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.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
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
#observed that class is the output
```

Out[3]:

	type	cap_shape	cap_surface	cap_color	bruises	odor	gill_attachment	gill_spacing	gill_size	gill_color	...	stalk_surface_belc
0	p	x	s	n	t	p	f	c	n	k	...	
1	e	x	s	y	t	a	f	c	b	k	...	
2	e	b	s	w	t	l	f	c	b	n	...	
3	p	x	y	w	t	p	f	c	n	n	...	
4	e	x	s	g	f	n	f	w	b	k	...	
...	...	...	...	...	...	...	...	...	...	...	...	
8119	e	k	s	n	f	n	a	c	b	y	...	
8120	e	x	s	n	f	n	a	c	b	y	...	
8121	e	f	s	n	f	n	a	c	b	n	...	
8122	p	k	y	n	f	y	f	c	n	b	...	
8123	e	x	s	n	f	n	a	c	b	y	...	

8124 rows × 23 columns

In [4]:

```
mushroom.describe()
```

Out[4]:

	type	cap_shape	cap_surface	cap_color	bruises	odor	gill_attachment	gill_spacing	gill_size	gill_color	...	stalk_surface_b
count	8124	8124	8124	8124	8124	8124	8124	8124	8124	8124	...	
unique	2	6	4	10	2	9	2	2	2	12	...	
top	e	x	y	n	f	n	f	c	b	b	...	
freq	4208	3656	3244	2284	4748	3528	7914	6812	5612	1728	...	

4 rows × 23 columns

In [5]:

```
mushroom.dtypes
```

Out[5]:

```
type                        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 [6]:

```
#converting data in numerical data
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
list1=['type','cap_shape','cap_surface','cap_color','bruises','odor','gill_attachment','gill_spacin
g','gill_size','gill_color','stalk_shape','stalk_root','stalk_surface_above_ring','stalk_surface_be
low_ring','stalk_color_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 [7]:

```
mushroom
```

Out[7]:

	type	cap_shape	cap_surface	cap_color	bruises	odor	gill_attachment	gill_spacing	gill_size	gill_color	...	stalk_surface_belc
0	1	5	2	4	1	6	1	0	1	4	...	
1	0	5	2	9	1	0	1	0	0	4	...	
2	0	0	2	8	1	3	1	0	0	5	...	
3	1	5	3	8	1	6	1	0	1	5	...	
4	0	5	2	3	0	5	1	1	0	4	...	
...	...	...	...	...	...	...	...	...	...	...	...	
8119	0	3	2	4	0	5	0	0	0	11	...	
8120	0	5	2	4	0	5	0	0	0	11	...	
8121	0	2	2	4	0	5	0	0	0	5	...	
8122	1	3	3	4	0	8	1	0	1	0	...	
8123	0	5	2	4	0	5	0	0	0	11	...	

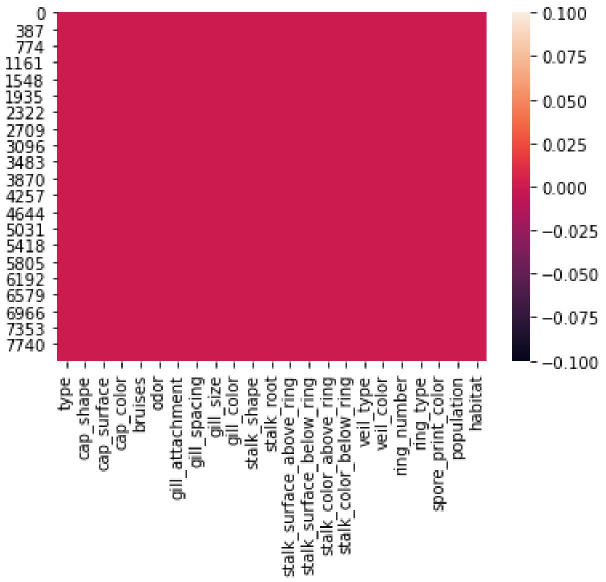
8124 rows × 23 columns

In [8]:

```
#identifing null values
sns.heatmap(mushroom.isnull())
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20148e7b280>
```



In [9]:

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

Out[9]:

```
type                        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 [10]:

```
mushroom.skew()
```

Out[10]:

```
type                        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                  0.061410
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 [11]:

```
mushroom.corr()
```

Out[11]:

	type	cap_shape	cap_surface	cap_color	bruises	odor	gill_attachment	gill_spacing	gill_size	gil
type	1.000000	0.052951	0.178446	-0.031384	-0.501530	-0.093552	0.129200	-0.348387	0.540024	0.
cap_shape	0.052951	1.000000	-0.050454	-0.048203	-0.035374	-0.021935	0.078865	0.013196	0.054050	0.
cap_surface	0.178446	-0.050454	1.000000	-0.019402	0.070228	0.045233	-0.034180	-0.282306	0.208100	0.
cap_color	-0.031384	-0.048203	-0.019402	1.000000	-0.000764	-0.387121	0.041436	0.144259	-0.169464	0.
bruises	-0.501530	-0.035374	0.070228	-0.000764	1.000000	-0.061825	0.137359	-0.299473	-0.369596	0.
odor	-0.093552	-0.021935	0.045233	-0.387121	-0.061825	1.000000	-0.059590	0.063936	0.310495	0.
gill_attachment	0.129200	0.078865	-0.034180	0.041436	0.137359	-0.059590	1.000000	0.071489	0.108984	0.
gill_spacing	-0.348387	0.013196	-0.282306	0.144259	-0.299473	0.063936	0.071489	1.000000	-0.108333	0.
gill_size	0.540024	0.054050	0.208100	-0.169464	-0.369596	0.310495	0.108984	-0.108333	1.000000	0.
gill_color	-0.530566	-0.006039	-0.161017	0.084659	0.527120	-0.129213	-0.128567	0.100193	-0.516736	1.
stalk_shape	-0.102019	0.063794	-0.014123	-0.456496	0.099364	0.459766	0.186485	0.080895	0.214576	0.
stalk_root	-0.379361	0.030191	-0.126245	0.321274	0.244188	-0.205215	0.144063	0.350548	-0.344345	0.
stalk_surface_above_ring	-0.334593	-0.030417	0.089090	-0.060837	0.460824	0.118617	-0.088916	-0.212359	0.056310	0.
stalk_surface_below_ring	-0.298801	-0.032591	0.107965	-0.047710	0.458983	0.061820	-0.116177	-0.213775	0.010894	0.
stalk_color_above_ring	-0.154003	-0.031659	0.066050	0.002364	0.083538	0.174532	0.099299	0.274574	0.296548	0.
stalk_color_below_ring	-0.146730	-0.030390	0.068885	0.008057	0.092874	0.169407	0.097160	0.253505	0.278708	0.
veil_type	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.
ring_number	-0.214366	-0.106534	-0.026147	-0.005822	0.056788	0.111905	0.093236	0.243014	-0.171362	0.
ring_type	-0.411771	-0.025457	-0.106407	0.162513	0.692973	-0.281387	-0.146689	-0.195897	-0.460872	0.
spore_print_color	0.171961	-0.073416	0.230364	-0.293523	-0.285008	0.469055	-0.029524	0.047323	0.622991	0.
population	0.298686	0.063413	0.021555	-0.144770	0.088137	-0.043623	0.165575	-0.529253	0.147682	0.

```
habitat 0.217 type cap_shape cap_surface cap_color bruises 0.075095 0.026610 odor gill_attachment gill_spacing gill_size gil
```

23 rows × 23 columns

In [12]:

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

In [13]:

```
#reduced skew
mushroom.skew()
```

Out[13]:

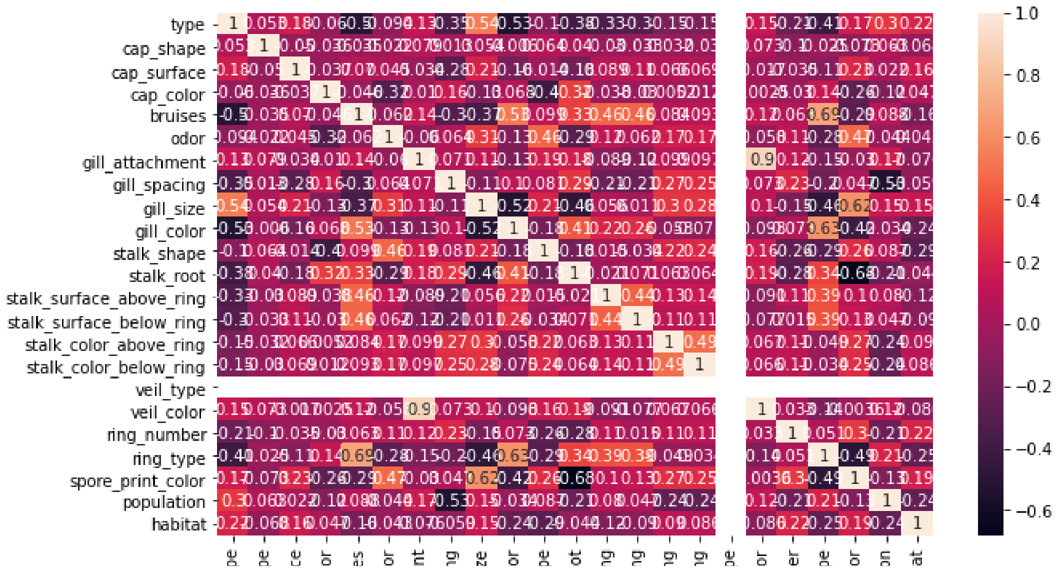
```
type                        0.071946
cap_shape                  -0.247052
cap_surface                -0.590859
cap_color                  -0.365280
bruises                     0.342750
odor                       -0.080790
gill_attachment            -5.977076
gill_spacing                1.840088
gill_size                   0.825797
gill_color                  0.061410
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
```

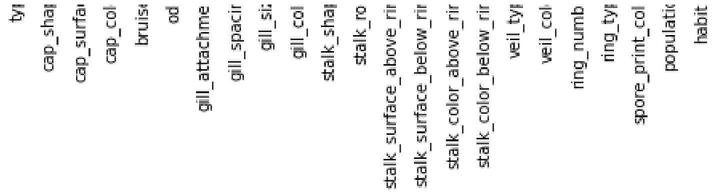
In [14]:

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

Out[14]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20148fd8460>
```





In [15]:

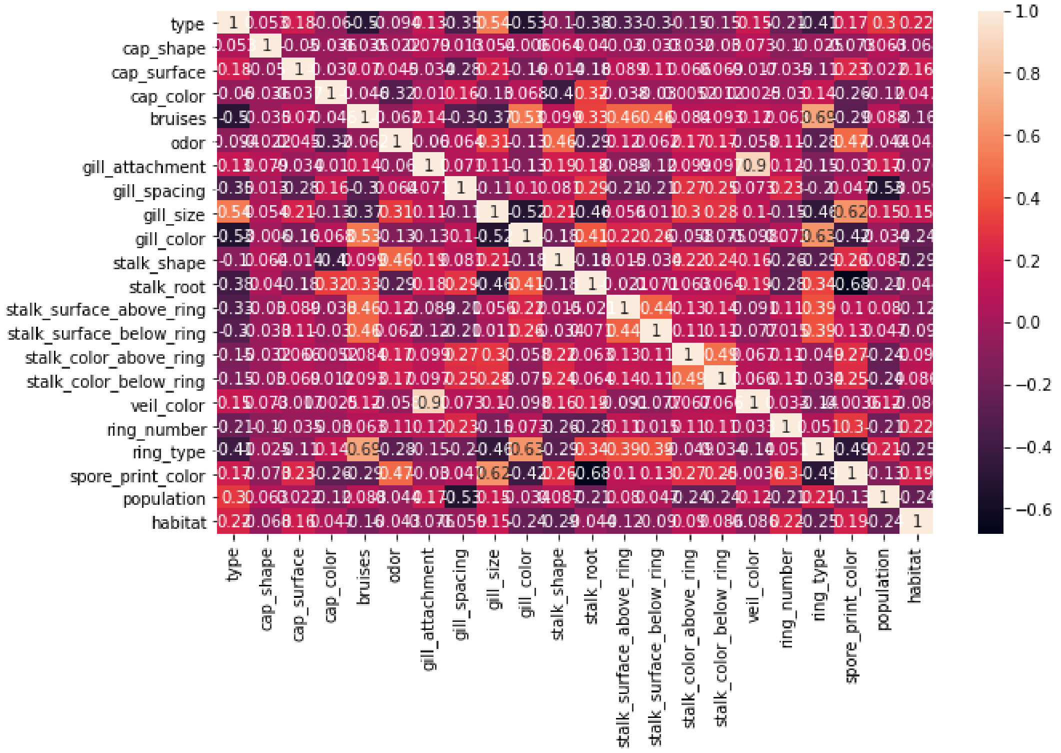
```
mushroom=mushroom.drop(['veil_type'],axis=1)
```

In [16]:

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

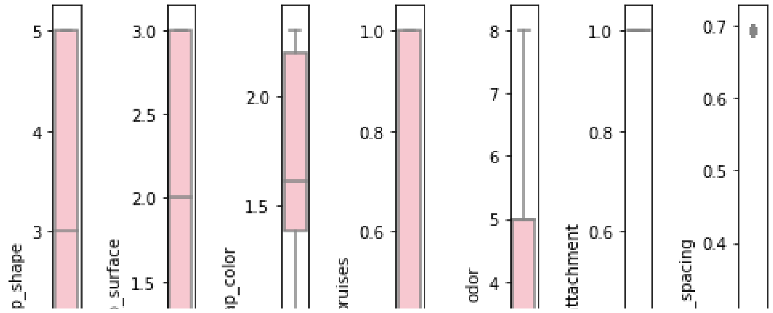
Out[16]:

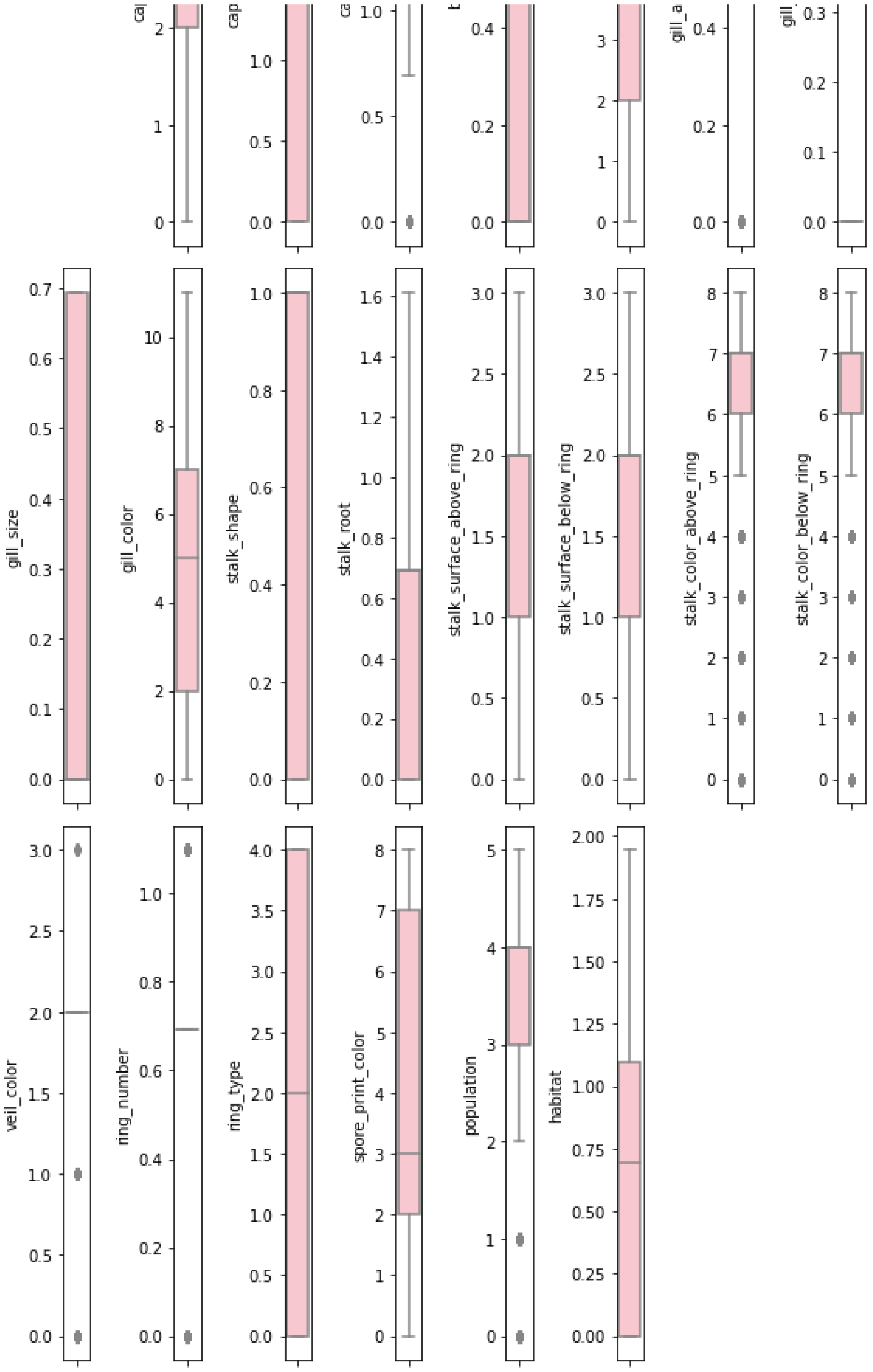
```
<matplotlib.axes._subplots.AxesSubplot at 0x20149ea2790>
```



In [17]:

```
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='pink',orient='v')
    plt.tight_layout()
```





In [18]:

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

```
(8124, 22)
(6472, 22)
```

In [19]:

```
mush
```

Out[19]:

	type	cap_shape	cap_surface	cap_color	bruises	odor	gill_attachment	gill_spacing	gill_size	gill_color	...	stalk_surface_abc
0	1	5	2	1.609438	1	6	1	0.000000	0.693147	4		

	type	cap_shape	cap_surface	cap_color	bruises	odor	gill_attachment	gill_spacing	gill_size	gill_color	...	stalk_surface_abc
0	1	5	2	1.609438	1	6	1	0.000000	0.693147	4	...	
1	0	5	2	2.302585	1	0	1	0.000000	0.000000	4	...	
2	0	0	2	2.197225	1	3	1	0.000000	0.000000	5	...	
3	1	5	3	2.197225	1	6	1	0.000000	0.693147	5	...	
4	0	5	2	1.386294	0	5	1	0.693147	0.000000	4	...	
...	...	...	...	...	...	...	...	...	...	...	...	
8113	1	3	3	1.098612	0	8	1	0.000000	0.693147	0	...	
8116	1	3	3	1.609438	0	7	1	0.000000	0.693147	0	...	
8117	1	3	2	1.098612	0	8	1	0.000000	0.693147	0	...	
8118	1	3	3	1.609438	0	2	1	0.000000	0.693147	0	...	
8122	1	3	3	1.609438	0	8	1	0.000000	0.693147	0	...	

6472 rows × 22 columns

In [20]:

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

In [21]:

```
x
```

Out[21]:

	cap_shape	cap_surface	cap_color	bruises	odor	gill_attachment	gill_spacing	gill_size	gill_color	stalk_shape	stalk_root	st
0	5	2	1.609438	1	6	1	0.000000	0.693147	4	0	1.386294	
1	5	2	2.302585	1	0	1	0.000000	0.000000	4	0	1.098612	
2	0	2	2.197225	1	3	1	0.000000	0.000000	5	0	1.098612	
3	5	3	2.197225	1	6	1	0.000000	0.693147	5	0	1.386294	
4	5	2	1.386294	0	5	1	0.693147	0.000000	4	1	1.386294	
...	...	...	...	...	...	...	...	...	...	...	...	
8113	3	3	1.098612	0	8	1	0.000000	0.693147	0	1	0.000000	
8116	3	3	1.609438	0	7	1	0.000000	0.693147	0	1	0.000000	
8117	3	2	1.098612	0	8	1	0.000000	0.693147	0	1	0.000000	
8118	3	3	1.609438	0	2	1	0.000000	0.693147	0	1	0.000000	
8122	3	3	1.609438	0	8	1	0.000000	0.693147	0	1	0.000000	

6472 rows × 20 columns

In [22]:

```
x.shape
```

Out[22]:

```
(6472, 20)
```

In [23]:

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

In [24]:

```
y
```

Out[24]:

```
0       1
1       0
```

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

In [25]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.30,random_state=50)
```

In [26]:

```
rf=RandomForestClassifier()
rf.fit(x_train,y_train)
rf.score(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
[[1027    0]
 [   0  915]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1027
           1       1.00      1.00      1.00       915

    accuracy                           1.00      1942
   macro avg       1.00      1.00      1.00      1942
weighted avg       1.00      1.00      1.00      1942
```

In [27]:

```
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
[[1027    0]
 [   0  915]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1027
           1       1.00      1.00      1.00       915

    accuracy                           1.00      1942
   macro avg       1.00      1.00      1.00      1942
weighted avg       1.00      1.00      1.00      1942
```

In [28]:

```
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))
```

```
0.9516992790937178
```

```
0.8516992790937178
[[975  52]
 [236 679]]
              precision    recall  f1-score   support

           0       0.81      0.95      0.87      1027
           1       0.93      0.74      0.83       915

    accuracy                           0.85      1942
   macro avg       0.87      0.85      0.85      1942
weighted avg       0.86      0.85      0.85      1942
```

In [29]:

```
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))
```

```
0.9979402677651905
[[1025    2]
 [   2  913]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1027
           1       1.00      1.00      1.00       915

    accuracy                           1.00      1942
   macro avg       1.00      1.00      1.00      1942
weighted avg       1.00      1.00      1.00      1942
```

In [30]:

```
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))
```

```
0.9891864057672503
[[1026    1]
 [  20  895]]
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      1027
           1       1.00      0.98      0.99       915

    accuracy                           0.99      1942
   macro avg       0.99      0.99      0.99      1942
weighted avg       0.99      0.99      0.99      1942
```

In [31]:

```
rf=RandomForestClassifier()
rf.fit(x_train,y_train)
rf.score(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
[[1027    0]
 [   0  915]]
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1027
           1       1.00      1.00      1.00       915

    accuracy                           1.00      1942
   macro avg       1.00      1.00      1.00      1942
weighted avg       1.00      1.00      1.00      1942
```

In [32]:

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

```
0.9577754891864058
[[988  39]
 [ 43 872]]
              precision    recall  f1-score   support

           0       0.96      0.96      0.96      1027
           1       0.96      0.95      0.96       915

    accuracy                           0.96      1942
   macro avg       0.96      0.96      0.96      1942
weighted avg       0.96      0.96      0.96      1942
```

In [33]:

```
#LogisticRegression is best
import joblib
joblib.dump(lr,'salary.pkl')
```

Out[33]:

```
['salary.pkl']
```