# Mushroom Identification and Machine learning

By: Ken Lo

# Table of contents



# Abstract

Mushroom foraging can act as both a wholesome activity and a source of food. However, it is important to be able to determine if a particular mushroom is edible or not. Although only 70-80 species of poisonous mushrooms are fatal if consumed, the deadly mushrooms tend to resemble some of the more familiar edible mushrooms in features. In turn, this project aims to help avoid poisonous features and successfully identify features that correlate to edibility. We will accomplish this by :

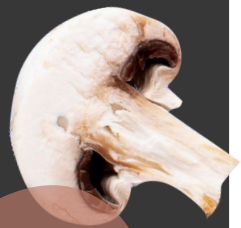
- Finding correlations between features and edibility
- Develop machine-learning models for prediction

## Mushroom Dataset

- Mushroom records drawn from the Audubon society field guide to North American Mushrooms (1981)
- The data set is comprised of 8124 hypothetical samples to around 23 species of gilled mushrooms in the Agaricus Lepiota Family. Features documented includes :

cap shape, cap surface, cap color, bruises, odor, gill attachment, gill spacing, gill size, gill color stalk shape, stalk root, stalk surface above/below ring, stalk color above/below ring, veil type, veil color, ring number, ring type, spore print color, population, habitat






1. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
2. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
3. cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r, pink=p,purple=u,red=e,white=w,yellow=y
4. bruises?: bruises=t,no=f
5. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s
6. gill-attachment: attached=a,descending=d,free=f,notched=n
7. gill-spacing: close=c,crowded=w,distant=d
8. gill-size: broad=b,narrow=n
9. 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
10. stalk-shape: enlarging=e,tapering=t
11. stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?
12. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
13. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
14. stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
15. stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
16. veil-type: partial=p,universal=u
17. veil-color: brown=n,orange=o,white=w,yellow=y
18. ring-number: none=n,one=o,two=t
19. ring-type: cobwebby=c,evanescent=e,flaring=f,large=l, none=n,pendant=p,sheathing=s,zone=z
20. spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r, orange=o,purple=u,white=w,yellow=y
21. population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y
22. habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

# 01 Feature Correlation

# Feature Correlation

## Aim:

Find correlation between features, directly compare their weight on the target edibility

## Prior Hypothesis:

There should exist features of higher importance, and other features would have some degree of dependence on another feature.

# Sub Aims

1

Data Exploration and
Data Preprocessing

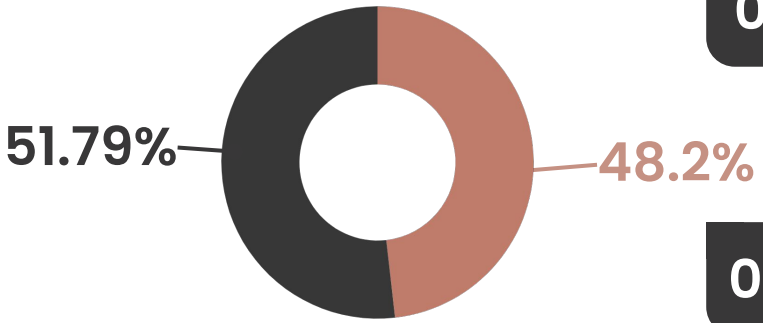
2

Visualizing Correlations

3

Feature Evaluation

# Data Exploration and Data Preprocessing




**01**

Looked at data distribution - Around 4208 edible samples, 3916 poisonous samples

**02**

Checked unique values in each feature. Dataset - purely categorical. Used label encoding for ordinal and binary data (type, bruises, ring-number, while the rest of the data was one-hot encoded

Categorical data :

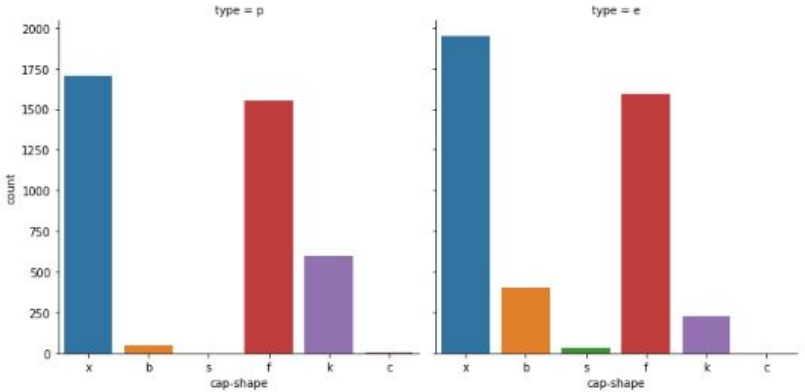
Ordinal data
Nominal data

**03**

Removed redundant features (veil-type for containing just one single value of p), replaced ? values (only in stalk-root)

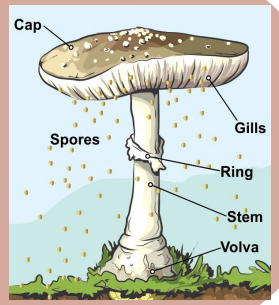
# Visualizing Correlations

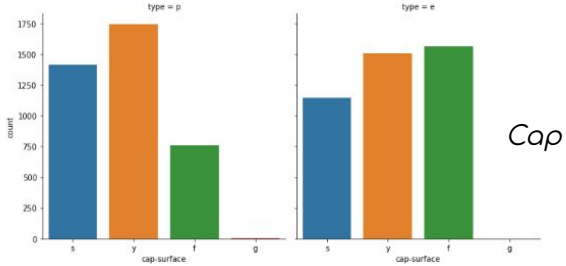
Poisonous



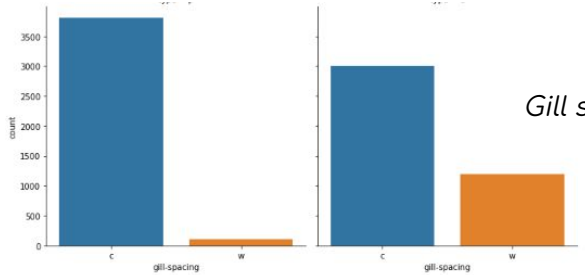

Edible

Distribution of values in a feature split between poisonous and edible



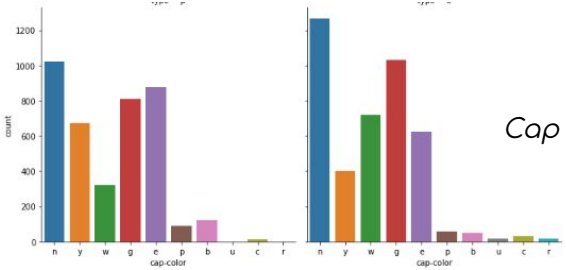




*Cap surface*

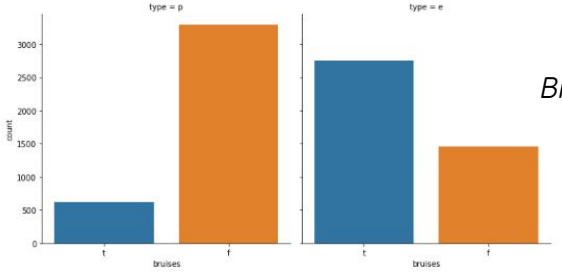



*Gill spacing*

bruised mushrooms - proportionately higher counts of edible samples
non-bruised mushrooms - proportionately higher counts of poisonous samples

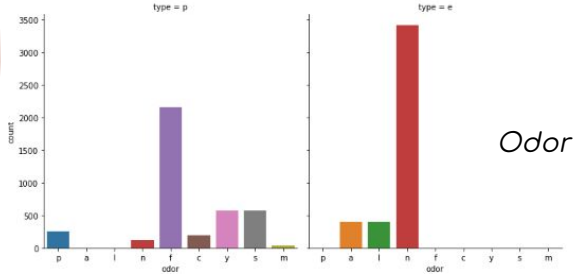



*Cap color*

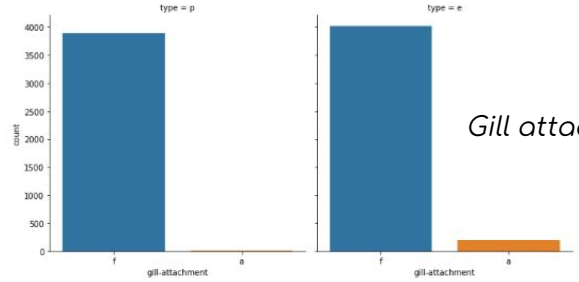



*Bruises*

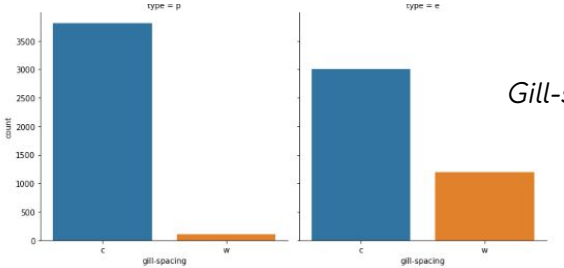
Odorless mushrooms - very high counts of edible samples
Foul smell mushrooms - high counts of poisonous samples



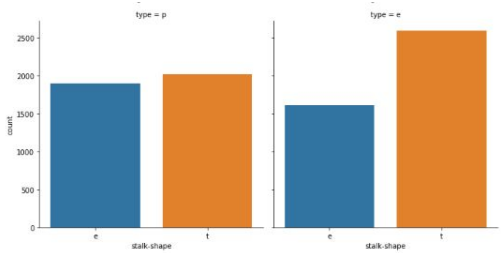

*Odor*




*Gill attachment*

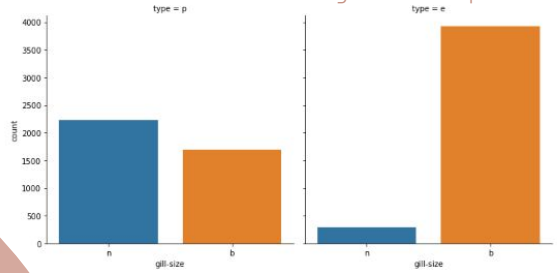



*Gill-spacing*

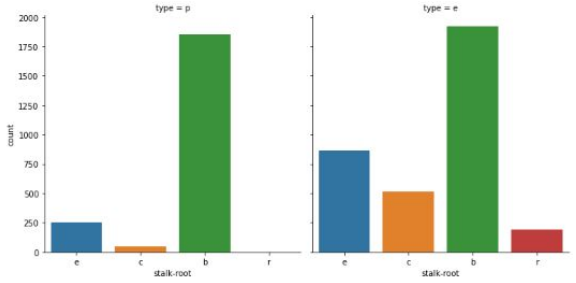



*Stalk-shape*

Gill size broad - high counts of edible samples
Gill size narrow - higher counts of poisonous samples than edible

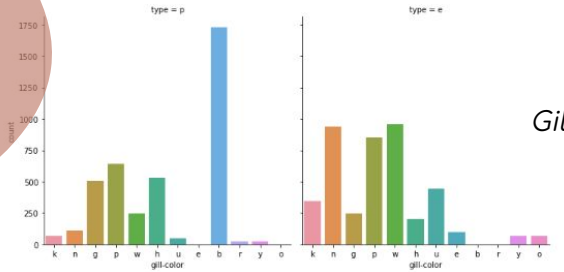



*Gill-size*

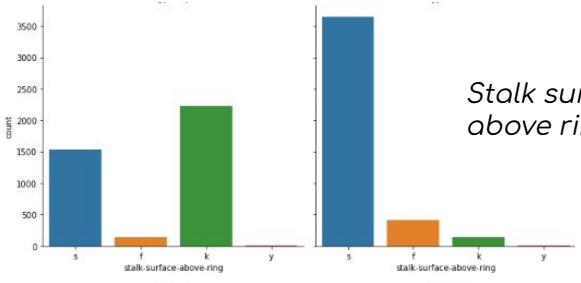



*Stalk-root*

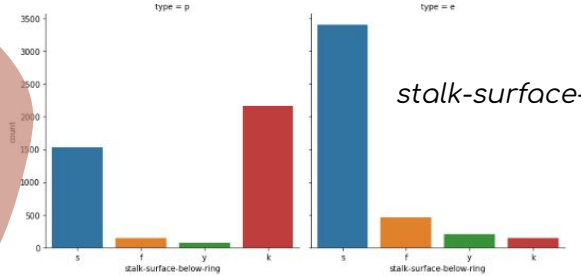
Gill color buff - one sided occurrences of poisonous samples



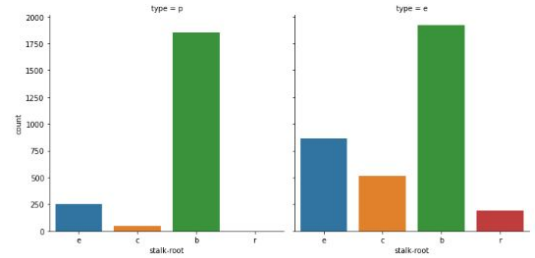

*Gill-color*



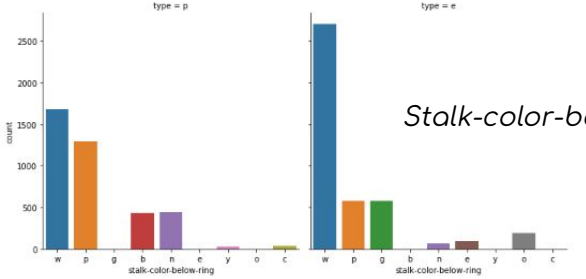

*Stalk surface above ring*



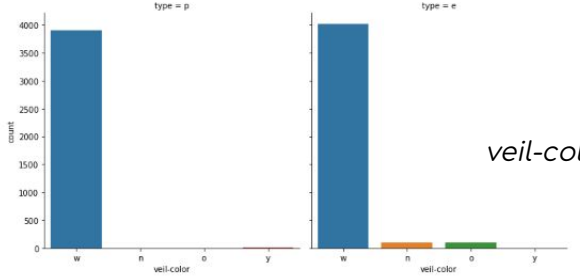

*stalk-surface-below-ring*




*Stalk-shape*



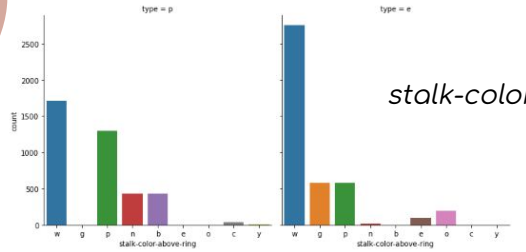

*Stalk-color-below ring*



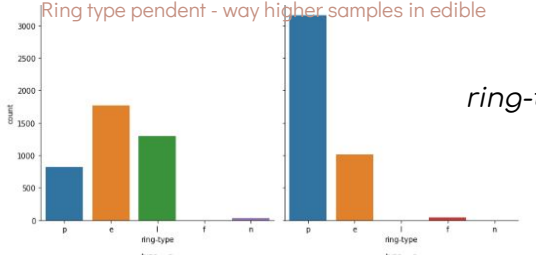

*veil-color*

Ring type large- zero counts in edible but around 1250 in poisonous

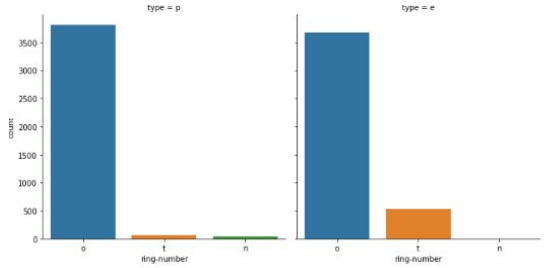
Ring type pendent - way higher samples in edible



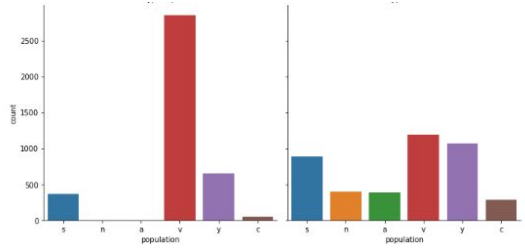

*stalk-color-above-ring*



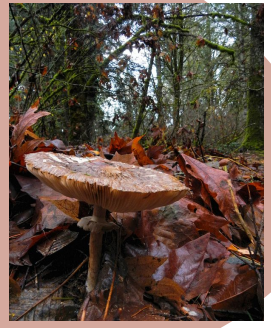

*ring-type*



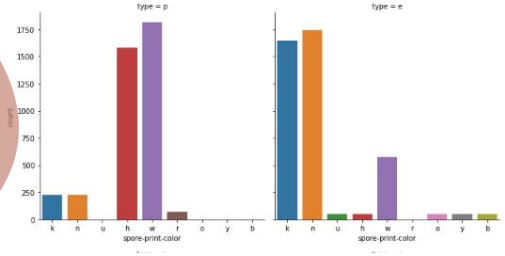

*ring-number*



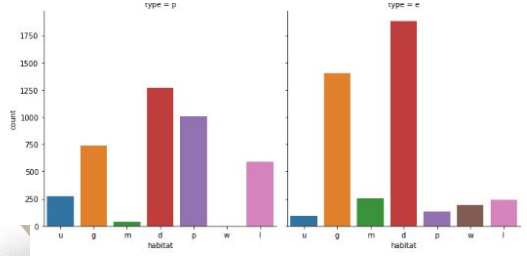

*population*



Spore print color - brown and black : very high counts of edible samples
Spore print color - chocolate/white: very high counts of poisonous samples




*spore-print-color*




*habitat*

# Count based percentages and Sample size

```
print("Odds odorless mushrooms are edible:", data[(data['odor'] == 'n') & (data['type'] == 'e')]['cap-shape'].count()/data[ (da
```

```
Odds odorless mushrooms are edible: 0.9659863945578231 %
Sample size being: 3528
```

```
print("Odds black spore print color mushrooms are edible:", data[(data['spore-print-color'] == 'k') & (data['type'] == 'e')]['c
```

```
Odds black spore print color mushrooms are edible: 0.8803418803418803 %
Sample size being: 1872
```

```
print("Odds brown spore print color mushrooms are edible:", data[(data['spore-print-color'] == 'n') & (data['type'] == 'e')]['c
```

```
Odds brown spore print color mushrooms are edible: 0.8861788617886179 %
Sample size being: 1968
```

```
print("Odds mushrooms with bruises are edible:", data[(data['bruises'] == 't') & (data['type'] == 'e')]['cap-shape'].count()/da
```

```
Odds mushrooms with bruises are edible: 0.8151658767772512 %
Sample size being: 3376
```

```
print("Odds broad gill mushrooms are edible:", data[(data['gill-size'] == 'b') & (data['type'] == 'e')]['cap-shape'].count()/da
```

```
Odds broad gill mushrooms are edible: 0.6985032074126871 %
Sample size being: 5612
```

```
print("Odds buff gill color mushrooms are edible:", data[(data['gill-color'] == 'b') & (data['type'] == 'e')]['cap-shape'].coun
```

```
Odds buff gill color mushrooms are edible: 0.0 %
Sample size being: 1728
```

```
print("Odds mushrooms without bruises are edible:", data[(data['bruises'] == 'f') & (data['type'] == 'e')]['cap-shape'].count()
```

```
Odds mushrooms without bruises are edible: 0.3066554338668913 %
Sample size being: 4748
```

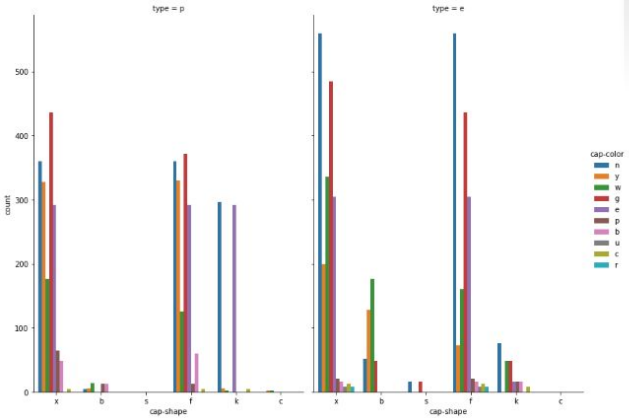
```
print("Odds chocolate spore print color mushrooms are edible:", data[(data['spore-print-color'] == 'h') & (data['type'] == 'e')
```

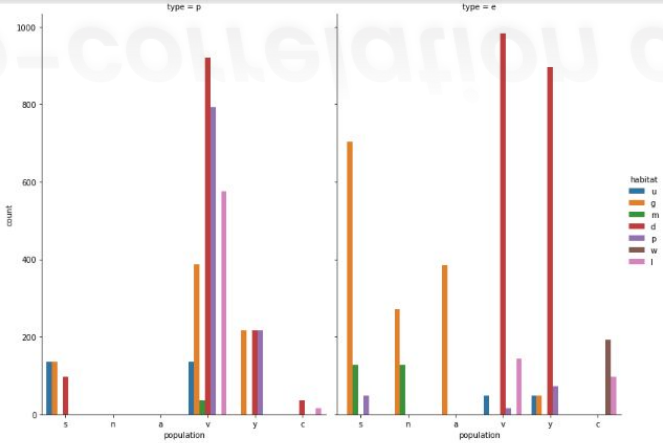
```
Odds chocolate spore print color mushrooms are edible: 0.029411764705882353 %
Sample size being: 1632
```

```
print("Odds white spore print color mushrooms are edible:", data[(data['spore-print-color'] == 'w') & (data['type'] == 'e')]['c
```

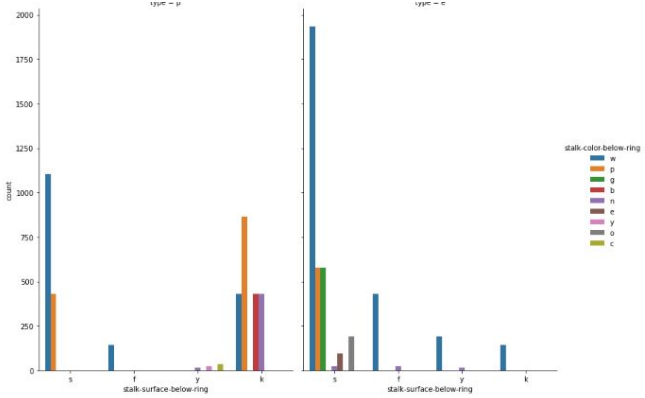
```
Odds white spore print color mushrooms are edible: 0.24120603015075376 %
Sample size being: 2388
```

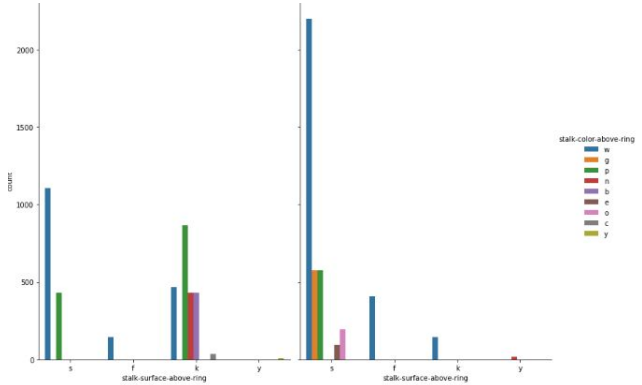
# Duo-correlation charts




Because these features have seemingly scattered charts for poisonous and edible distribution, they aren't useful on its on so I want to see if it's possible to find correlation by integrating them side by side

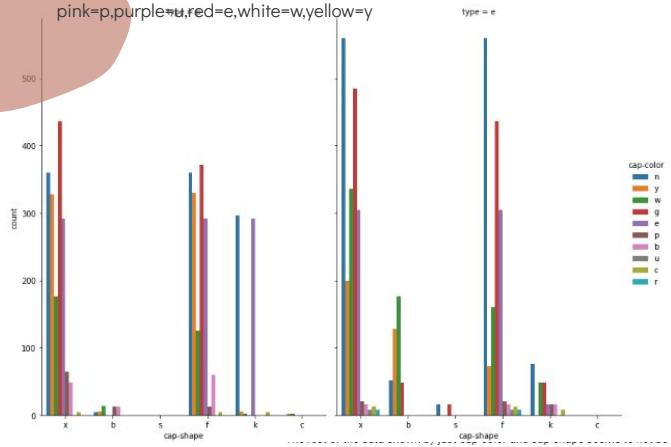


# Discoveries

These charts helped hone specifically on features that exist whilst another feature exist to show edibility within clusters

cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r,
pink=p,purple=u,red=e,white=w,yellow=y




```
[ ] print("Odds white cap bell shaped mushrooms are edible:", data[(data['cap-color

Odds white cap bell shaped mushrooms are edible: 0.9263157894736842 %
Sample size being: 190

[ ] print("Odds bell shaped mushrooms are edible:", data[(data['cap-shape'] == 'b')

Odds bell shaped mushrooms are edible: 0.8938053097345132 %
Sample size being: 452

[ ] print("Odds knobbed shaped mushrooms are edible:", data[(data['cap-shape'] == '

Odds knobbed shaped mushrooms are edible: 0.2753623188405797 %
Sample size being: 828

[ ] print("Odds yellow cap mushrooms are edible:", data[(data['cap-color'] == 'y')

Odds yellow cap mushrooms are edible: 0.373134328358209 %
Sample size being: 1072
```

Most of the data shows that knobbed shaped mushrooms should be avoided as well as yellow mushrooms in general, especially if they are flat shaped.

We can also see that bell shaped mushrooms have a very good chance of being edible, especially the white colored ones.

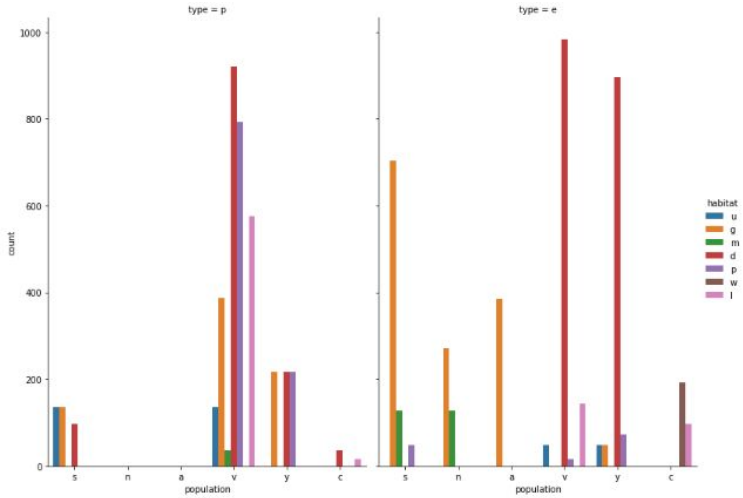
## Odds white cap bell shaped mushrooms are edible: 92%

**Odds bellshaped mushroom in general are edible : 89.3%**

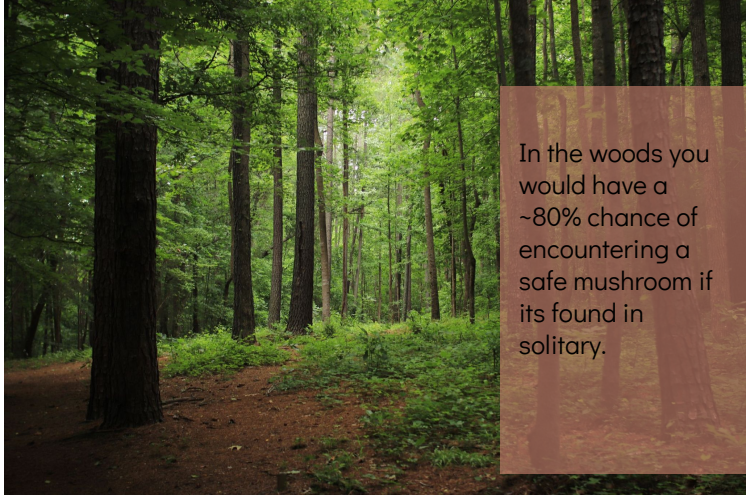
**Odds knobbed shaped mushrooms are edible : 27.5%**

**Odds yellow cap mushrooms are edible : 37.3%**

Seems familiar?




```
print("Odds for edible scattered mushrooms in grass habitat :", data[(data['habitat'] ...

Odds for edible scattered mushrooms in grass habitat : 0.8380952380952381 %
Sample size being: 840

print("Odds for edible numerous mushrooms in grass habitat :", data[(data['habitat'] ==

Odds for edible numerous mushrooms in grass habitat : 1.0 %
Sample size being: 272

print("Odds for edible abudant mushrooms in grass habitat :", data[(data['habitat'] ==

Odds for edible abudant mushrooms in grass habitat : 1.0 %
Sample size being: 304

print("Odds for edible several mushrooms in grass habitat :", data[(data['habitat'] ==

Odds for edible several mushrooms in grass habitat : 0.0 %
Sample size being: 388

print("Odds for edible solitary mushrooms in grass habitat :", data[(data['habitat'] ==

Odds for edible solitary mushrooms in grass habitat : 0.18181818181818182 %
Sample size being: 264

print("Odds solitary mushroom is edible in the woods:", data[(data['population'] == 'y

Odds solitary mushroom is edible in the woods: 0.8057553956834532 %
Sample size being: 1112

print("Odds mushroom is edible on paths:", data[(data['habitat'] == 'p') & (data['type

Odds mushroom is edible on paths: 0.11888111888111888 %
Sample size being: 1144

print("Odds mushroom is edible on leaves:", data[(data['habitat'] == 'l') & (data['typ

Odds mushroom is edible on leaves: 0.28846153846153844 %
Sample size being: 832
```



In the woods you would have a ~80% chance of encountering a safe mushroom if its found in solitary.

habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

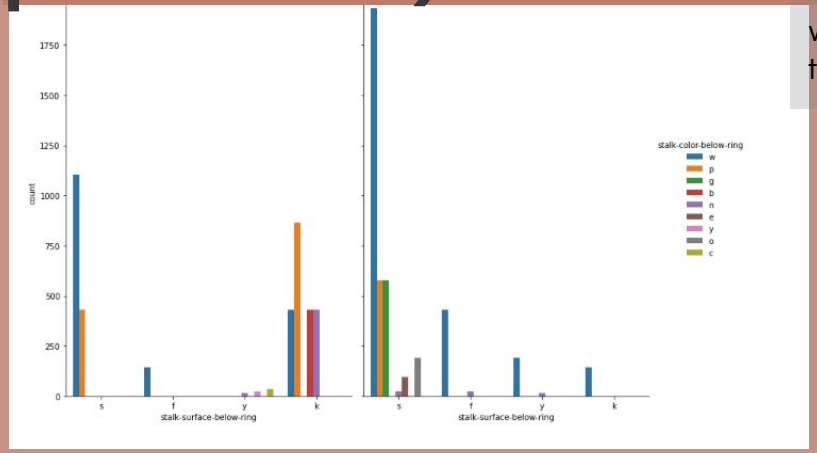
population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y

In grassy habitats, scattered, numerous, abundant mushrooms will be quite safe, the more you see the better! As opposed to seeing them in several or solitary population.

## Odds for scattered mushrooms in a grass habitat are edible: 83%

Avoid mushrooms found on leaves and paths altogether.

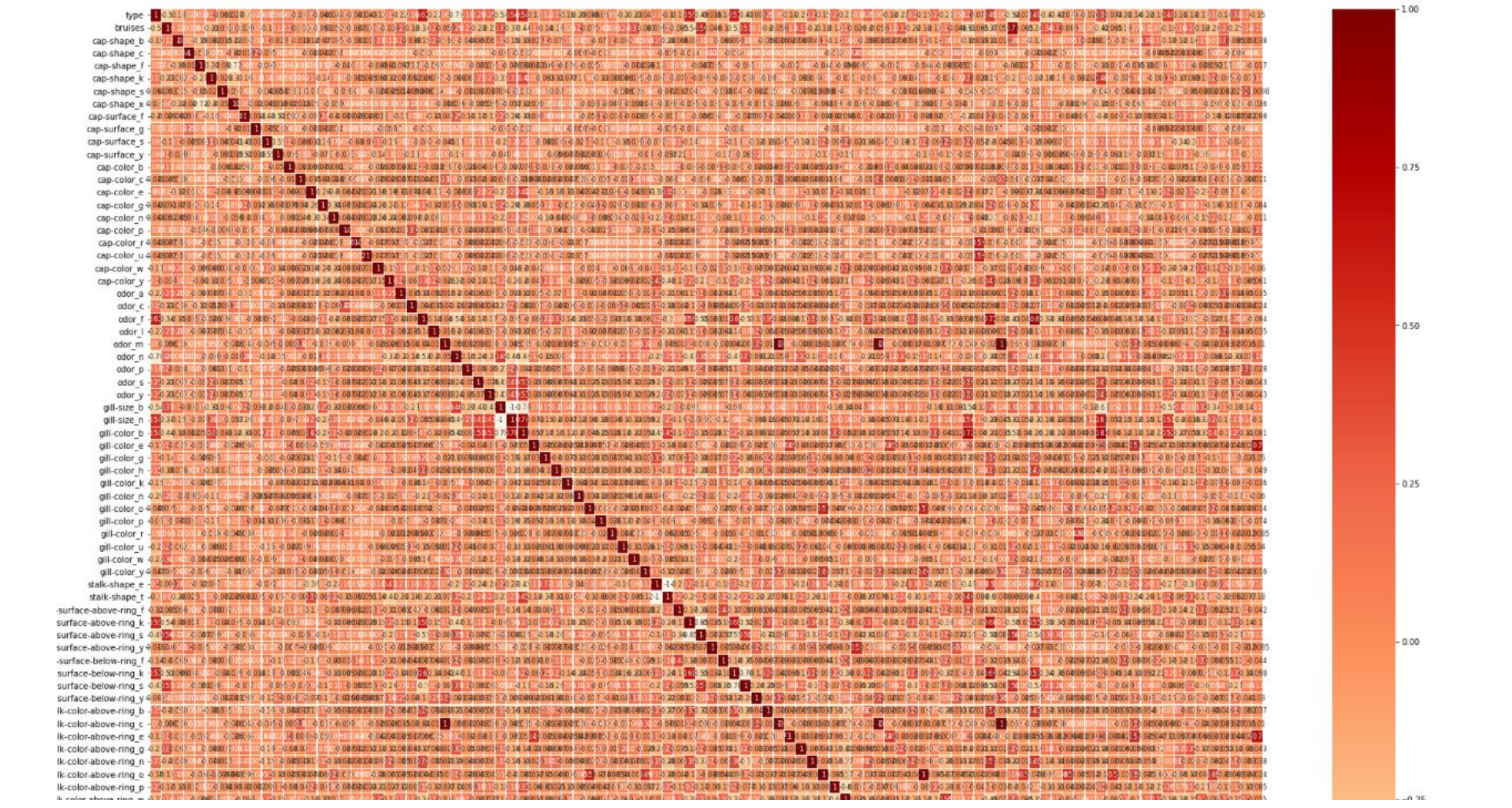
# Odds silky textured mushroom on the stalk below ring are edible : 0.0625% (aka its poisonous)




white fly agaric , silky textured, very poisonous


```
[35] print("Odds silky mushroom is edible :", data[(data['stalk-su

     Odds silky mushroom is edible : 0.0625 %
     Sample size being: 2304

[36] print("Odds smooth mushroom is edible :", data[(data['stalk-s

     Odds smooth mushroom is edible : 0.6888168557536467 %
     Sample size being: 4936

[37] print("Odds fibrous mushroom is edible :", data[(data['stalk-

     Odds fibrous mushroom is edible : 0.76 %
     Sample size being: 600

[38] print("Odds scaly mushroom is edible :", data[(data['stalk-su

     Odds scaly mushroom is edible : 0.7323943661971831 %
     Sample size being: 284
```

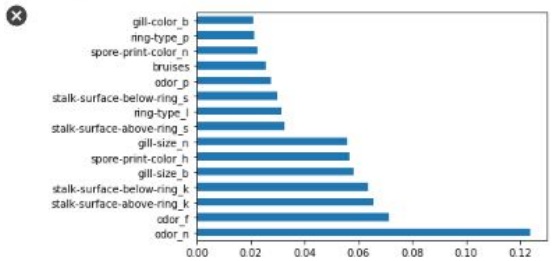
stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y


type
bruises
cap-shape_b
cap-shape_c
cap-shape_f
cap-shape_k
cap-shape_s
cap-shape_x
cap-surface_f
cap-surface_g
cap-surface_s
cap-surface_y
cap-color_b
cap-color_c
cap-color_e
cap-color_g
cap-color_n
cap-color_p
cap-color_r
cap-color_u
cap-color_w
cap-color_y
odor_a
odor_c
odor_f
odor_l
odor_m
odor_n
odor_p
odor_s
odor_y
gill-size_b
gill-size_n
gill-color_b
gill-color_e
gill-color_g
gill-color_h
gill-color_k
gill-color_n
gill-color_o
gill-color_p
gill-color_r
gill-color_u
gill-color_w
gill-color_y
stalk-shape_e
stalk-shape_t
-surface-above-ring_f
surface-above-ring_k
surface-above-ring_s
surface-above-ring_y
-surface-below-ring_f
surface-below-ring_k
surface-below-ring_s
surface-below-ring_y
lk-color-above-ring_b
lk-color-above-ring_c
lk-color-above-ring_e
lk-color-above-ring_g
lk-color-above-ring_n
lk-color-above-ring_o
lk-color-above-ring_p
1.00
0.75
0.50
0.25
0.00
-0.25

To decrease dimensionality  I removed data with very similar distributions (veil-color, ring-number, gill-spacing)

# 02 Model Training

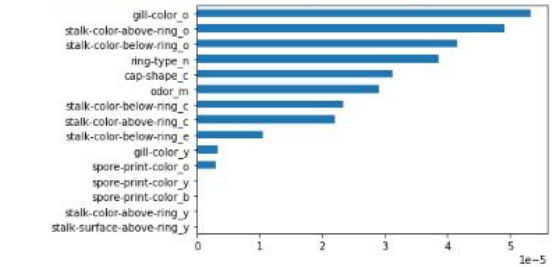
# Finding feature importance




The top 15 most major features to look for in determining edibility.

```
[47] feat_importances.nsmallest(15).plot(kind='barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f29693b0ee0>
```




Taking out the smallest 14 in feature importance. These are data that are either too irrelevant in the grandscheme or have too little sample data to make any reliable judgement.

Further cropping of data

Features that were previously examined to be skewed to one of either edible or poisonous are now given an exact correlation value. Whether a mushroom has an odor, which as a feature has a -79% correlation which almost by itself can tell you that a mushroom is edible. On the other hand, if there were to be an odor and that odor happened to be of a foul smell, which has a 62% correlation, there is a really high chance of being a poisonous mushroom. A mushroom's gill size being broad or narrow seems to heavily correlate to edibility, with narrow gills having a 54% correlation and broad gills having a -54% correlation. Other notable features are stalk-surface-above-ring-k (silky surface), and spore-print-color-h (color a mushroom’s spores produce) both being a substantial indicators for poisonous mushrooms.

# Model accuracy

| | Naive Bayes | Decision Tree | Random Forest |
|---|---|---|---|
| Accuracy | 100% | 100% | 100% |
| Standard deviation | 0 | 0 | 0 |

# Naive Bayes Model

```
[51] nb = GaussianNB()
     nb.fit(X_train,Y_train.values.ravel())
     nb_y_pred=nb.predict(X_test)
     nb_metrics = p1_metrics(Y_test.values.ravel(), nb_y_pred, include_cm=False)
     print("NB Training Metrics")
     for metric in nb_metrics:
         print(metric+":",nb_metrics[metric])
```

```
NB Training Metrics
auc: 1.0
f1: 1.0
accuracy: 1.0
sensitivity: 1.0
specificity: 1.0
precision: 1.0
```

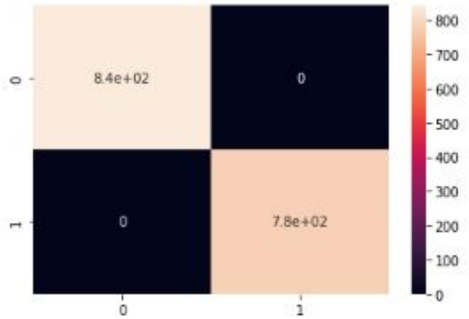
Perfect Score on the Naive Bayes Model, trying cross validation with kfold 20 splits

```
[52] scores = cross_val_score(nb, X_train, Y_train.values.ravel(), cv = kf)

     print("cross-validation mean score of %0.2f with a standard deviation of %0.2f" % (scores.mean(), scores.std()))
```

```
cross-validation mean score of 1.00 with a standard deviation of 0.00
```

```
predictNB = nb.predict(X_test)
cm = confusion_matrix(predictNB, Y_test)
sns.heatmap(cm, annot=True)

print(classification_report(predictNB, Y_test))
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 843 |
| 1 | 1.00 | 1.00 | 1.00 | 782 |
| accuracy | | | 1.00 | 1625 |
| macro avg | 1.00 | 1.00 | 1.00 | 1625 |
| weighted avg | 1.00 | 1.00 | 1.00 | 1625 |



# Decision Tree Model

```
decisionTree = DecisionTreeClassifier(random_state = 23800756)
decisionTree.fit(X_train,Y_train.values.ravel())
decisionTree_y_pred=decisionTree.predict(X_test)
decisionTree_metrics = p1_metrics(Y_test.values.ravel(), decisionTree_y_pred, include_cm=False)
print("Decision Tree Training Metrics")
for metric in nb_metrics:
    print(metric+":",nb_metrics[metric])
```

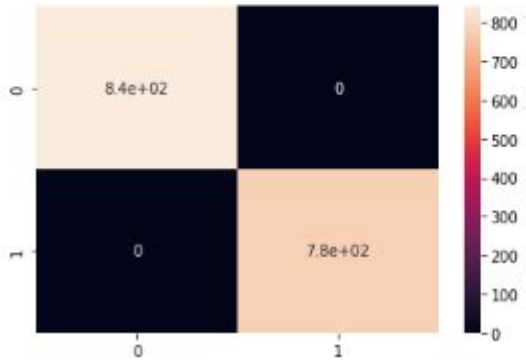
```
Decision Tree Training Metrics
auc: 1.0
f1: 1.0
accuracy: 1.0
sensitivity: 1.0
specificity: 1.0
precision: 1.0
```

```
scores = cross_val_score(decisionTree, X_train, Y_train.values.ravel(), cv = kf)

print("cross validation mean score of %0.2f with a standard deviation of %0.2f" % (scores.mean(), scores.std()
```

```
cross validation mean score of 1.00 with a standard deviation of 0.00
```

```
predictdt = decisionTree.predict(X_test)
cm = confusion_matrix(predictdt, Y_test)
sns.heatmap(cm, annot=True)

print(classification_report(predictdt, Y_test))
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 843 |
| 1 | 1.00 | 1.00 | 1.00 | 782 |
| accuracy | | | 1.00 | 1625 |
| macro avg | 1.00 | 1.00 | 1.00 | 1625 |
| weighted avg | 1.00 | 1.00 | 1.00 | 1625 |



# Random Forest Model

```
rf = RandomForestClassifier(random_state = 23800756)
rf.fit(X_train,Y_train.values.ravel())
rf_y_pred=rf.predict(X_test)
rf_metrics = p1_metrics(Y_test.values.ravel(), rf_y_pred, include_cm=False)
print("RandomForest Training Metrics")
for metric in nb_metrics:
    print(metric+":",nb_metrics[metric])
```

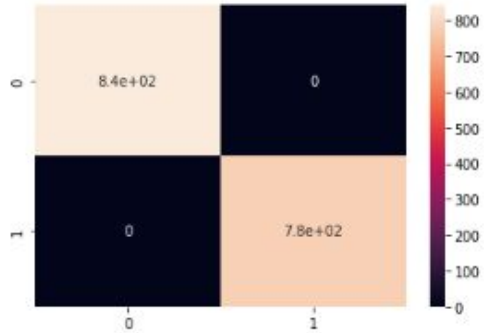
```
RandomForest Training Metrics
auc: 1.0
f1: 1.0
accuracy: 1.0
sensitivity: 1.0
specificity: 1.0
precision: 1.0
```

```
scores = cross_val_score(rf,X_train, Y_train.values.ravel(), cv = kf)

print("cross validation mean score of %0.2f with a standard deviation of %0.2f" % (scores.mean(), scores.std()))
```

```
cross validation mean score of 1.00 with a standard deviation of 0.00
```

```
predictrf = rf.predict(X_test)
cm = confusion_matrix(predictrf, Y_test)
sns.heatmap(cm, annot=True)

print(classification_report(predictrf, Y_test))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       843
           1       1.00      1.00      1.00       782

    accuracy                           1.00      1625
   macro avg       1.00      1.00      1.00      1625
weighted avg       1.00      1.00      1.00      1625
```



# 03 Conclusions

# What does this mean?

Having 100% accuracy across the board made me skeptical of my process, but after thorough checking:

Seems that the model just has a very clear indication of edibility as all 3 classification had perfect scores. This means the models are able to look at the existence(or non-existence) of the top few correlating features with conjunction with other high correlating feature to perfectly predict edibility.

Since all 3 models had 100% accuracy we can say at the very least if a mushroom were to have the exact features of a sample, there would be a very high chance that our model would be able to identify the edibility correctly. Of course, we have to account for the 8124 sample size which is relatively small, which means that there could be a mushroom out there with the exact traits but of the opposite type. The feature correlation study is also very helpful for casual foragers without access to the models, as we showed quite a few notably significant features that show edibility.