In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
df=pd.read_csv('mushrooms.csv')
df.head()
```

Out[2]:

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	st co abo
0	p	x	s	n	t	p	f	c	n	k	...	s	
1	e	x	s	y	t	a	f	c	b	k	...	s	
2	e	b	s	w	t	l	f	c	b	n	...	s	
3	p	x	y	w	t	p	f	c	n	n	...	s	
4	e	x	s	g	f	n	f	w	b	k	...	s	

5 rows × 23 columns

In [3]:

```
df.info()
```

```
<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
```

In [4]:

```
df.index
```

Out[4]:
```
RangeIndex(start=0, stop=8124, step=1)
```

In [5]:
```
df.columns
```

Out[5]:
```
Index(['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'],
      dtype='object')
```

In [6]:
```
df['class'].value_counts()
```

Out[6]:
```
e    4208
p    3916
Name: class, dtype: int64
```

In [7]:
```
df.describe().transpose()
```

Out[7]:

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

In [8]:

```
x=df.drop('class',axis=1)
x=pd.get_dummies(x,drop_first=True)
y=df['class']
```

In [9]:

```
x.head()
```

Out[9]:

	cap-shape_c	cap-shape_f	cap-shape_k	cap-shape_s	cap-shape_x	cap-surface_g	cap-surface_s	cap-surface_y	cap-color_c	cap-color_e
0	0	0	0	0	1	0	1	0	0	0
1	0	0	0	0	1	0	1	0	0	0
2	0	0	0	0	0	0	1	0	0	0
3	0	0	0	0	1	0	0	1	0	0
4	0	0	0	0	1	0	1	0	0	0

5 rows × 95 columns

In [10]:

```
y.head()
```

Out[10]:

```
0    p
1    e
2    e
3    p
4    e
Name: class, dtype: object
```

In [11]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score


x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

model=LogisticRegression()
model.fit(x_train,y_train)

train_pred=model.predict(x_train)
test_pred=model.predict(x_test)
model.score(x_test,y_test)
```

Out[11]:

```
1.0
```

In [12]:

```
test_pred=model.predict(x_test)
```

In [13]:

```
model.score(x_test,y_test)
```

Out[13]:

```
1.0
```

In [14]:

```
model.score(x_train,y_train)
```

Out[14]:

```
1.0
```

In [18]:

```
from sklearn.model_selection import cross_val_score
scores=cross_val_score(model,x,y,cv=5)
scores.mean()
```

Out[18]:

```
0.8942316786661614
```

In [ ]: