Document

May 2, 2025

0.0.1 *Muhammad Anas*

0.0.2 *1001805539*

0.0.3 *Dr. Amir Farbin*

0.0.4 *Python II DATA-3402*

0.0.5 *Mushroom Dataset*

0.0.6 *https://www.kaggle.com/datasets/uciml/mushroom-classification/data*

Time: Donated to UCI ML April 27, 1987.

This dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom drawn from The Audubon Society Field Guide to North American Mushrooms (1981). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like "leaflets three, let it be'' for Poisonous Oak and Ivy.

0.0.7 Additional Variable Information

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

0.0.8 Objectives:

- Understand the dataset.
- Preprocess the data.
- Find best features.
- Evaluate different ML models
- Select the best model
- Provide reasoning for the selection of model.
-

0.1 Understanding the data set.

[1]:
```
from functions import *


df = read_csv("mushrooms.csv")
store_information(df)

print_file_content("dfhead.txt")
print_file_content("dfdescribe.txt")
print_file_content("dfinformatin.txt")
```

```
------------------------------Dataframe Head------------------------------

  class cap-shape cap-surface cap-color bruises odor gill-attachment gill-
spacing gill-size gill-color stalk-shape stalk-root stalk-surface-above-ring
stalk-surface-below-ring stalk-color-above-ring stalk-color-below-ring veil-type
veil-color ring-number ring-type spore-print-color population habitat
0     p         x           s         n       t    p               f
c         n          k           e          e                         s
s                       w                         w         p          w           o
p                 k           s       u
1     e         x           s         y       t    a               f
c         b          k           e          c                         s
s                       w                         w         p          w           o
p                 n           n       g
2     e         b           s         w       t    l               f
c         b          n           e          c                         s
s                       w                         w         p          w           o
p                 n           n       m
```

```
3     p         x           y         w       t    p              f
c         n           n           e         e                      s
s                       w                     w         p          w           o
p                 k           s       u
4     e         x           s         g       f    n              f
w         b           k           t         e                      s
s                       w                     w         p          w           o
e                 n           a       g
-------------------------------------------------------------------------------
------------------------------Dataframe
Description------------------------------

       class cap-shape cap-surface cap-color bruises  odor gill-attachment gill-
spacing gill-size gill-color stalk-shape stalk-root stalk-surface-above-ring
stalk-surface-below-ring stalk-color-above-ring stalk-color-below-ring veil-type
veil-color ring-number ring-type spore-print-color population habitat
count   8124      8124        8124      8124    8124  8124            8124
8124      8124       8124        8124       8124                    8124
8124                    8124                    8124      8124       8124
8124      8124              8124       8124    8124
unique     2         6           4        10       2     9              2
2         2         12           2          5                        4
4                       9                     9         1          4           3
5                 9          6       7
top        e         x           y         n       f     n              f
c         b           b           t          b                       s
s                       w                     w         p          w           o
p                 w          v       d
freq    4208      3656        3244      2284    4748  3528            7914
6812      5612       1728        4608       3776                    5176
4936                    4464                    4384      8124       7924
7488      3968              2388       4040    3148
-------------------------------------------------------------------------------
------------------------------Dataframe
Information------------------------------

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8124 entries, 0 to 8123
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   class                     8124 non-null   object
 1   cap-shape                 8124 non-null   object
 2   cap-surface               8124 non-null   object
 3   cap-color                 8124 non-null   object
 4   bruises                   8124 non-null   object
 5   odor                      8124 non-null   object
 6   gill-attachment           8124 non-null   object
```

```
 7   gill-spacing              8124 non-null   object
 8   gill-size                 8124 non-null   object
 9   gill-color                8124 non-null   object
 10  stalk-shape               8124 non-null   object
 11  stalk-root                8124 non-null   object
 12  stalk-surface-above-ring  8124 non-null   object
 13  stalk-surface-below-ring  8124 non-null   object
 14  stalk-color-above-ring    8124 non-null   object
 15  stalk-color-below-ring    8124 non-null   object
 16  veil-type                 8124 non-null   object
 17  veil-color                8124 non-null   object
 18  ring-number               8124 non-null   object
 19  ring-type                 8124 non-null   object
 20  spore-print-color         8124 non-null   object
 21  population                8124 non-null   object
 22  habitat                   8124 non-null   object
dtypes: object(23)
memory usage: 1.4+ MB
```

---

[2]:
```
## No missing values
missing_val = df.isnull().sum()
print(missing_val)
```

```
class                     0
cap-shape                 0
cap-surface               0
cap-color                 0
bruises                   0
odor                      0
gill-attachment           0
gill-spacing              0
gill-size                 0
gill-color                0
stalk-shape               0
stalk-root                0
stalk-surface-above-ring  0
stalk-surface-below-ring  0
stalk-color-above-ring    0
stalk-color-below-ring    0
veil-type                 0
veil-color                0
ring-number               0
ring-type                 0
spore-print-color         0
population                0
habitat                   0
```

```
dtype: int64
```

---

[3]:
```
## printing all the unique variables present in each features/columns
printing_column(df)
```

```
class unique values:
['p' 'e']

cap-shape unique values:
['x' 'b' 's' 'f' 'k' 'c']

cap-surface unique values:
['s' 'y' 'f' 'g']

cap-color unique values:
['n' 'y' 'w' 'g' 'e' 'p' 'b' 'u' 'c' 'r']

bruises unique values:
['t' 'f']

odor unique values:
['p' 'a' 'l' 'n' 'f' 'c' 'y' 's' 'm']

gill-attachment unique values:
['f' 'a']

gill-spacing unique values:
['c' 'w']

gill-size unique values:
['n' 'b']

gill-color unique values:
['k' 'n' 'g' 'p' 'w' 'h' 'u' 'e' 'b' 'r' 'y' 'o']

stalk-shape unique values:
['e' 't']

stalk-root unique values:
['e' 'c' 'b' 'r' '?']

stalk-surface-above-ring unique values:
['s' 'f' 'k' 'y']

stalk-surface-below-ring unique values:
['s' 'f' 'y' 'k']
```

```
stalk-color-above-ring unique values:
['w' 'g' 'p' 'n' 'b' 'e' 'o' 'c' 'y']

stalk-color-below-ring unique values:
['w' 'p' 'g' 'b' 'n' 'e' 'y' 'o' 'c']

veil-type unique values:
['p']

veil-color unique values:
['w' 'n' 'o' 'y']

ring-number unique values:
['o' 't' 'n']

ring-type unique values:
['p' 'e' 'l' 'f' 'n']

spore-print-color unique values:
['k' 'n' 'u' 'h' 'w' 'r' 'o' 'y' 'b']

population unique values:
['s' 'n' 'a' 'v' 'y' 'c']

habitat unique values:
['u' 'g' 'm' 'd' 'p' 'w' 'l']
```

0.2

We can observe there is one missing value. This was always mentioned in the description. Let us see the count of that value.

[4]:
```
invalid = ['?']
show_invalid_entries(df, 'stalk-root', invalid)
```

```
Number of invalid stalk-root entries: 2480
stalk-root
?    2480
Name: count, dtype: int64
```

Dropping column veil-type because it only has one value as can be seen above, however dropping stalk-root column because it has a lot of missing values. It will make a difference if we replace it by mode value.

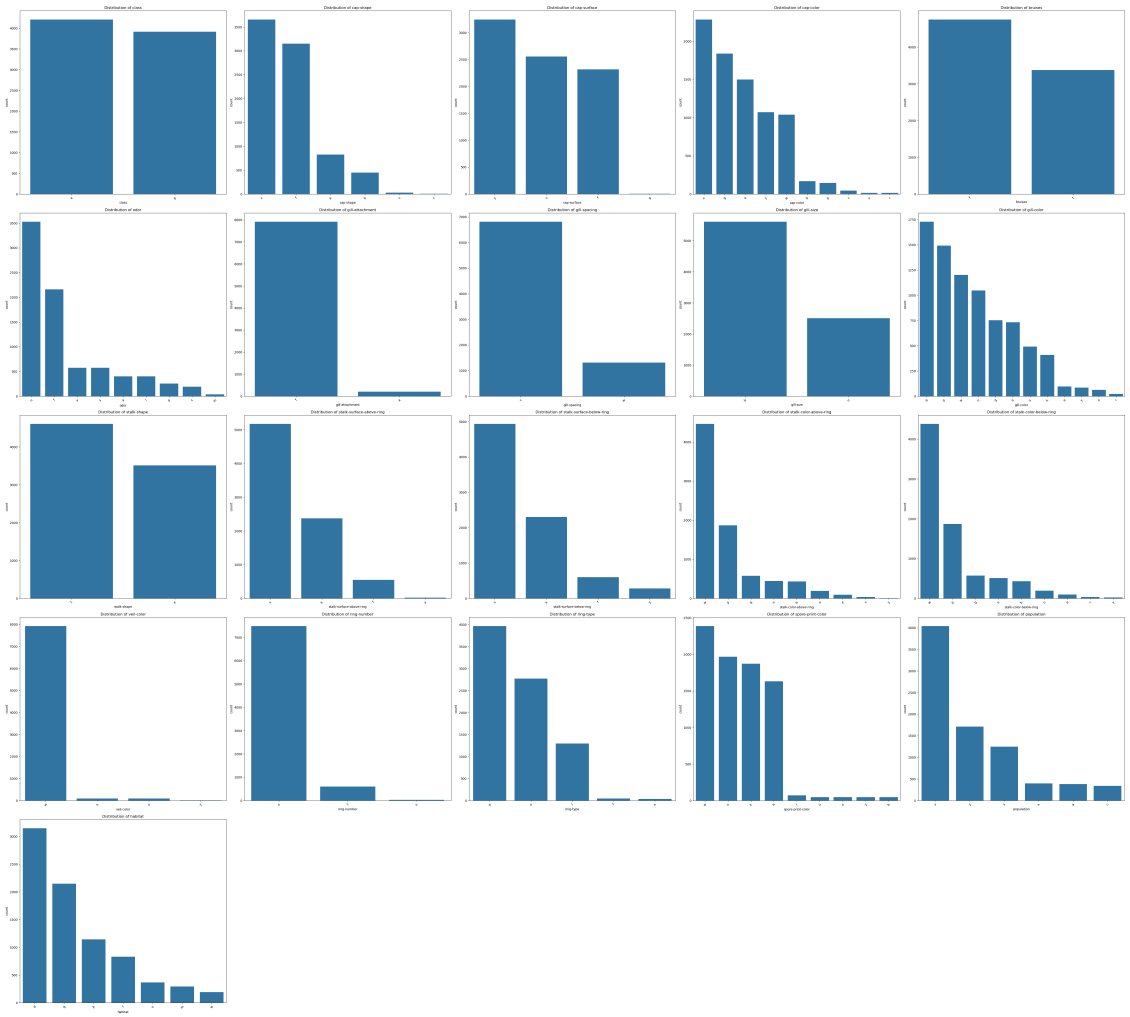
[5]:
```
df = df.drop(columns= 'veil-type')
df = df.drop(columns= 'stalk-root')
```

From the above information we can observe that this data set contains 23 columns and 8124 rows (records). By inspecting the information we observe that all the columns are categorical and none of them are numerical. Just looking at the information instance the observation dictates no missing value in any column. But we cannot assume much just by looking at this. We need more information for this.

0.2.1 Data Visualization:

Trying to get plots for initial visualization.

[6]:
```
get_plots(df)
```



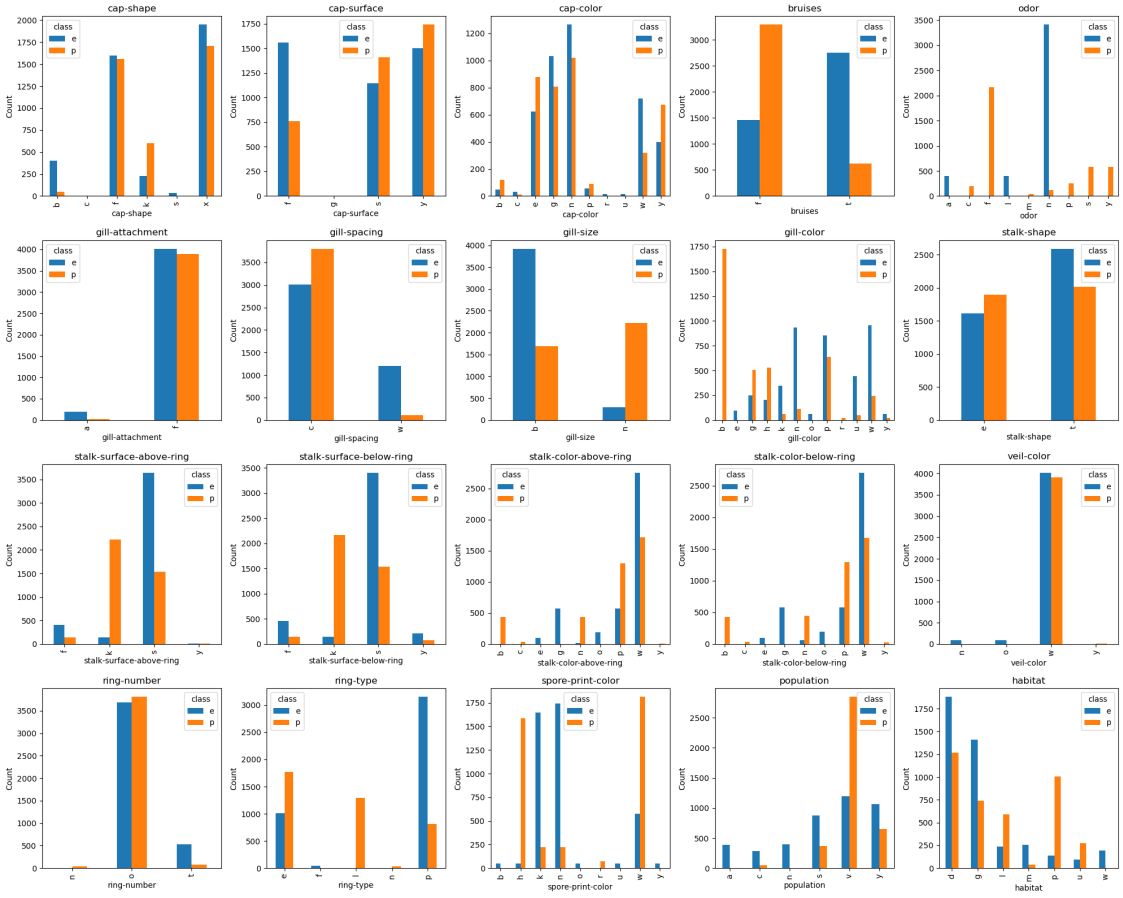
0.2.2 Observation:

Initial plots showcase brief information about the data set. We see that our target variable has only two values, which directs us towards binary classification. Column "veil-type" shows that there is

only one value in there. We can make an assumption that this value will have no real impact on both classes since it's a common trait shared among both the classes.

Plotting target column (class) against all the features to visualize the distribution of class e (edible) and p (poisnous).

[7]:
```
## calling function to visualize the distribution.
targetVfeatures('class', df)
```



[8]:
```
## printing the above plots as a tabular form.
## Representing each class in perecentages.
print_target_distributions('class', df)
```

```
Feature: cap-shape

 cap-shape          e        p

 b              89.38    10.62
```

c	0	100
f	50.63	49.37
k	27.54	72.46
s	100	0
x	53.28	46.72

Feature: cap-surface

cap-surface	e	p
f	67.24	32.76
g	0	100
s	44.76	55.24
y	46.36	53.64

Feature: cap-color

cap-color	e	p
b	28.57	71.43
c	72.73	27.27
e	41.6	58.4
g	56.09	43.91
n	55.34	44.66
p	38.89	61.11
r	100	0
u	100	0
w	69.23	30.77
y	37.31	62.69

```
Feature: bruises

 bruises        e      p

 f          30.67  69.33

 t          81.52  18.48


Feature: odor

 odor         e      p

 a          100      0

 c            0    100

 f            0    100

 l          100      0

 m            0    100

 n         96.6    3.4

 p            0    100

 s            0    100

 y            0    100


Feature: gill-attachment

 gill-attachment         e      p

 a                   91.43   8.57

 f                   50.75  49.25


Feature: gill-spacing

 gill-spacing         e      p

 c                44.16  55.84
```

```
  w               91.46    8.54

Feature: gill-size

  gill-size         e       p

  b             69.85   30.15

  n             11.46   88.54

Feature: gill-color

  gill-color          e        p

  b                 0      100

  e               100        0

  g                32.98    67.02

  h                27.87    72.13

  k                84.31    15.69

  n                89.31    10.69

  o               100        0

  p                57.1     42.9

  r                 0      100

  u                90.24     9.76

  w                79.53    20.47

  y                74.42    25.58

Feature: stalk-shape

  stalk-shape         e       p

  e               45.96   54.04
```

```
  t               56.25   43.75


Feature: stalk-surface-above-ring

  stalk-surface-above-ring        e      p

  f                           73.91   26.09

  k                            6.07   93.93

  s                           70.32   29.68

  y                           66.67   33.33


Feature: stalk-surface-below-ring

  stalk-surface-below-ring        e      p

  f                           76      24

  k                            6.25   93.75

  s                           68.88   31.12

  y                           73.24   26.76


Feature: stalk-color-above-ring

  stalk-color-above-ring          e       p

  b                            0      100

  c                            0      100

  e                          100        0

  g                          100        0

  n                            3.57    96.43

  o                          100        0

  p                           30.77    69.23

  w                           61.65    38.35
```

```
  y                           0      100
```

```
Feature: stalk-color-below-ring
```

```
  stalk-color-below-ring          e        p
  b                           0      100
  c                           0      100
  e                         100        0
  g                         100        0
  n                          12.5     87.5
  o                         100        0
  p                          30.77    69.23
  w                          61.68    38.32
  y                           0      100
```

```
Feature: veil-color
```

```
  veil-color          e        p
  n              100        0
  o              100        0
  w               50.68    49.32
  y                0      100
```

```
Feature: ring-number
```

```
  ring-number         e        p
  n                0      100
  o               49.15    50.85
```

```
  t               88       12
```

```
Feature: ring-type
```

```
  ring-type          e        p
  e              36.31    63.69
  f             100        0
  l               0      100
  n               0      100
  p              79.44    20.56
```

```
Feature: spore-print-color
```

```
  spore-print-color          e        p
  b                      100        0
  h                        2.94    97.06
  k                       88.03    11.97
  n                       88.62    11.38
  o                      100        0
  r                        0      100
  u                      100        0
  w                       24.12    75.88
  y                      100        0
```

```
Feature: population
```

```
  population          e       p
  a              100       0
  c               84.71   15.29
```

```
 n              100       0

 s               70.51   29.49

 v               29.5    70.5

 y               62.15   37.85
```

```
Feature: habitat
```

```
 habitat          e       p

 d            59.72   40.28

 g            65.55   34.45

 l            28.85   71.15

 m            87.67   12.33

 p            11.89   88.11

 u            26.09   73.91

 w           100       0
```

0.2.3 Modeling:

Before we get our base line model we are going to one-hot encode all the columns. We are encoding because all our features are categorical. Since sklearn libararies doesn't support non-numeric column we cannot get the base line model.

Models to try:

- Logistic Regression
- Decision Tree

[9]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.linear_model import LogisticRegression
```

[10]:
```
X = df.drop("class", axis=1)
y = df["class"]
X = pd.get_dummies(X).astype(int)
```

Base Model:

Using Logistic Regression as base model.

[11]:
```
model = LogisticRegression()
classifyModel(model, X, y)
```

```
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

           e       1.00      1.00      1.00       843
           p       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
```

Our base model is performing too good. However, we can try another model to observe any changes.

Decision Tree Classifier:

Trying Decision Tree Classifier.

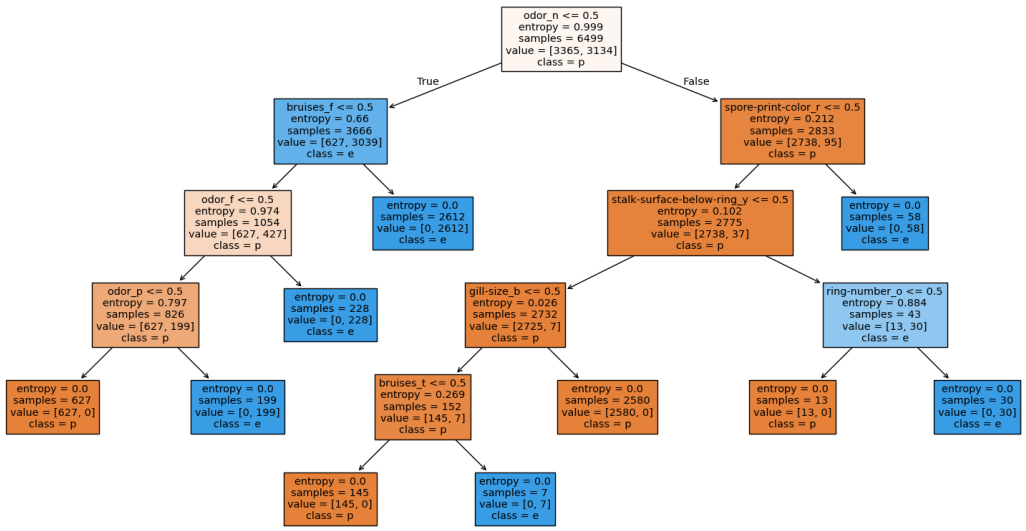
[ ]:
```
## using entropy as the criteria, random_state = 42 so it always produces same␣
 ↪tree.
model = DecisionTreeClassifier(random_state=42, criterion= 'entropy')
classifyModel(model, X, y)
```

```
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

           e       1.00      1.00      1.00       843
           p       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
```

We observe identical results for both the models.

Visualizing how the tree is actually performing.

[13]:
```
plt.figure(figsize=(20,10))
tree.plot_tree(model, feature_names=X.columns, class_names=y, filled=True)
plt.show()
```



We can observe that the tree is of depth 5 (root at 0). On every leaf node we are getting entropy of 0.0 which makes is obvious that there is no impurity.

[14]:
```
model = DecisionTreeClassifier(random_state=42, criterion= 'entropy')
cross_validate_decision_tree(model, X, y)
```

```
Fold 1:
Accuracy: 1.0
Classification Report:
Class      Precision    Recall    F1-Score    Support
-------  -----------  --------  ----------  ---------
e             1.0000    1.0000      1.0000  1040.0000
p             1.0000    1.0000      1.0000   991.0000
----------------------------------------
Fold 2:
Accuracy: 1.0
Classification Report:
Class      Precision    Recall    F1-Score    Support
-------  -----------  --------  ----------  ---------
e             1.0000    1.0000      1.0000  1045.0000
p             1.0000    1.0000      1.0000   986.0000
----------------------------------------
Fold 3:
Accuracy: 1.0
Classification Report:
Class      Precision    Recall    F1-Score    Support
-------  -----------  --------  ----------  ---------
e             1.0000    1.0000      1.0000  1047.0000
p             1.0000    1.0000      1.0000   984.0000
```

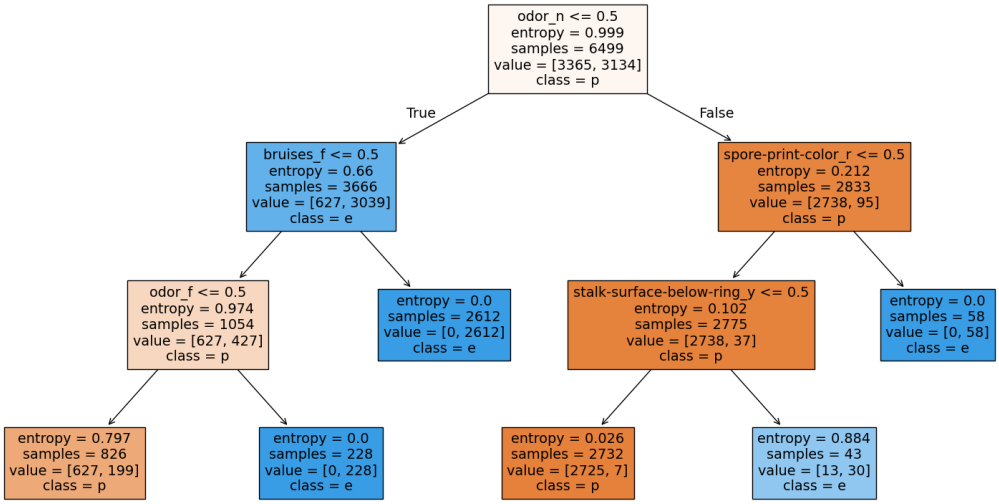
```
----------------------------------------
Fold 4:
Accuracy: 1.0
Classification Report:
Class      Precision    Recall    F1-Score    Support
-------  -----------  --------  ----------  ---------
e             1.0000    1.0000      1.0000  1076.0000
p             1.0000    1.0000      1.0000   955.0000
----------------------------------------
Mean Performance:
Mean Accuracy: 1.0000
```

There is nothing much that can be done. However, we can plot a decision tree using a fixed depth, choosing between entropy or gini index for information gain.

[15]:
```
## choosing entropy for criteria and keeping max depth at 3
model = DecisionTreeClassifier(random_state=42, criterion= 'entropy',␣
 ↪max_depth= 3)
classifyModel(model, X, y)
```

```
Accuracy: 0.9624615384615385
Classification Report:
              precision    recall  f1-score   support

           e       0.94      1.00      0.96       843
           p       1.00      0.93      0.96       782

    accuracy                           0.96      1625
   macro avg       0.97      0.96      0.96      1625
weighted avg       0.96      0.96      0.96      1625
```

[16]:
```
plt.figure(figsize=(20,10))
tree.plot_tree(model, feature_names=X.columns, class_names=y, filled=True)
plt.show()
```



We can observe that two of the leaf nodes have a very high entropy however our metrics are still in good shape.

Result:

The mushroom dataset is well-structured and required minimal preprocessing. Except for one column containing missing values, the data was clean and ready for modeling. A decision tree classifier was used, and cross-validation was applied to evaluate performance. Hyperparameter tuning was performed, however cross-validation and hyperparameter tuning did not lead to significant changes in accuracy or other evaluation metrics. This suggests that the dataset may exhibit strong class separability, making it straightforward to classify using basic models.