiers for edibility. I repeated the same process for creating the charts as with the single variable correlation. Then, I exported all 484 charts to Excel.

```
(Debug) In[ ]:= storeList = {};
allCharts2Variables = Table[
    ((good = Table[mushroomCharacteristics[[n, {characteristic1, characteristic2}]] →
         mushroomCharacteristics[[n, 1]], {n, 2, 8000}]);
     storeList = AppendTo[storeList, Normal[ KeySort[Counts[good]]]];)
   , {characteristic1, 2, 23}, {characteristic2, 2, 23}];
```

```
Export["file name.xls", storeList // TableForm, "XLSX", "FieldSeparators" → " "]
```

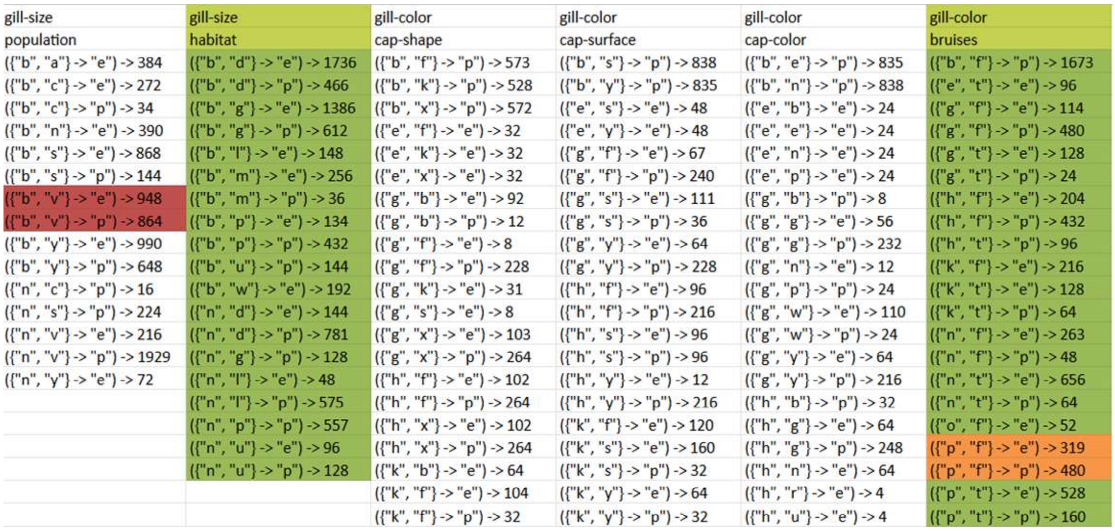
Now export 484 section labels

```
(Debug) In[ ]:= storeListLabels = {};
allCharts2Variables = Table[
    storeList = AppendTo[storeListLabels, {mushroomCharacteristics[[1, characteristic1]],
        mushroomCharacteristics[[1, characteristic2]]}];
   , {characteristic1, 2, 23}, {characteristic2, 2, 23}];
```

```
Export["file name.xls", storeListLabels // TableForm, "XLSX", "FieldSeparators" → " "]
```

I did not analyze each of the 484 combinations, instead, I took a look at the combination of the significant single-variable characteristics found earlier.

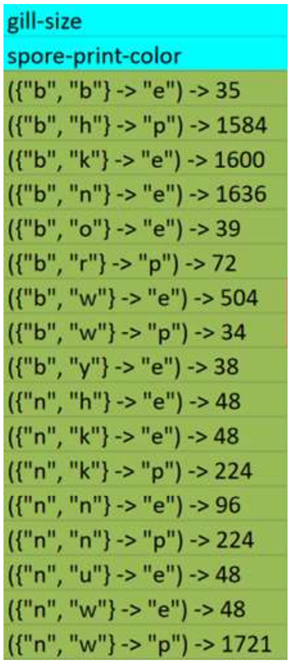
Link to full spreadsheet: https://docs.google.com/spreadsheets/d/1k_XzSnoty2YYJPgamyNX-Hta-moemOKxMo5OMPXV5BU4/edit?usp=sharing



gill-size	gill-size	gill-color	gill-color	gill-color	gill-color
population	habitat	cap-shape	cap-surface	cap-color	bruises
({"b", "a"} -> "e") -> 384	({"b", "d"} -> "e") -> 1736	({"b", "f"} -> "p") -> 573	({"b", "s"} -> "p") -> 838	({"b", "e"} -> "p") -> 835	({"b", "f"} -> "p") -> 1673
({"b", "c"} -> "e") -> 272	({"b", "d"} -> "p") -> 466	({"b", "k"} -> "p") -> 528	({"b", "y"} -> "p") -> 835	({"b", "n"} -> "p") -> 838	({"e", "t"} -> "e") -> 96
({"b", "c"} -> "p") -> 34	({"b", "g"} -> "e") -> 1386	({"b", "x"} -> "p") -> 572	({"e", "s"} -> "e") -> 48	({"e", "b"} -> "e") -> 24	({"g", "f"} -> "e") -> 114
({"b", "n"} -> "e") -> 390	({"b", "g"} -> "p") -> 612	({"e", "f"} -> "e") -> 32	({"e", "y"} -> "e") -> 48	({"e", "e"} -> "e") -> 24	({"g", "f"} -> "p") -> 480
({"b", "s"} -> "e") -> 868	({"b", "l"} -> "e") -> 148	({"e", "k"} -> "e") -> 32	({"g", "f"} -> "e") -> 67	({"e", "n"} -> "e") -> 24	({"g", "t"} -> "e") -> 128
({"b", "s"} -> "p") -> 144	({"b", "m"} -> "e") -> 256	({"e", "x"} -> "e") -> 32	({"g", "f"} -> "p") -> 240	({"e", "p"} -> "e") -> 24	({"g", "t"} -> "p") -> 24
({"b", "v"} -> "e") -> 948	({"b", "m"} -> "p") -> 36	({"g", "b"} -> "e") -> 92	({"g", "s"} -> "e") -> 111	({"g", "b"} -> "p") -> 8	({"h", "f"} -> "e") -> 204
({"b", "v"} -> "p") -> 864	({"b", "p"} -> "e") -> 134	({"g", "b"} -> "p") -> 12	({"g", "s"} -> "p") -> 36	({"g", "g"} -> "e") -> 56	({"h", "f"} -> "p") -> 432
({"b", "y"} -> "e") -> 990	({"b", "p"} -> "p") -> 432	({"g", "f"} -> "e") -> 8	({"g", "y"} -> "e") -> 64	({"g", "g"} -> "p") -> 232	({"h", "t"} -> "p") -> 96
({"b", "y"} -> "p") -> 648	({"b", "u"} -> "p") -> 144	({"g", "f"} -> "p") -> 228	({"g", "y"} -> "p") -> 228	({"g", "n"} -> "e") -> 12	({"k", "f"} -> "e") -> 216
({"n", "c"} -> "p") -> 16	({"b", "w"} -> "e") -> 192	({"g", "k"} -> "e") -> 31	({"h", "f"} -> "e") -> 96	({"g", "p"} -> "p") -> 24	({"k", "t"} -> "e") -> 128
({"n", "s"} -> "p") -> 224	({"n", "d"} -> "e") -> 144	({"g", "s"} -> "e") -> 8	({"h", "f"} -> "p") -> 216	({"g", "w"} -> "e") -> 110	({"k", "t"} -> "p") -> 64
({"n", "v"} -> "e") -> 216	({"n", "d"} -> "p") -> 781	({"g", "x"} -> "e") -> 103	({"h", "s"} -> "e") -> 96	({"g", "w"} -> "p") -> 24	({"n", "f"} -> "e") -> 263
({"n", "v"} -> "p") -> 1929	({"n", "g"} -> "p") -> 128	({"g", "x"} -> "p") -> 264	({"h", "s"} -> "p") -> 96	({"g", "y"} -> "e") -> 64	({"n", "f"} -> "p") -> 48
({"n", "y"} -> "e") -> 72	({"n", "l"} -> "e") -> 48	({"h", "f"} -> "e") -> 102	({"h", "y"} -> "e") -> 12	({"g", "y"} -> "p") -> 216	({"n", "t"} -> "e") -> 656
	({"n", "l"} -> "p") -> 575	({"h", "f"} -> "p") -> 264	({"h", "y"} -> "p") -> 216	({"h", "b"} -> "p") -> 32	({"n", "t"} -> "p") -> 64
	({"n", "p"} -> "p") -> 557	({"h", "x"} -> "e") -> 102	({"k", "f"} -> "e") -> 120	({"h", "g"} -> "e") -> 64	({"o", "f"} -> "e") -> 52
	({"n", "u"} -> "e") -> 96	({"h", "x"} -> "p") -> 264	({"k", "s"} -> "e") -> 160	({"h", "g"} -> "p") -> 248	({"p", "f"} -> "e") -> 319
	({"n", "u"} -> "p") -> 128	({"k", "b"} -> "e") -> 64	({"k", "s"} -> "p") -> 32	({"h", "n"} -> "e") -> 64	({"p", "f"} -> "p") -> 480
		({"k", "f"} -> "e") -> 104	({"k", "y"} -> "e") -> 64	({"h", "r"} -> "e") -> 4	({"p", "t"} -> "e") -> 528
		({"k", "f"} -> "p") -> 32	({"k", "y"} -> "p") -> 32	({"h", "u"} -> "e") -> 4	({"p", "t"} -> "p") -> 160

The best combination was either spore-print color or gill size, or spore-print color or odor. Almost all of the combinations are unique, where the combination of two attributes are either all poisonous or all edible.

Combination of spore print color and gill size:



gill-size
spore-print-color
({"b", "b"} -> "e") -> 35
({"b", "h"} -> "p") -> 1584
({"b", "k"} -> "e") -> 1600
({"b", "n"} -> "e") -> 1636
({"b", "o"} -> "e") -> 39
({"b", "r"} -> "p") -> 72
({"b", "w"} -> "e") -> 504
({"b", "w"} -> "p") -> 34
({"b", "y"} -> "e") -> 38
({"n", "h"} -> "e") -> 48
({"n", "k"} -> "e") -> 48
({"n", "k"} -> "p") -> 224
({"n", "n"} -> "e") -> 96
({"n", "n"} -> "p") -> 224
({"n", "u"} -> "e") -> 48
({"n", "w"} -> "e") -> 48
({"n", "w"} -> "p") -> 1721

I will be using this chart to predict edibility given spore print and gill size. Spore print is the color of the spores found in the gills of a mushroom. A common technique to extract the color of the spores is to leave the cap of the the mushroom on a piece of paper overnight.

For example, if the gill size is “n” narrow, and spore print is “w” white, based on UCI dataset (last line in the chart) there were only 48 entries for that combination that resulted in an edible mushroom, in comparison to an overwhelming poisonous 1721 entries. Therefore, this mushroom with a narrow gill size and white spore print would be categorized as poisonous. Similarly, mushroom entries with a broad gill size and a brown spore print (“n”) were all edible per the UCI dataset.

Analysis

When identifying good predictors of edibility, I also focused on the factors that predict if a mushroom is most likely poisonous.

Because there are only two classes, “edible” and “poisonous”, I could have also just focused on identifying attributes that result in edibility. If a mushroom does not contain those attributes then, then it would be classified as poisonous. Using this approach however, I would have had to identify many more attributes that uniquely indicate if a mushroom is edible. For example, only edible mushrooms have an abundant population or a sunken cap shape.

Given my goal to extract these characteristics from an image, this approach of only identifying edible characteristics would not have been time feasible, because I would have had to have created many more training image datasets to extract them. However, this is an outlet for future work: focusing only on the characteristics that identify a mushroom as edible, to see if there is a greater accuracy.

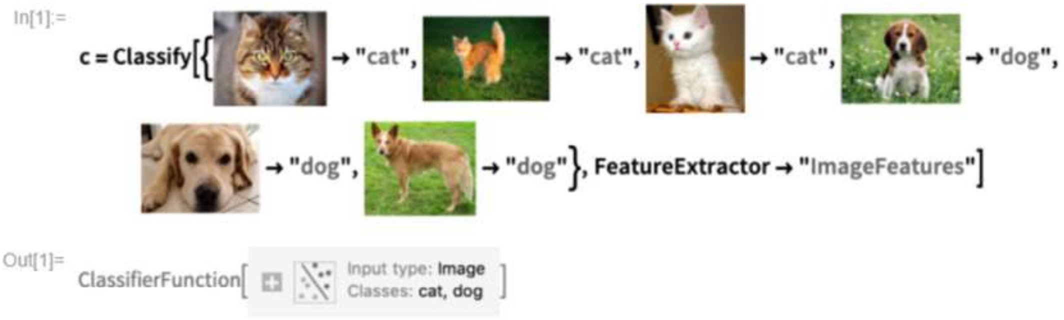
Creating training image datasets for spore print and gill size

From the previous step, I now have a way of predicting edibility given spore print and gill size.

Next, I created functions to extract the spore print and gill size by creating training image datasets.

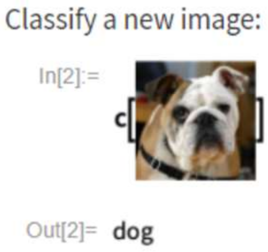
The overall goal is to use these two *Classify* functions (machine learning functions in the Wolfram Language) to extract spore print and gill size from an image, and then use the UCI chart in the previous step to predict edibility.

Using the Classify function, you can label training images with the desired aspect you wish to extract, such as whether the image is a cat or dog. For example:



```
In[1]:=  c = Classify[{ → "cat",  → "cat",  → "cat",  → "dog",
          → "dog",  → "dog"}, FeatureExtractor → "ImageFeatures"]

Out[1]=  ClassifierFunction[ Input type: Image
                             Classes: cat, dog ]
```



Classify a new image:

```
In[2]:= c[ ]

Out[2]= dog
```

Thank you the Wolfram Documentation center for providing this example (Wolfram Language & System).

I will repeat this process, but instead of labelling the images with whether they're a cat or dog, I'll create two training sets: one with mushrooms labeled with spore print, and one labeled with gill size. With the training sets I'll be able to create functions that extract these two characteristics, and then use the UCI dataset to predict edibility.

From an earlier section, my image dataset consists of 98 edible mushrooms, 33 inedible mushrooms, and 35 lethally poisonous mushrooms. Each individual mushroom contains approximately 10 - 15 images, creating an image dataset of approximately 3000 images.

Before I can use the images however, I need to webscrape further data from WildFoodUK regarding spore print and gill size, so that I can label the images with the correct characteristic.

Spore print

For each individual mushroom on WildFoodUK, there is a description of the spore print. For the edible, nonedible, and lethally poisonous mushrooms, I webscraped the first 6 words describing spore print, and then manually converted into the categorical classification as seen in the UCI dataset.

Spore prints for the 98 edible mushrooms:

```
(Debug) In[ ]:= listediblesporeprintcolors =
  Table[textMaybe = (Import[evenbetterhyperlinks[[n]], "Plaintext"]);
   Which[StringContainsQ[textMaybe, "Spore Print"], StringExtract[
     StringTake[textMaybe, {Last@Flatten@StringPosition[textMaybe, "Spore Print"],
       Last@Flatten@StringPosition[textMaybe, "Spore Print"] + 30}], 2 ;; 8],
    StringContainsQ[textMaybe, "Spore"], StringExtract[StringTake[textMaybe,
      {Last@Flatten@StringPosition[textMaybe, "Spore"], Last@Flatten@StringPosition[
          textMaybe, "Spore"] + 30}], 2 ;; 8]], {n, 1, Length@evenbetterhyperlinks}]

(Debug) Out[ ]= {{"Dark", "purple/brown.", "Ellipsoi"}, {"Purple/chocolate", "brown.", "Ell"},
  {"Brown.", "Subglobose.", ""}, {"Chocolate", "brown.", "Ellipsoid."},
  {"Chocolate", "brown.", "Ovoid.", "You"}, {"Brown.", "Ellipsoid.", ""},
  {"Purple/brown.", "Ellipsoid.", ""}, {"Brown.", "Ovoid.", ""},
  {"Purple/brown.", "Ellipsoid.", ""}, {"White,", "ellipsoid.", "These", "ar"},
  {"White,", "subglobose.", ""}, {"White.", "Ellipsoid.", "You", "shoul"},
  {"White.", "Spherical.", ""}, {"White.", "Ovoid.", ""}, {"White.", "Spherical.", ""},
  {"White", "to", "pale", "cream.", "Ellips"}, {"White.", "Ellipsoid", "and", "smooth"},
{"White.", "Sausage", "shaped.", ""}, {"Olivaceous/brown.", "Subfusifo"},
{"Cinnamon.", "Subfusiform,", "elli"}, {"Pale", "straw", "coloured.", "Ellips"},
{"Olive", "brown.", "Subfusiform.", ""}, {"Olive-brown.", "Subfusiform", "to"},
{"Green/brown.", "Subfusiform.", ""}, {"Olivaceous", "brown.", "Ellipsoid"},
{"Brown.", "Subfusiform.", ""}, {"Brown.", "Subfusiform.", ""},
{"Olive-brown.", "Subfusiform", "to"}, {"Olive-brown.", "Subfusiform", "to"},
{"Olive-brown.", "Subfusiform.", ""}, {"Oche-sienna", "coloured.", "Subfu"},
{"Brown.", "Subfusiform.", "The", "ima"}, {"Olive", "green", "to", "brown.", "Subfu"},
{"Olive/brown.", "Subfusiform.", ""}, {"Green/brown.", "Subfusiform.", ""},
{"Olive", "Brown.", "Ellipsoid.", ""}, {"Olive", "brown.", "Subfusiform.", ""},
{"White.", "Ellipsoid.", ""}, {"Yellow/brown.", "Spherical", "wit"},
{"Ochraceous.", "Ellipsoid.", ""}, {"Off-white.", "Subglobose.", ""},
{"White.", "Ellipsoid.", ""}, {"White.", "Subglobose.", "You", "shou"},
{"Pink.", "Ellipsoid.", ""}, {"Date", "brown.", "Mitriform.", ""},
{"Blackish", "brown.", "Ellipsoid.", ""}, {"Cream", "to", "salmon", "to", "yellow.", ""},
{"Brown", "Taste", ""}, {"Pink/pale", "ochre.", "Ovate.", ""},
{"Slightly", "off", "white.", "Ellipso"}, {"White.", "Ellipsoid,", "cylindric"},
{"White,", "broadly", "ellipsoid.", "Y"}, {"White.", "Broadly", "ellipsoid", "to"},
{"White,", "Ellipsoid.", ""}, {"Off", "white", "to", "cream.", "Ellipso"},
{"White.", "Ellipsoid.", "You", "shoul"}, {"White.", "Ellipsoid.", ""},
{"White.", "Ellipsoid.", ""}, {"White.", "Ellipsoid.", ""},
{"White", "to", "pale", "yellow.", "Ellip"}, {"White,", "ellipsoid.", ""},
{"White.", "Ellipsoid.", ""}, {"White.", "Ellipsoid.", ""},
{"White.", "Globose", "with", "spines."}, {"White,", "cream/yellow.", "Globos"},
{"Cream", "coloured.", "Subglobose."}, {"Pale", "ochre", "to", "pale", "salmon", "c"},
{"Pale", "ochre.", "Ellipsoid", "with"}, {"White.", "Ellipsoid", "to", "broadly"},
{"White-cream.", "Subglobose.", "Yo"}, {"Off-white", "to", "pale", "pink.", "Ell"},
{"Off", "white", "to", "Pale", "pink.", "Ell"}, {"Olive/brown.", ""},
{"Olive/brown.", "Globose", "with", "f"}, {"The", "spores", "start", "inside", "the"},
{"White/off", "white.", "Ellipsoid."}, {"White.", "Ovoid,", "dextrinoid.", ""},
{"White.", "Ellipsoid.", ""}, {"White.", "Ovoid,", "ellipsoid.", ""},
{"Pa", "le", "cream", "to", "yellow.", "Elli"}, {"Cream.", "Ellipsoid.", ""},
{"Cream", "to", "pale", "yellow.", ""}, {"Pale", "cream", "to", "yellow.", "Ellip"},
{"White.", "Has", "many", "spores", "and", ""}, {"Off", "white.", "Broadly", "ellipsoi"},
{"Pale", "yellow.", "Oblong.", "As", "the"}, {"Lilac.", "Cylindrical.", ""},
{"White.", "Ovoid,", "ellipsoid.", "Yo"}, {"White", "to", "off", "white", "to", "pale"},
{"Off", "white.", "Subglobose", "to", "gl"}, {"Pale", "ochre.", "Ellipsoid", "with", ""},
{"White", "to", "pale", "cream.", "Broadl"}, {"White/cream.", "Broadly", "ovoid."},
{"Pale", "cream.", "Subglobose.", ""}, {"White/off", "white.", "Ellipsoid,"},
{"White.", "Ellipsiod", "to", "cylindr"}, {"Colourless", "unless", "in", "large", ""},
{"Blackish", "brown", "to", "brown.", "Ov"}}
```

I created a text file to convert these descriptions consistent with the the UCI categorical entries of “white”, “brown”, “chocolate”, “buff”, “orange”, “green” and “purple”.

As you can tell, there are many discrepancies between these two datasets. In what category should “Dark chocolate purple” go in? Because only 40 out of the 8000 entries in the UCI characteristic dataset were labeled as purple, “Dark chocolate purple” did not go into the “purple” category, but instead in the “chocolate” category (approximately 1000 of those entries), with a chocolate spore print being a much more common classification. Similarly, “Olive brown” went in the “brown” category, instead of the less common “green” category.

New and improved spore prints:

```
ediblePrints = Import["text file", "List"]
```

```
(Debug) Out[ ]= {Chocolate, Chocolate, Brown, Chocolate, Chocolate, Brown, Chocolate, Brown,
  Chocolate, White, White, White, White, White, White, White, White, White, Brown,
  Orange, Buff, Brown, Brown, Green, Brown, Brown, Brown, Brown, Brown, Brown, Buff,
  Brown, Green, Brown, Green, Brown, Brown, White, Buff, Purple, White, White,
  White, Pink, Brown, Black, Pink, Brown, Purple, White, White, White, White, White,
  White, White, White, White, White, Yellow, White, White, White, White, White,
  White, Purple, Purple, White, White, White, White, Brown, Brown, Brown, White,
  White, White, White, Yelow, White, Yellow, Yellow, White, White, Yellow, Purple,
  White, White, White, Purple, White, White, White, White, White, Clear, Black}
```

I repeated this process for inedible and lethally poisonous mushrooms.

Inedible spore prints:

```
(Debug) Out[ ]= {White, White, White, White, Brown, Brown, Pink, White, White, Black, White,
  Orange, White, Black, White, Brown, White, White, White, White, Black, White,
  Clear, Yellow, Brown, Brown, White, Purple, Green, White, Pink, White, White}
```

Lethally poisonous spore prints:

```
(Debug) Out[ ]= {Brown, Chocolate, White, White, White, White, White, White, White, Brown, Black, White,
  Brown, Clear, White, White, White, White, Chocolate, Brown, Brown, Brown, White, Yellow,
  White, White, White, White, Brown, White, White, White, Brown, Chocolate, White}
```

After webscraping the spore print, I assigned it to the images it corresponds with. ie edible mushroom #1 images -> edible mushroom #1 spore print.

```
(Debug) In[ ]:= edibleSporeTrainingData =
   Table[(Table[(edibleMushroomsBetter[[each]])[[n]] → ediblePrints[[each]],
      {n, 1, Length@edibleMushroomsBetter[[each]]}]),
    {each, 1, Length@edibleMushroomsBetter}];
```

Example labelling for edible mushroom #2:

```
(Debug) In[ ]:= edibleSporeTrainingData[[2]]
```

{

```
→ "Chocolate",
```

→ "Chocolate",

→ "Chocolate",

→ "Chocolate",

→ "Chocolate",

→ "Chocolate",

→ "Chocolate",

→ "Chocolate",

→ "Chocolate",

→ "Chocolate",

→ "Chocolate",

→ "Chocolate",

→ "Chocolate",

→ "Chocolate",

→ "Chocolate",

→ "Chocolate", → "Chocolate",

→ "Chocolate", → "Chocolate",

→ "Chocolate", → "Chocolate",

→ "Chocolate", → "Chocolate"}

(Debug) In[ ]:=

```
inedibleSporeTrainingData =
  Table[(Table[(inedibleMushroomsBetter[[each]])[[n]] → ediblePrints[[each]],
     {n, 1, Length@inedibleMushroomsBetter[[each]]}]),
   {each, 1, Length@inedibleMushroomsBetter}];
```

inedible mushroom #7

(Debug) In[ ]:=

```
inedibleSporeTrainingData[[7]]
```

{ → "Pink", → "Pink", → "Pink",

→ "Pink", → "Pink", → "Pink",

→ "Pink", → "Pink", → "Pink",

→ "Pink", → "Pink", → "Pink",

→ "Pink", → "Pink", → "Pink",

→ "Pink", → "Pink", → "Pink"}

```
(Debug) In[]:= lethallyinedibleSporeTrainingData = Table[
        (Table[(lethallyPoisonousMushrooms[[each]])[[n]] → lethallyinediblePrints[[each]],
          {n, 1, Length@lethallyPoisonousMushrooms[[each]]}]),
        {each, 1, Length@lethallyPoisonousMushrooms}];
```

lethally poisonous mushroom #23

```
(Debug) In[]:= lethallyinedibleSporeTrainingData[[23]]
```

{ → "White", → "White", → "White",

→ "White", → "White", → "White",

→ "White", → "White", → "White",

→ "White", → "White", → "White",

→ "White", → "White", → "White",

→ "White", → "White", → "White"}

I combined all the training images to create a classify function to extract spore print.

```
(Debug) In[ ]:= sporePrintTrainingData = Flatten[Join[edibleSporeTrainingData,
           inedibleSporeTrainingData, lethallyinedibleSporeTrainingData]];
```

Took 2 hours to process:

```
extractSporePrint = Classify[sporePrintTrainingData]
```

```
(Debug) Out[•]= ClassifierFunction[ Input type: Image
                                     Number of classes: 12 ]
```

```
(Debug) In[•]:= extractSporePrint[
```

```
]
```

```
(Debug) Out[•]= Orange
```

```
(Debug) In[•]:= extractSporePrint[
```

```
]
```

```
(Debug) Out[•]= Brown
```

```
(Debug) In[•]:= extractSporePrint[
```

```
]
```

```
(Debug) Out[•]= White
```

So far, the functions accurately extracts spore print!

Gill Size

Per the UCI dataset, the two classifications for gill size are “broad” and “narrow”. I was unsure if this was the horizontal or vertical width of the gills - there wasn’t documentation for it. Because another characteristic was gill spacing, I assumed this to be the vertical width.

For the WildFoodUK mushrooms, there was not specifically a category for gill size, but instead cap size. I assumed that if the cap size is “broad”, than the gill size would be broad as well, and if the cap size was narrow, the gill size would be narrow as well. Of course, this is not always the case.

Cap size was listed numerically in cm for each mushroom entry on WildFoodUK. To categorize it into “broad” and “narrow”, I took the average of the cap sizes for all the mushrooms (12 cm, 5 inches). Then, if the value was less, the gill size was labeled as “narrow”, if it was greater the gill size was labeled as “broad”. Sometimes, the gill size varied drastically, and for those entries I just removed the data.

Extracting all the cap sizes for lethal mushrooms :

```
(Debug) In[ ]:= listlethallycapsizes =
                 Table[textMaybe = (Import[goodhyperlinksLethallyPoisonous[[n]], "Plaintext"]);
                  StringExtract[StringTake[textMaybe,
                    {Last@Flatten@StringPosition[textMaybe, "Average Cap width"],
                     Last@Flatten@StringPosition[textMaybe, "Average Cap width"] + 30}], 3],
                  {n, 1, Length@goodhyperlinksLethallyPoisonous}]

(Debug) Out[ ]= {10, 16, 7, 25, 10, 12, 12, 4, 20, 8, 5, 13, 4, 4-14, 3-8,
                 6, 7, 8, 7, 6, 4, 6, 10, 12, 5, 4, 5, 8, 20, 10, 10, 10, 12, 8, 6}
```

Removing ones with too large of a range :

```
(Debug) In[ ]:= listlethallycapsizes = ReplacePart[listlethallycapsizes, {14 -> Nothing, 15 → Nothing}]

(Debug) Out[ ]= {10, 16, 7, 25, 10, 12, 12, 4, 20, 8, 5, 13, 4, 6,
                 7, 8, 7, 6, 4, 6, 10, 12, 5, 4, 5, 8, 20, 10, 10, 10, 12, 8, 6}
```

Then repeat for inedible and edible mushrooms.

Combining all the cap sizes into one large list:

```
(Debug) In[ ]:= allcapsizes = ToExpression@
                  Flatten[Join[{listediblecapsizes, listinediblecapsizes, listlethallycapsizes}]]

(Debug) Out[ ]= {25, 20, 15, 20, 10, 15, 10, 10, 10, 10, 10, 10, 9, 15, 15, 5, 20, 8, 10, 12, 8, 15, 10, 15, 20,
                 20, 16, 10, 15, 8, 12, 6, 20, 10, 15, 15, 80, 10, 5, 60, 20, 10, 4, 15, 5, 7, 20, 10, 30,
                 18, 4, 10, 4, 3, 5, 4, 4, 3, 8, 8, 6, 6, 9, 10, 8, 45, 10, 12, 12, 4, 5, 14, 12, 30, 5, 80,
                 15, 5, 3, 15, 8, 5, 5, 15, 7, 5, 10, 10, 15, 10, 10, 13, 6, 25, 8, 12, 10, 15, 20, 10, 15,
                 3, 0.5, 12, 6, 6, 3, 4.5, 4, 5, 10, 20, 4, 25, 15, 4, 1, 4, 7, 6, 10, 10, 16, 7, 25, 10, 12,
                 12, 4, 20, 8, 5, 13, 4, 6, 7, 8, 7, 6, 4, 6, 10, 12, 5, 4, 5, 8, 20, 10, 10, 10, 12, 8, 6}
```

Taking average:

```
(Debug) In[ ]:= Mean[allcapsizes]

(Debug) Out[ ]= 11.8247
```

The average of all cap sizes is approximately 12cm. Therefore, if a cap size is less than 12 cm, it will classified as “narrow”, and if it is greater than 12 cm, it will be classified as “broad”.

If the mushroom has a variable cap width, such as 4 - 14 cm, then that entry was classified as “clear”, and was later deleted from the training dataset.

Lethal poisonous mushroom gill size classification:

```
(Debug) In[ ]:= lethallyPoisonousGillSize =
                 Table[Which[ToExpression[listlethallycapsizes][[n]] > 12, "broad",
                   ToExpression[listlethallycapsizes][[n]] > 0, "narrow", True, "clear"],
                  {n, Length@ToExpression[listlethallycapsizes]}]

(Debug) Out[ ]= {narrow, broad, narrow, broad, narrow, narrow, narrow, narrow,
                 broad, narrow, narrow, broad, narrow, clear, clear, narrow, narrow,
                 narrow, narrow, narrow, narrow, narrow, narrow, narrow, narrow, narrow,
                 narrow, narrow, broad, narrow, narrow, narrow, narrow, narrow, narrow}
```

Inedible gill size classification:

```
(Debug) In[ ]:= inedibleGillSize = Table[Which[ToExpression[listinediblecapsizes][[n]] > 12, "broad",
                   ToExpression[listinediblecapsizes][[n]] > 0, "narrow", True, "clear"],
                  {n, Length@ToExpression[listinediblecapsizes]}]
```

```
(Debug) Out[ ]= {narrow, narrow, clear, clear, broad, broad, narrow, clear, broad, narrow, narrow,
                 narrow, clear, narrow, clear, narrow, narrow, narrow, narrow, narrow, narrow, broad,
                 narrow, broad, clear, broad, narrow, narrow, narrow, narrow, narrow, clear, narrow}
```

Edible gill size classification:

```
(Debug) In[ ]:= edibleGillSize = Table[Which[ToExpression[listediblecapsizes][[n]] > 12,
                   "broad", ToExpression[listediblecapsizes][[n]] > 0, "narrow", True, "clear"],
                  {n, Length@ToExpression[listediblecapsizes]}]
```

```
(Debug) Out[ ]= {broad, broad, broad, broad, narrow, broad, narrow, narrow, narrow, narrow, narrow, narrow,
                 narrow, broad, clear, broad, clear, narrow, broad, narrow, narrow, narrow, narrow,
                 broad, narrow, broad, broad, broad, broad, narrow, broad, narrow, narrow, narrow,
                 broad, narrow, broad, broad, broad, narrow, narrow, broad, broad, narrow, narrow,
                 broad, narrow, narrow, broad, narrow, broad, clear, broad, narrow, narrow, narrow,
                 narrow, narrow, narrow, narrow, narrow, narrow, narrow, narrow, narrow, narrow,
                 narrow, narrow, broad, narrow, narrow, narrow, narrow, narrow, broad, narrow,
                 broad, narrow, broad, broad, narrow, narrow, broad, narrow, narrow, narrow, broad,
                 narrow, narrow, narrow, narrow, broad, narrow, narrow, broad, narrow, broad, narrow}
```

Similar to gill size, I now associated the mushroom images to their gill size. ie lethal mushroom #1 -> lethal mushroom #1 gill size.

Creating the training datasets to classify gill size:

Lethal mushrooms:

```
(Debug) In[ ]:= lethallyinedibleGillsTrainingData =
                  Table[(Table[(lethallyPoisonousMushrooms[[each]])[[n]] → lethallyPoisonousGillSize[[
                        each]], {n, 1, Length@lethallyPoisonousMushrooms[[each]]}]),
                   {each, 1, Length@lethallyPoisonousMushrooms}];
```

Inedible mushrooms:

```
(Debug) In[ ]:= inedibleGillsTrainingData =
                  Table[(Table[(inedibleMushroomsBetter[[each]])[[n]] → inedibleGillSize[[each]],
                      {n, 1, Length@inedibleMushroomsBetter[[each]]}]),
                   {each, 1, Length@inedibleMushroomsBetter}];
```

Example gill classification for inedible mushroom #1:

```
(Debug) In[ ]:= inedibleGillsTrainingData[[1]]
```

{ → "narrow", → "narrow",

→ "narrow", → "narrow",

→ "narrow", → "narrow",

→ "narrow", → "narrow",

→ "narrow", → "narrow",

→ "narrow", → "narrow",

→ "narrow", → "narrow",

→ "narrow", → "narrow",

→ "narrow", → "narrow", → "narrow",

→ "narrow", → "narrow", → "narrow",

→ "narrow", → "narrow"}

And last and not least gill size classification for edible mushrooms:

```
(Debug) In[]:= edibleGillsTrainingData =
                 Table[(Table[(edibleMushroomsBetter[[each]])[[n]] → edibleGillSize[[each]],
                    {n, 1, Length@edibleMushroomsBetter[[each]]}]),
                  {each, 1, Length@edibleMushroomsBetter}];
```

Sanity check, to makes sure there is a proper amount of training data:

```
(Debug) In[]:= Length@inedibleGillsTrainingData

(Debug) Out[]= 33
```

```
(Debug) In[]:= Length@lethallyinedibleGillsTrainingData

(Debug) Out[]= 35
```

```
(Debug) In[]:= Length@edibleGillsTrainingData

(Debug) Out[]= 98
```

There is indeed training data for all the 98, 33, and 35 edible, inedible, and lethally poisonous mushrooms, respectively.

Combining the edible, inedible, and lethally poisonous gill size training data, to create a function to classify gill size:

```
(Debug) In[]:= allGillsTrainingData = Flatten[Join[edibleGillsTrainingData,
                   inedibleGillsTrainingData, lethallyinedibleGillsTrainingData]];
```

This classify function only took 30 minutes to run in comparison to the 2 hours for spore print, probably because there are only 2 categorizations (broad and narrow).

```
extractGillSize = Classify[allGillsTrainingData]
```

```
(Debug) Out[]= ClassifierFunction[ Input type: Image
                                   Classes: broad, narrow ]
```

```
(Debug) In[]:= extractGillSize[
```

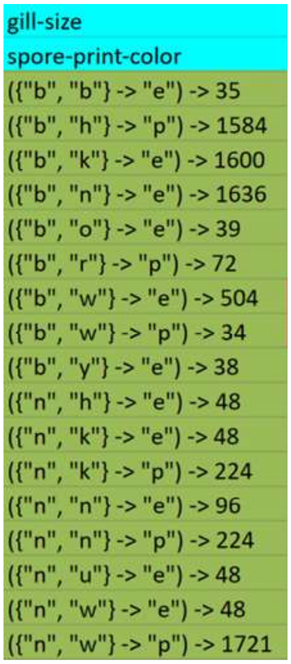
```
]
```

```
(Debug) Out[]= narrow
```

Putting it all together

I have the counts from the UCI dataset to identify edibility of a mushroom given the spore print and gill size. And I have two Classify functions to extract spore print and gill size.

Refresher for the UCI classification chart:



gill-size
spore-print-color
({"b", "b"} -> "e") -> 35
({"b", "h"} -> "p") -> 1584
({"b", "k"} -> "e") -> 1600
({"b", "n"} -> "e") -> 1636
({"b", "o"} -> "e") -> 39
({"b", "r"} -> "p") -> 72
({"b", "w"} -> "e") -> 504
({"b", "w"} -> "p") -> 34
({"b", "y"} -> "e") -> 38
({"n", "h"} -> "e") -> 48
({"n", "k"} -> "e") -> 48
({"n", "k"} -> "p") -> 224
({"n", "n"} -> "e") -> 96
({"n", "n"} -> "p") -> 224
({"n", "u"} -> "e") -> 48
({"n", "w"} -> "e") -> 48
({"n", "w"} -> "p") -> 1721

Now I need to be able to automatically, given gill size and spore print, predict edibility using that chart.

Importing raw data again:

```
mushroomCharacteristics = Import["csv file"];
```

```
(Debug) In[]:= twoVariables = Table[mushroomCharacteristics[[n, {9, 21}]] →
                  mushroomCharacteristics[[n, 1]], {n, 2, 8000}];
```

```
(Debug) In[ ]:= KeySort[Counts[twoVariables]]

(Debug) Out[ ]= <|({b, b} → e) → 35, ({b, h} → p) → 1584, ({b, k} → e) → 1600,
  ({b, n} → e) → 1636, ({b, o} → e) → 39, ({b, r} → p) → 72,
  ({b, w} → e) → 504, ({b, w} → p) → 34, ({b, y} → e) → 38, ({n, h} → e) → 48,
  ({n, k} → e) → 48, ({n, k} → p) → 224, ({n, n} → e) → 96, ({n, n} → p) → 224,
  ({n, u} → e) → 48, ({n, w} → e) → 48, ({n, w} → p) → 1721|>
```

```
(Debug) In[ ]:= sporePrintGillSize = Association@Keys@KeySort[Counts[twoVariables]]
```

Some of the classifications were incorrectly labeled, just because it took the first value of the key. For example, {“b”, “w”} was first labeled as poisonous instead of the much more common edible.

```
(Debug) In[ ]:= sporePrintGillSize = <|{"b", "b"} → "e", {"b", "h"} → "p", {"b", "k"} → "e",
   {"b", "n"} → "e", {"b", "o"} → "e", {"b", "r"} → "p", {"b", "w"} → "e", {"b", "y"} → "e",
   {"n", "h"} → "e", {"n", "k"} → "p", {"n", "n"} → "p", {"n", "u"} → "e", {"n", "w"} → "p"|>

(Debug) Out[ ]= <|{b, b} → e, {b, h} → p, {b, k} → e, {b, n} → e, {b, o} → e, {b, r} → p,
  {b, w} → e, {b, y} → e, {n, h} → e, {n, k} → p, {n, n} → p, {n, u} → e, {n, w} → p|>
```

Classify edibility:

```
(Debug) In[ ]:= predict[gillsize_, sporeprint_] := First@
    Values@KeyTake[sporePrintGillSize, {{ToString[gillsize], ToString[sporeprint]}}]
```

Example, using only the UCI dataset, predict edibility of a mushroom with a broad gill size (“b”) and chocolate spore print (“h”):

```
(Debug) In[ ]:= predict["b", "h"]

{"p"}
```

Now moving onto the image section.

Classifier functions from previous section:

```
(Debug) In[ ]:= extractGillSize = ClassifierFunction[ Input type: Image
                                                      Classes: broad, narrow
                                                      Method: LogisticRegression
                                                      Number of training examples: 2994 ];
```

```
(Debug) In[ ]:= extractSporePrint = ClassifierFunction[ Input type: Image
                                                        Number of classes: 12
                                                        Method: LogisticRegression
                                                        Number of training examples: 3198 ];
```

When creating the training datasets, it was easier, organizationally, to label the spore print as “white” or “chocolate”, and gill size “broad” or “narrow”, instead of the UCI labels of “w” or “h”, or “b” and “n”. Converting between the labels:

(Debug) In[ ]:=

```
replaceProperSporePrint[sporeprint_] :=
 Which[sporeprint == "Chocolate", "h", sporeprint == "White", "w", sporeprint == "Brown",
  "n", sporeprint == "Buff", "b", sporeprint == "Green", "r", sporeprint == "Pink",
  "o", sporeprint == "Purple", "u", sporeprint == "Yellow", "y"]
```

(Debug) In[ ]:=

```
replaceProperGillSize[gillsize_] :=
 Which[gillsize == "narrow", "n", gillsize == "broad", "b"]
```

Here is an example identifying the edibility of the edible king boletas mushroom :

(Debug) In[ ]:= extractSporePrint[ ]

(Debug) Out[ ]= Brown

(Debug) In[ ]:= extractGillSize[ ]

(Debug) Out[ ]= broad

```
replaceProperSporePrint["Brown"]
replaceProperGillSize["broad"]
```

(Debug) In[ ]:= predict["b", "n"]

```
{e}
```

Correctly predicts that the king boletas mushroom is edible, with a broad gill size and brown spore print!

Accuracy

To predict accuracy, I took 2 random images from each edible, inedible, and lethally poisonous mushroom. As a result, I had 196 edible mushrooms images, 66 inedible images, and 70 lethally poisonous mushroom images.

Random Training Images

(Debug) In[ ]:=

```
randomEdible = Flatten[#[[3 ;; 4]] & /@ edibleMushroomsBetter];
```

(Debug) In[ ]:=

```
randomInedible = Flatten[#[[3 ;; 4]] & /@ inedibleMushroomsBetter];
```

```
(Debug) In[ ]:= randomLethal = Flatten[#[[3 ;; 4]] & /@ lethallyPoisonousMushrooms];
```

Running predictions on random images using the two classifier functions that extract spore print and gill size from an image, and then using the UCI dataset chart to predict edibility.

```
(Debug) In[ ]:= ediblePredictionsBruh =
                 Table[gillSize1 = replaceProperGillSize[extractGillSize[randomEdible[[n]]]];
                  sporePrint1 = replaceProperSporePrint[extractSporePrint[randomEdible[[n]]]];
                  predict[gillSize1, sporePrint1], {n, Length@randomEdible}]

(Debug) Out[ ]= {e, e, e, e, p, e, p, e, e, e, e, e, e, p, p, p, e, e, p, p, p, p, p, p, p, p, e, e, p, p, p, p, e,
                 e, p, p, p, e, e, First[{}], e, p, p, p, p, p, First[{}], First[{}], p, e, e, e, e, e,
                 e, e, e, e, p, p, First[{}], First[{}], p, p, e, First[{}], p, e, p, e, p, p, p, e, p,
                 e, e, e, e, p, p, p, e, e, e, e, First[{}], p, p, p, First[{}], First[{}], First[{}],
                 First[{}], p, p, e, e, p, p, e, e, e, p, e, e, p, p, p, p, p, p, p, p, p, p, p, p, p,
                 First[{}], p, p, p, p, p, p, p, p, p, p, p, p, e, First[{}], e, e, p, e, p, p, p, p, p,
                 p, p, p, p, p, e, e, e, p, e, e, p, e, e, p, First[{}], First[{}], p, p, First[{}],
                 First[{}], e, e, p, p, p, p, First[{}], e, First[{}], First[{}], p, p, p, p, p, p,
                 p, p, p, e, p, p, p, p, p, e, p, p, First[{}], First[{}], First[{}], First[{}]}
```

Some of the combinations, such as a broad gill size and green spore print, were not in the UCI dataset, and that is why there is an error of “First[{}]” for some mushrooms.

```
(Debug) In[ ]:= Count[ediblePredictionsBruh, "e"]

(Debug) Out[ ]= 64

(Debug) In[ ]:= Count[ediblePredictionsBruh, "p"]

(Debug) Out[ ]= 108
```

There is a 37% accuracy in using gill size and spore print to predict edible mushrooms.

```
(Debug) In[ ]:= (64) / (64 + 108) // N

(Debug) Out[ ]= 0.372093
```

This is quite surprisingly low! One reason for this is that it is hard to distinguish between a chocolate and brown spore print. Personally, I believe spore print should be measure in RGB. Most of the edible mushrooms have a correctly classified broad gill size. However, a chocolate spore print paired with a broad gill size, in contrast to a brown spore print, results in a poisonous prediction. A possibility is that I incorrectly interpreted some of the WildFoodUK spore print descriptions as “chocolate” instead of “brown”, resulting in a higher percentage of images being incorrectly classified with the poisonous indicator.

Another possible explanation is that the image dataset that I’m using, WildFoodUK, does not contain that many entities for mushrooms from the Agarics family. Instead, it documents the mushrooms found in the UK. And the UCI dataset entries are all from the Agaricus family. This means that edible mushrooms in the UK do not highly resemble Agaricus mushrooms.

```
(Debug) In[◦]:= inediblePredictionsBruh =
                 Table[gillSize1 = replaceProperGillSize[extractGillSize[randomInedible[[n]]]];
                  sporePrint1 = replaceProperSporePrint[extractSporePrint[randomInedible[[n]]]];
                  predict[gillSize1, sporePrint1], {n, Length@randomInedible}]

(Debug) Out[◦]= {p, p, p, e, e, e, p, p, e, e, e, e, e, First[{}], p, p, e, p, p, First[{}], p, p, First[{}],
                 First[{}], e, e, p, p, e, e, p, p, p, p, p, p, p, p, p, p, First[{}], p, e, e, p, First[{}],
                 e, e, p, p, p, e, p, p, e, e, First[{}], First[{}], p, p, First[{}], First[{}], e, e, p, p}

(Debug) In[◦]:= Count[inediblePredictionsBruh, "e"]

(Debug) Out[◦]= 22

(Debug) In[◦]:= Count[inediblePredictionsBruh, "p"]

(Debug) Out[◦]= 34
```

70% accuracy for inedible mushrooms. Decent.

```
(Debug) In[◦]:= 39 / (17 + 39) // N

(Debug) Out[◦]= 0.696429

(Debug) In[◦]:= lethalPredictionsBruh =
                 Table[gillSize1 = replaceProperGillSize[extractGillSize[randomLethal[[n]]]];
                  sporePrint1 = replaceProperSporePrint[extractSporePrint[randomLethal[[n]]]];
                  predict[gillSize1, sporePrint1], {n, Length@randomLethal}]

(Debug) Out[◦]= {p, p, e, e, p, p, p, p, p, p, p, p, p, e, p, p, p, p, p, e, First[{}], p,
                 e, e, p, p, First[{}], First[{}], p, p, p, p, p, p, p, p, p, p, p, p, p, p, p,
                 p, e, e, p, p, p, p, p, p, p, p, p, p, p, e, p, p, p, p, p, p, e, p, p, p, p, p}

(Debug) In[◦]:= Count[lethalPredictionsBruh, "e"]
                Count[lethalPredictionsBruh, "p"]

(Debug) Out[◦]= 10

(Debug) Out[◦]= 57
```

And a stunning 87% accuracy for predicting lethally poisonous mushrooms!

```
(Debug) In[◦]:= 57 / (57 + 10) // N

(Debug) Out[◦]= 0.850746
```

After creating the training datasets, I noticed that a large amount of lethally poisonous mushrooms have a narrow gill size and white spore print color. This high accuracy is the result then, of the two classifier functions correctly and with a high accuracy identifying the white spore print and narrow gill size! Another explanation for this accuracy is that a much higher proportion of lethally poisonous mushrooms documented by WildFooUK belong to the Agaricus family, which the UCI dataset documents.

Some More Analysis

In the training datasets, inedible and lethally poisonous mushrooms were simply grouped together in the “poisonous” category. A 70% accuracy for inedible mushrooms signifies that 70% do have the spore print and gill size of poisonous mushrooms, while 30% have the spore print and gill size of edible mushrooms. Perhaps a different combination of two characteristics, such as spore print and odor, would result in a greater accuracy for inedible mushrooms - an outlet for future work.

Again, a possibility for the low accuracy of 37% for classifying may be a result of me incorrectly misinterpreting the the spore print listed on the WildFoodUK documentation as “chocolate” instead of “brown” when creating the training datasets. It’s difficult to tell the difference between a brown (edible) and chocolate (poisonous) spore print from a description.

And the solid 85% accuracy for lethally poisonous mushrooms is probably due to the fact that a white spore print and narrow gill size are easier to identify from an image.

But hey! It’s better to have a greater false negative than a false positive (it’s better to not eat an edible mushroom, than to eat a lethally poisonous mushroom).

Limitations

The image dataset from WildFoodUK does not primarily contain mushrooms from the Agaricus family, which the UCI dataset is based of. These regional differences may have lead to some discrepancies classifying mushrooms because mushrooms from northern Europe may have other characteristics that better determine edibility. However, WildFoodUK was the only dataset I found that allowed me to webscrape info from hundreds of websites. The layout of it was pertinent for collecting data for spore print and gill size, and as a result for the creation of the training datasets to extract features from images.

Another limitation was that gill size was not directly a part of the descriptions on WildFoodUK. To determine gill size, I used cap size, assuming that if the cap is narrow, than the gills would be narrow as well, and vice versa.

It was difficult to group some spore print descriptions from WildFookUK to match with the categorical entries of the UCI dataset. Some example descriptions were “Deep chocolate purple”, and “Olive brown”, both of which fit into multiple categories. A possible solution would be to measure spore print in RGB intervals.

Conclusion and Future Work

I learned a lot about using the Wolfram Language to process data, and structuring my code neatly to elegantly analyze the datasets.

Using spore print and gill size, there is a 37% for predicting edible mushrooms, 70% for predicting inedible mushrooms, and 85% accuracy for lethally poisonous mushrooms. That means there is total stunning accuracy of 64%. There is indeed a higher overall accuracy of using a combination of multiple datasets to predict edibility, rather than using one dataset (compared to the project Deep Shrooms which utilized a pure image dataset with a resulting 55% accuracy (Koivisto et al.)).

The main cause for the 37% accuracy for edible mushrooms is most likely incorrectly labeling some spore prints as “chocolate” instead of “brown” when creating the training datasets. On the other hand, the 85% accuracy for lethally poisonous mushrooms indicates that they are easier to spot from an image, with the most common lethally poisonous mushroom having a narrow gill size and a white spore print.

Future work includes on determining whether the combination of spore print and odor, or gill size or odor, are better predictors of edibility. WildFookUK does include descriptions for odor! However, many of the odors are described to be to as “mushroom” smell, which is not a category in the UCI dataset. If a mushroom smells like a mushroom, however, it would probably go into the UCI dataset category of “none”.

Moreover, user input could include the season to filter similar-looking mushrooms that are found in different seasons.

Furthermore, it’d be great if both the image and characteristics dataset contained mushrooms purely from the Agaricus family, to limit error. A possibility is isolating, from WildFoodUK, all the mushrooms which scientific names start with “Agaricus”. Last time I check though, there were only 13 mushrooms from the Agaricus family on WildFookUK, and I am not sure if that would be enough training data. But it’s worth a try!

The goal for this project was to provide a framework for utilizing and connecting multiple datasets. If in the future, if more datasets documenting mushrooms characteristics and images become available (especially if they document the same family of mushrooms), a similar method to this paper’s can be used to optimize the edibility prediction.

Have fun mushroom hunters! Even with the stunning 64% accuracy, it’s still interesting to see what type of gill size and spore print the image functions extract :)

References

Al-Mejibli I., and Abd D. (2017). Mushroom Diagnosis Assistance System Based On Machine Learning by Using Mobile Devices. Journal of AL-Quaisiyah for computer science and mathematics. http://www.qu.edu.iq/journalcm/index.php/journalcm/article/view/319/289

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA:

University of California, School of Information and Computer Science.

Koivisto, T., et al. (2017). Deep Shrooms: classifying mushroom images. Github. https://tuomonieminen.github.io/deep-shrooms/

Shields, T. (2021). The best apps for mushroom identification (and why a book is better). FreshCap Mushrooms. https://learn.freshcap.com/tips/mushroom-identification-app/

Wolfram Language & System. FeatureExtractor. Documentation Center. https://reference.wolfram.com/language/ref/FeatureExtractor.html