

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
mushroom=pd.read_csv('mushroom.csv')
```

In [3]:

```
mushroom
```

Out[3]:

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number
0	p	x	s	n	t	p	f	c	n	k	...	s	w	w	p	w	c
1	e	x	s	y	t	a	f	c	b	k	...	s	w	w	p	w	c
2	e	b	s	w	t	l	f	c	b	n	...	s	w	w	p	w	c
3	p	x	y	w	t	p	f	c	n	n	...	s	w	w	p	w	c
4	e	x	s	g	f	n	f	w	b	k	...	s	w	w	p	w	c
...
8119	e	k	s	n	f	n	a	c	b	y	...	s	o	o	p	o	c
8120	e	x	s	n	f	n	a	c	b	y	...	s	o	o	p	n	c
8121	e	f	s	n	f	n	a	c	b	n	...	s	o	o	p	o	c
8122	p	k	y	n	f	y	f	c	n	b	...	k	w	w	p	w	c
8123	e	x	s	n	f	n	a	c	b	y	...	s	o	o	p	o	c

8124 rows × 23 columns

In [4]:

```
mushroom.describe()
```

Out[4]:

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number
count	8124	8124	8124	8124	8124	8124	8124	8124	8124	8124	...	8124	8124	8124	8124	8124	8
unique	2	6	4	10	2	9	2	2	2	12	...	4	9	9	1	4	
top	e	x	y	n	f	n	f	c	b	b	...	s	w	w	p	w	
freq	4208	3656	3244	2284	4748	3528	7914	6812	5612	1728	...	4936	4464	4384	8124	7924	7

4 rows × 23 columns

4 rows × 23 columns

In [5]:

```
mushroom.dtypes
```

Out[5]:

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

In [21]:

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
list1=['class','cap-shape','cap-surface','cap-color','bruises','odor','gill-attachment','gill-spacing','gill-size','gill-color','stalk-shape','stalk-root','stalk-surface-below-ring','stalk-surface-above-ring','stalk-color-below-ring','veil-type','veil-color','ring-number','ring-type','spore-print-color','population','habitat']
for val in list1:
    mushroom[val]=le.fit_transform(mushroom[val].astype(str))
```

In [22]:

```
mushroom
```

Out[22]:

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number
0	1	5	2	4	1	6	1	0	1	10	...	2	7	7	0	2	...
1	0	5	2	9	1	0	1	0	0	10	...	2	7	7	0	2	...
2	0	0	2	8	1	3	1	0	0	11	...	2	7	7	0	2	...
3	1	5	3	8	1	6	1	0	1	11	...	2	7	7	0	2	...
4	0	5	2	3	0	5	1	1	0	10	...	2	7	7	0	2	...
...
8119	0	3	2	4	0	5	0	0	0	7	...	2	5	5	0	1	...
8120	0	5	2	4	0	5	0	0	0	7	...	2	5	5	0	0	...
8121	0	2	2	4	0	5	0	0	0	11	...	2	5	5	0	1	...
8122	1	3	3	4	0	8	1	0	1	0	...	1	7	7	0	2	...
8123	0	5	2	4	0	5	0	0	0	7	...	2	5	5	0	1	...

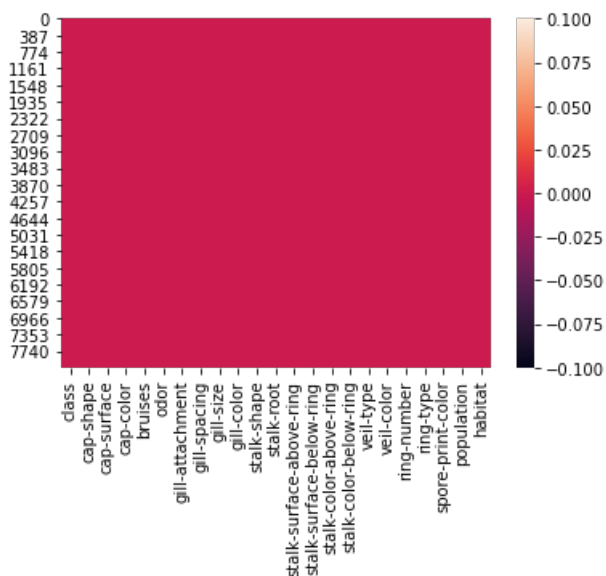
8124 rows × 23 columns

In [23]:

```
sns.heatmap(mushroom.isnull())
```

Out[23]:

<matplotlib.axes._subplots.AxesSubplot at 0x20c291373a0>



In [24]:

```
mushroom.isnull().sum()
```

Out[24]:

```
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
```

In [25]:

```
mushroom.skew()
```

Out[25]:

```
class 0.071946
cap-shape -0.247052
cap-surface -0.590859
cap-color 0.706965
bruises 0.342750
```

```

odor -0.080790
gill-attachment -5.977076
gill-spacing 1.840088
gill-size 0.825797
gill-color -0.013462
stalk-shape -0.271345
stalk-root 0.947852
stalk-surface-above-ring -1.098739
stalk-surface-below-ring -0.757703
stalk-color-above-ring -1.835434
stalk-color-below-ring -1.791593
veil-type 0.000000
veil-color -6.946944
ring-number 2.701657
ring-type -0.290018
spore-print-color 0.548426
population -1.413096
habitat 0.985548
dtype: float64

```

In [26]:

```
mushroom.corr()
```

Out[26]:

	class	cap- shape	cap- surface	cap- color	bruises	odor	gill- attachment	gill- spacing	gill-size	gill-color	...	stalk- surface- below- ring	stalk- color- above- ring
class	1.000000	0.052951	0.178446	0.031384	0.501530	0.093552	0.129200	0.348387	0.540024	0.407918	...	0.298801	0.154003
cap-shape	0.052951	1.000000	0.050454	0.048203	0.035374	0.021935	0.078865	0.013196	0.054050	0.004837	...	0.032591	0.031659
cap-surface	0.178446	0.050454	1.000000	0.019402	0.070228	0.045233	-0.034180	0.282306	0.208100	0.215089	...	0.107965	0.066050
cap-color	0.031384	0.048203	0.019402	1.000000	0.000764	0.387121	0.041436	0.144259	0.169464	0.308644	...	0.047710	0.002364
bruises	0.501530	0.035374	0.070228	0.000764	1.000000	0.061825	0.137359	0.299473	0.369596	0.275185	...	0.458983	0.083538
odor	0.093552	0.021935	0.045233	0.387121	0.061825	1.000000	-0.059590	0.063936	0.310495	0.306347	...	0.061820	0.174532
gill-attachment	0.129200	0.078865	0.034180	0.041436	0.137359	0.059590	1.000000	0.071489	0.108984	0.056529	...	0.116177	0.099299
gill-spacing	0.348387	0.013196	0.282306	0.144259	0.299473	0.063936	0.071489	1.000000	0.108333	0.232567	...	0.213775	0.274574
gill-size	0.540024	0.054050	0.208100	0.169464	0.369596	0.310495	0.108984	0.108333	1.000000	0.551397	...	0.010894	0.296548
gill-color	0.407918	0.004837	0.215089	0.308644	0.275185	0.306347	-0.056529	0.232567	0.551397	1.000000	...	0.030965	0.073407
stalk-shape	0.102019	0.063794	0.014123	0.456496	0.099364	0.459766	0.186485	0.080895	0.214576	0.328393	...	0.034399	0.223407
stalk-root	0.379361	0.030191	0.126245	0.321274	0.244188	0.205215	0.144063	0.350548	0.344345	0.542456	...	0.087454	0.157703
stalk-surface-above-ring	0.334593	0.030417	0.089090	0.060837	0.460824	0.118617	-0.088916	0.212359	0.056310	0.015833	...	0.437164	0.132703
stalk-surface-below-ring	0.298801	0.032591	0.107965	0.047710	0.458983	0.061820	-0.116177	0.213775	0.010894	0.030965	...	1.000000	0.106933
stalk-color-above-ring	0.154003	0.031659	0.066050	0.002364	0.083538	0.174532	0.099299	0.274574	0.296548	0.073407	...	0.106933	1.000000
stalk-color-below-ring	0.146730	0.030390	0.068885	0.008057	0.092874	0.169407	0.097160	0.253505	0.278708	0.075194	...	0.110656	0.491593
veil-type	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
veil-color	0.145142	0.072560	0.016603	0.036130	0.119770	0.057747	0.897518	0.073363	0.103809	0.049998	...	0.077284	0.067301
ring-number	0.041222	0.100524	0.000447	0.005000	0.056788	0.111905	0.093236	0.243014	0.174000	0.005554	...	0.040006	0.084500

number	0.214366	0.106534	0.026147	0.005822					0.171362	0.025554				
ring-type	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	
spore-print-color	0.171961	0.073416	0.230364	0.293523	0.285008	0.469055	-0.029524	0.047323	0.622991	0.597511	...	0.130974	0.2715	
population	0.298686	0.063413	0.021555	0.144770	0.088137	0.043623	0.165575	0.529253	0.147682	0.169374	...	0.046797	0.2402	
habitat	0.217179	0.042221	0.163887	0.033925	0.075095	0.026610	-0.030304	0.154680	0.161418	0.127340	...	0.039628	0.0425	

23 rows × 23 columns

In [27]:

```
for col in mushroom.columns:
    if mushroom.skew().loc[col]>0.55:
        mushroom[col]=np.log1p(mushroom[col])
```

In [28]:

```
mushroom.skew()
```

Out[28]:

```
class                0.071946
cap-shape           -0.247052
cap-surface         -0.590859
cap-color           -0.365280
bruises             0.342750
odor               -0.080790
gill-attachment     -5.977076
gill-spacing         1.840088
gill-size           0.825797
gill-color          -0.013462
stalk-shape         -0.271345
stalk-root          0.129453
stalk-surface-above-ring -1.098739
stalk-surface-below-ring -0.757703
stalk-color-above-ring -1.835434
stalk-color-below-ring -1.791593
veil-type           0.000000
veil-color          -6.946944
ring-number         1.481287
ring-type           -0.290018
spore-print-color    0.548426
population          -1.413096
habitat             0.342186
dtype: float64
```

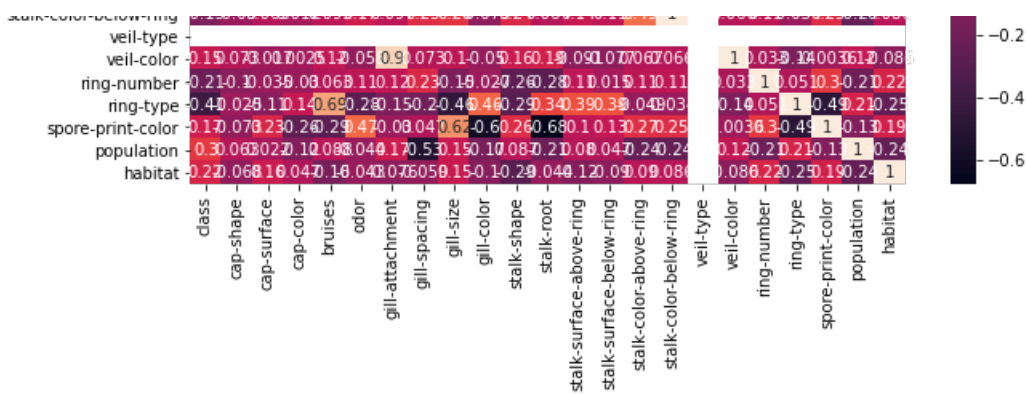
In [29]:

```
plt.figure(figsize=(10,6))
sns.heatmap(mushroom.corr(),annot=True)
```

Out[29]:

<matplotlib.axes._subplots.AxesSubplot at 0x20c2f47b3d0>





In [30]:

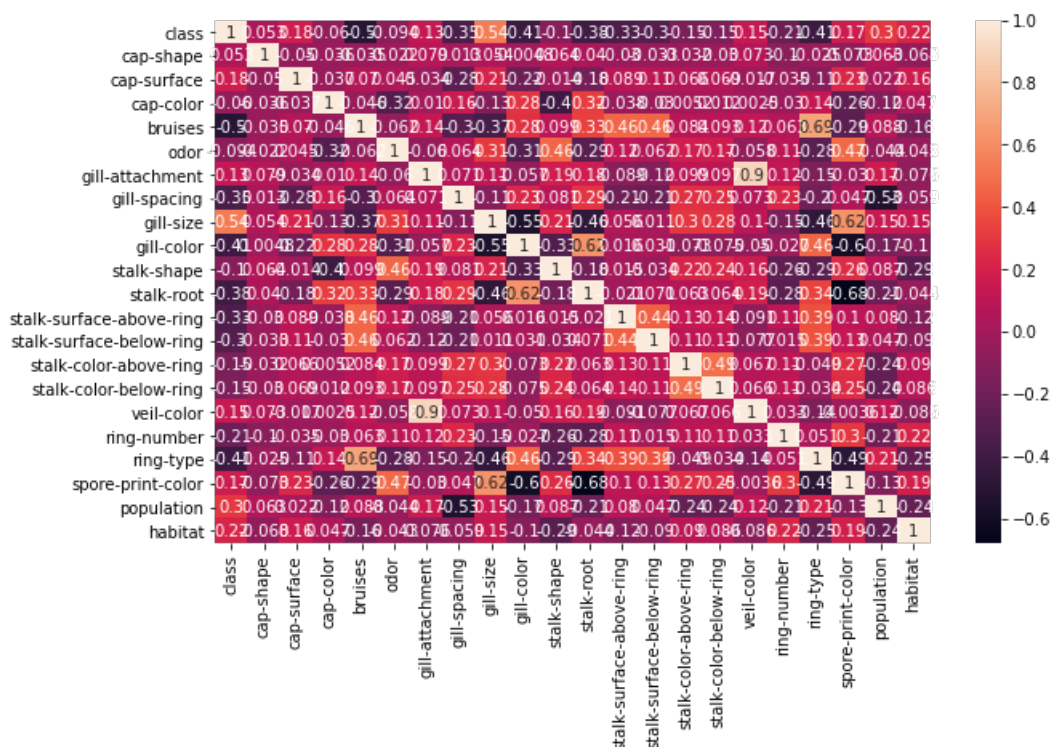
```
mushroom=mushroom.drop(['veil-type'],axis=1)
```

In [31]:

```
plt.figure(figsize=(10,6))
sns.heatmap(mushroom.corr(),annot=True)
```

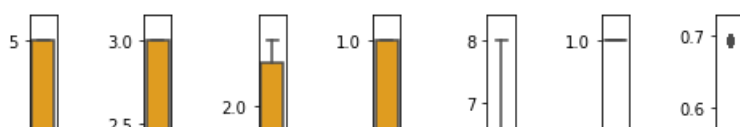
Out[31]:

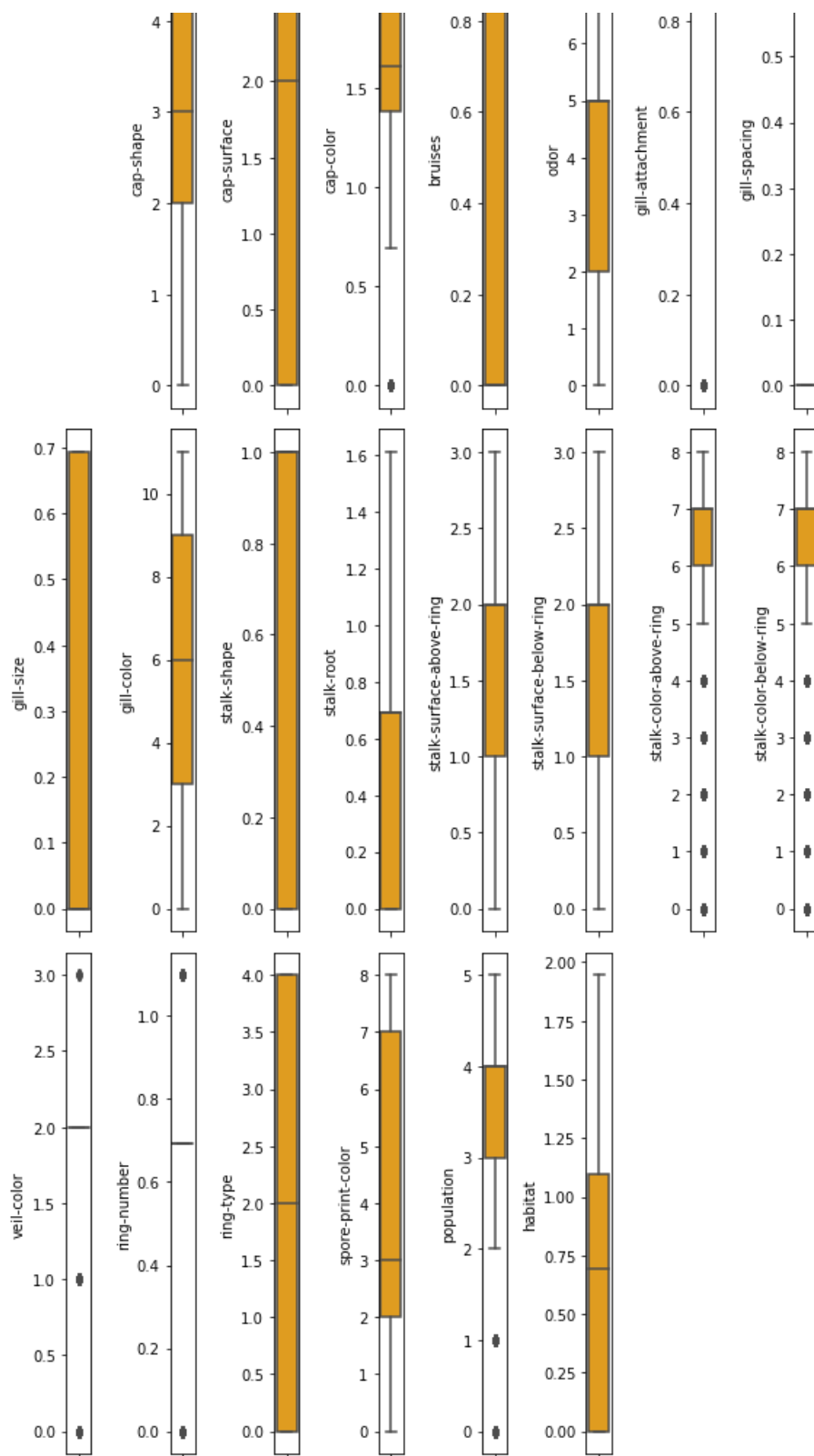
<matplotlib.axes._subplots.AxesSubplot at 0x20c30bdf4f0>



In [32]:

```
col=mushroom.columns.values
ncol=8
nrow=8
plt.figure(figsize=(ncol,5*ncol))
for i in range(1,len(col)):
    plt.subplot(nrow,ncol,i+1)
    sns.boxplot(mushroom[col[i]],color='orange',orient='v')
plt.tight_layout()
```





In [33]:

```
from scipy.stats import zscore
z_score=abs(zscore(mushroom))
print(mushroom.shape)
mus=mushroom.loc[(z_score<3).all(axis=1)]
print(mus.shape)
```

(8124, 22)

(6472, 22)

In [34]:

```
mus
```

Out[34]:

	class	cap- shape	cap- surface	cap- color	bruises	odor	gill- attachment	gill- spacing	gill-size	gill- color	...	stalk- surface- above- ring	stalk- surface- below- ring	stalk- color- above- ring	stalk- color- below- ring	c
0	1	5	2	1.609438	1	6	1	0.000000	0.693147	10	...	2	2	7	7	
1	0	5	2	2.302585	1	0	1	0.000000	0.000000	10	...	2	2	7	7	
2	0	0	2	2.197225	1	3	1	0.000000	0.000000	11	...	2	2	7	7	
3	1	5	3	2.197225	1	6	1	0.000000	0.693147	11	...	2	2	7	7	
4	0	5	2	1.386294	0	5	1	0.693147	0.000000	10	...	2	2	7	7	
...
8113	1	3	3	1.098612	0	8	1	0.000000	0.693147	0	...	1	1	6	6	
8116	1	3	3	1.609438	0	7	1	0.000000	0.693147	0	...	2	1	6	7	
8117	1	3	2	1.098612	0	8	1	0.000000	0.693147	0	...	1	2	6	7	
8118	1	3	3	1.609438	0	2	1	0.000000	0.693147	0	...	1	2	6	7	
8122	1	3	3	1.609438	0	8	1	0.000000	0.693147	0	...	2	1	7	7	

6472 rows × 22 columns



In [35]:

```
x=mus.iloc[:,1:-1]
x
```

Out[35]:

	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	s
0	5	2	1.609438	1	6	1	0.000000	0.693147	10	0	1.386294	2	2	7	
1	5	2	2.302585	1	0	1	0.000000	0.000000	10	0	1.098612	2	2	7	
2	0	2	2.197225	1	3	1	0.000000	0.000000	11	0	1.098612	2	2	7	
3	5	3	2.197225	1	6	1	0.000000	0.693147	11	0	1.386294	2	2	7	
4	5	2	1.386294	0	5	1	0.693147	0.000000	10	1	1.386294	2	2	7	
...
8113	3	3	1.098612	0	8	1	0.000000	0.693147	0	1	0.000000	1	1	6	
8116	3	3	1.609438	0	7	1	0.000000	0.693147	0	1	0.000000	2	1	6	
8117	3	2	1.098612	0	8	1	0.000000	0.693147	0	1	0.000000	1	2	6	
8118	3	3	1.609438	0	2	1	0.000000	0.693147	0	1	0.000000	1	2	6	
8122	3	3	1.609438	0	8	1	0.000000	0.693147	0	1	0.000000	2	1	7	

6472 rows × 20 columns



In [36]:

```
x.shape
```

Out[36]:

(6472, 20)

In [37]:

```
y=mus.iloc[:,0]
y
```

Out[37]:


```

0      1
1      0
2      0
3      1
4      0
..
8113   1
8116   1
8117   1
8118   1
8122   1
Name: class, Length: 6472, dtype: int32

```

In [38]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.22,random_state=43)
```

In [39]:

```
lr=LogisticRegression()
```

In [41]:

```

lr.fit(x_train,y_train)
lr.score(x_train,y_train)
pred=lr.predict(x_test)
print(accuracy_score(y_test,pred))
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))

```

```

0.9627808988764045
[[729  22]
 [ 31 642]]

```

	precision	recall	f1-score	support
0	0.96	0.97	0.96	751
1	0.97	0.95	0.96	673
accuracy			0.96	1424
macro avg	0.96	0.96	0.96	1424
weighted avg	0.96	0.96	0.96	1424

In [42]:

```

lrscores=cross_val_score(lr,x,y,cv=6)
print(lrscores)
print(lrscores.mean(),lrscores.std())

```

```

[0.74235403 0.91566265 0.91380908 1.          1.          0.92207792]
0.9156506144457951 0.08593218742315831

```

In [45]:

```

rf=RandomForestClassifier()
rf.fit(x_train,y_train)
predrf=rf.predict(x_test)
print(accuracy_score(y_test,predrf))
print(confusion_matrix(y_test,predrf))
print(classification_report(y_test,predrf))

```

```

1.0
[[751  0]
 [  0 673]]

```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	751
1	1.00	1.00	1.00	673
accuracy			1.00	1424

accuracy			1.00	1424
macro avg	1.00	1.00	1.00	1424
weighted avg	1.00	1.00	1.00	1424

In [46]:

```
rfscores=cross_val_score(rf,x,y,cv=6)
print(rfscores)
print(rfscores.mean(),rfscores.std())
```

```
[0.76274328 1.          1.          1.          1.          0.9109462 ]
0.9456149129126755 0.08801032562756267
```

In [47]:

```
svc=SVC(kernel='rbf')
svc.fit(x_train,y_train)
svc.score(x_train,y_train)
predsvc=svc.predict(x_test)
print(accuracy_score(y_test,predsvc))
print(confusion_matrix(y_test,predsvc))
print(classification_report(y_test,predsvc))
```

```
0.9957865168539326
[[749  2]
 [ 4 669]]
```

	precision	recall	f1-score	support
0	0.99	1.00	1.00	751
1	1.00	0.99	1.00	673
accuracy			1.00	1424
macro avg	1.00	1.00	1.00	1424
weighted avg	1.00	1.00	1.00	1424

In [48]:

```
svcscores=cross_val_score(svc,x,y,cv=6)
print(svcscores)
print(svcscores.mean(),svcscores.std())
```

```
[0.73123262 0.97219648 0.89620019 1.          1.          0.9109462 ]
0.9184292471727927 0.09290783424027775
```

In [49]:

```
knn=KNeighborsClassifier()
knn.fit(x_train,y_train)
knn.score(x_train,y_train)
predknn=knn.predict(x_test)
print(accuracy_score(y_test,predknn))
print(confusion_matrix(y_test,predknn))
print(classification_report(y_test,predknn))
```

```
0.9985955056179775
[[749  2]
 [ 0 673]]
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	751
1	1.00	1.00	1.00	673
accuracy			1.00	1424
macro avg	1.00	1.00	1.00	1424
weighted avg	1.00	1.00	1.00	1424

In [50]:

In [50]:

```
knn_scores=cross_val_score(knn,x,y,cv=6)
print(knn_scores)
print(knn_scores.mean(),knn_scores.std())
```

```
[0.74513438 0.99258573 0.9592215 1.          1.          0.9554731 ]
0.942069118489084 0.08992688005587735
```

In [51]:

```
gnb=GaussianNB()
gnb.fit(x_train,y_train)
gnb.score(x_train,y_train)
predgnb=gnb.predict(x_test)
print(accuracy_score(y_test,predgnb))
print(confusion_matrix(y_test,predgnb))
print(classification_report(y_test,predgnb))
```

```
0.8441011235955056
[[726  25]
 [197 476]]
```

	precision	recall	f1-score	support
0	0.79	0.97	0.87	751
1	0.95	0.71	0.81	673
accuracy			0.84	1424
macro avg	0.87	0.84	0.84	1424
weighted avg	0.86	0.84	0.84	1424

In [52]:

```
gnb_scores=cross_val_score(gnb,x,y,cv=6)
print(gnb_scores)
print(gnb_scores.mean(),gnb_scores.std())
```

```
[0.62372567 0.58480074 0.79703429 1.          1.          0.9109462 ]
0.8194178168360612 0.16705233907872133
```

In [53]:

```
dtc=DecisionTreeClassifier()
dtc.fit(x_train,y_train)
dtc.score(x_train,y_train)
preddtc=dtc.predict(x_test)
print(accuracy_score(y_test,preddtc))
print(confusion_matrix(y_test,preddtc))
print(classification_report(y_test,preddtc))
```

```
1.0
[[751  0]
 [ 0 673]]
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	751
1	1.00	1.00	1.00	673
accuracy			1.00	1424
macro avg	1.00	1.00	1.00	1424
weighted avg	1.00	1.00	1.00	1424

In [54]:

```
dtc_scores=cross_val_score(dtc,x,y,cv=6)
print(dtc_scores)
print(dtc_scores.mean(),dtc_scores.std())
```

```
[0.88137164 1.          0.99629286 1.          1.          0.9109462 ]
```

```
[0.9647684501385019 0.04927717154693356]
0.9647684501385019 0.04927717154693356
```

In [55]:

```
import joblib
joblib.dump(rf, 'mushroom.pkl')
```

Out[55]:

```
['mushroom.pkl']
```

In []: