



```
In [68]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import LabelEncoder
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split

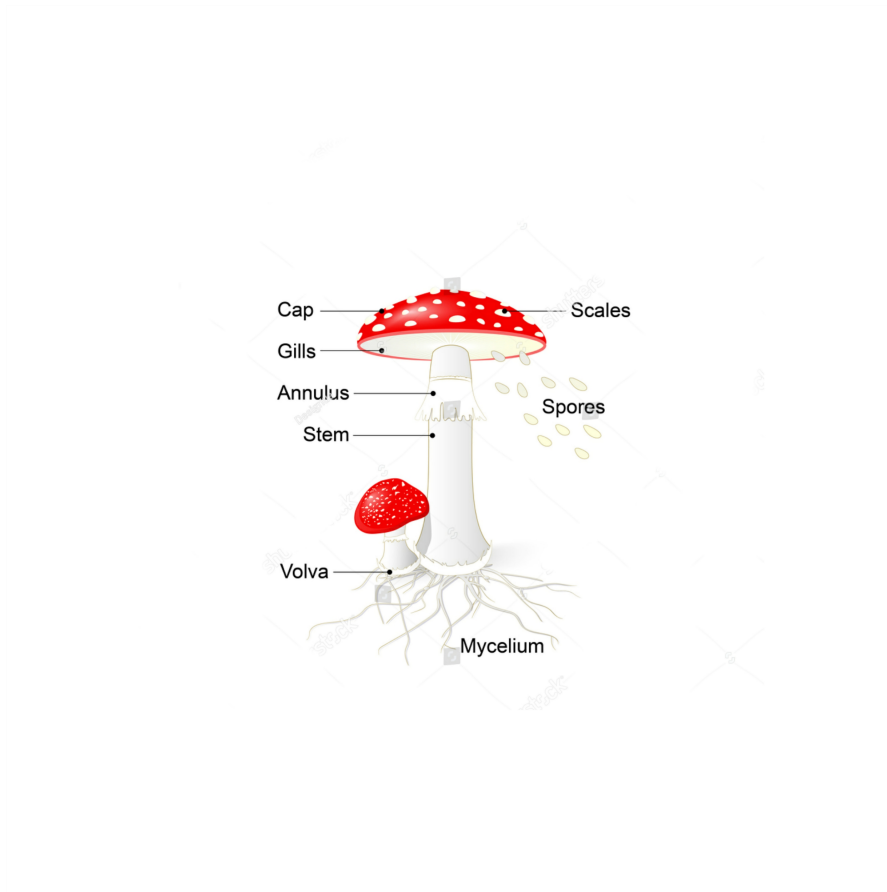
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier

from sklearn.metrics import accuracy_score

import joblib
```

```
from tkinter import *
```



*Attribute Information: (classes: edible=e, poisonous=p)*

*cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s*

*cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s*

*cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,red=e,white=w,yellow=y*

*bruises: bruises=t,no=f*

*odor: almond=a,anise=l,creosote=c,fishy=y,foul=f,musty=m,none=n,pungent=p,spicy=s*

*gill-attachment: attached=a,descending=d,free=f,notched=n*

*gill-spacing: close=c,crowded=w,distant=d*

*gill-size: broad=b,narrow=n*

*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*

*stalk-shape: enlarging=e,tapering=t*

*stalk-root: bulbous=b,club=c,cup=u,equal=e,rhizomorphs=z,rooted=r,missing=?*

*stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s*

*stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s*

*stalk-color-above-ring:*

*brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y*

*stalk-color-below-ring:*

*brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y*

*veil-type: partial=p,universal=u*

*veil-color: brown=n,orange=o,white=w,yellow=y*

*ring-number: none=n,one=o,two=t*

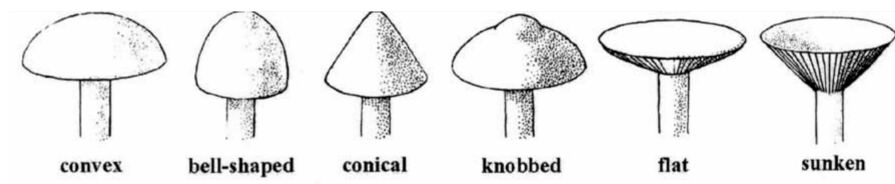
*ring-type: cobwebby=c,evanescent=e,flaring=f,large=l,none=n,pendant=p,sheathing=s,zone=z*

*spore-print-color:*

*black=k,brown=n,buff=b,chocolate=h,green=r,orange=o,purple=u,white=w,yellow=y*

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

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



```
In [56]: dataset = pd.read_csv('mushrooms.csv')
```

```
In [4]: pd.set_option('display.max_columns',None)
```

```
In [5]: dataset.head()
```

```
Out[5]:
```

	class	cap- shape	cap- surface	cap- color	bruises	odor	gill- attachment	gill- spacing	gill- size	gill- color	stalk- shape	stalk- root	stal surfac abov rii
0	p	x	s	n	t	p	f	c	n	k	e	e	
1	e	x	s	y	t	a	f	c	b	k	e	c	
2	e	b	s	w	t	l	f	c	b	n	e	c	
3	p	x	y	w	t	p	f	c	n	n	e	e	
4	e	x	s	g	f	n	f	w	b	k	t	e	

```
In [6]: dataset.tail()
```

Out[6]:

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	su a
8119	e	k	s	n	f	n	a	c	b	y	e	?	
8120	e	x	s	n	f	n	a	c	b	y	e	?	
8121	e	f	s	n	f	n	a	c	b	n	e	?	
8122	p	k	y	n	f	y	f	c	n	b	t	?	
8123	e	x	s	n	f	n	a	c	b	y	e	?	

In [7]: dataset.shape

Out[7]: (8124, 23)

```
In [10]: print('Number of Rows:',dataset.shape[0])
print('Number of Columns:',dataset.shape[1])
```

Number of Rows: 8124  
Number of Columns: 23

In [11]: dataset.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 [12]: dataset.isnull().sum()

```

Out[12]: 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 [13]: dataset.describe()
```

```
Out[13]:
```

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root
<b>count</b>	8124	8124	8124	8124	8124	8124	8124	8124	8124	8124	8124	8124
<b>unique</b>	2	6	4	10	2	9	2	2	2	12	2	5
<b>top</b>	e	x	y	n	f	n	f	c	b	b	t	b
<b>freq</b>	4208	3656	3244	2284	4748	3528	7914	6812	5612	1728	4608	3776

```
In [14]: dataset = dataset.astype('category')
```

```
In [15]: dataset.dtypes
```

```
Out[15]: class                category
cap-shape                category
cap-surface              category
cap-color                category
bruises                 category
odor                   category
gill-attachment         category
gill-spacing            category
gill-size               category
gill-color              category
stalk-shape             category
stalk-root              category
stalk-surface-above-ring category
stalk-surface-below-ring category
stalk-color-above-ring  category
stalk-color-below-ring  category
veil-type               category
veil-color              category
ring-number             category
ring-type               category
spore-print-color        category
population               category
habitat                 category
dtype: object
```

```
In [17]: le = LabelEncoder()
for column in dataset.columns:
    dataset[column]=le.fit_transform(dataset[column])
```

```
In [18]: dataset.head()
```

```
Out[18]:
```

	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
0	1	5	2	4	1	6	1	0	1	4	0	3	
1	0	5	2	9	1	0	1	0	0	4	0	2	
2	0	0	2	8	1	3	1	0	0	5	0	2	
3	1	5	3	8	1	6	1	0	1	5	0	3	
4	0	5	2	3	0	5	1	1	0	4	1	3	

```
In [19]: X = dataset.drop('class',axis=1)
y = dataset['class']
```

```
In [27]: pca1 = PCA(n_components=7)
pca_fit = pca1.fit_transform(X)
```

```
In [28]: pca1.explained_variance_ratio_
```

```
Out[28]: array([0.33758733, 0.16581098, 0.12285821, 0.06796611, 0.05831173,
0.05092539, 0.04670841])
```

```
In [29]: sum(pca1.explained_variance_ratio_)
```

```
Out[29]: 0.8501681635086222
```

```
In [32]: X_train,X_test,y_train,y_test = train_test_split(pca_fit,y,test_size=0.20,rand=
```

```
In [37]: lr = LogisticRegression()  
lr.fit(X_train,y_train)  
  
knn = KNeighborsClassifier()  
knn.fit(X_train,y_train)  
  
svc = SVC()  
svc.fit(X_train,y_train)  
  
dt = DecisionTreeClassifier()  
dt.fit(X_train,y_train)  
  
rm = RandomForestClassifier()  
rm.fit(X_train,y_train)  
  
gb = GradientBoostingClassifier()  
gb.fit(X_train,y_train)
```

```
Out[37]: ▾ GradientBoostingClassifier
          GradientBoostingClassifier()
```

```
In [38]: y_pred1 = lr.predict(X_test)
y_pred2 = knn.predict(X_test)
y_pred3 = svc.predict(X_test)
y_pred4 = dt.predict(X_test)
y_pred5 = rm.predict(X_test)
y_pred6 = gb.predict(X_test)
```

```
In [41]: print('Accuracy LogisticRegression:', accuracy_score(y_test, y_pred1))
print('Accuracy KNeighborsClassifier:', accuracy_score(y_test, y_pred2))
print('Accuracy SVC:', accuracy_score(y_test, y_pred3))
print('Accuracy DecisionTreeClassifier:', accuracy_score(y_test, y_pred4))
print('Accuracy RandomForestClassifier:', accuracy_score(y_test, y_pred5))
print('Accuracy GradientBoostingClassifier:', accuracy_score(y_test, y_pred6))
```

```
Accuracy LogisticRegression: 0.8344615384615385
Accuracy KNeighborsClassifier: 0.9833846153846154
Accuracy SVC: 0.952
Accuracy DecisionTreeClassifier: 0.976
Accuracy RandomForestClassifier: 0.9975384615384615
Accuracy GradientBoostingClassifier: 0.9384615384615385
```

[illegible]



```
In [59]: final_data
```

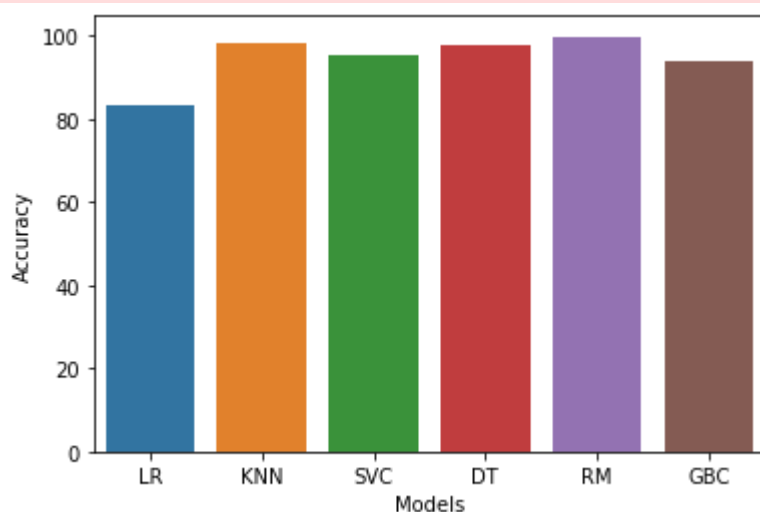
```
Out[59]:
```

	Models	Accuracy
0	LR	83.446154
1	KNN	98.338462
2	SVC	95.200000
3	DT	97.600000
4	RM	99.753846
5	GBC	93.846154

```
In [61]: sns.barplot(final_data['Models'],final_data['Accuracy'])  
plt.show()
```

C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```



```
In [62]: rf_model = RandomForestClassifier()  
rf_model.fit(pca_fit,y)
```

```
Out[62]:
```

RandomForestClassifier

RandomForestClassifier()

```
In [63]: joblib.dump(rf_model,'Mushroom_prediction')
```

```
Out[63]: ['Mushroom_prediction']
```

```
In [64]: model = joblib.load('Mushroom_prediction')
```

```
In [65]: p = model.predict(pca1.transform([[5,2,4,1,6,1,0,1,4,0,3,2,2,7,7,0,2,1,4,2,3,5]]))
```

```
C:\Users\prasad_jadhav\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but PCA was fitted with feature names
warnings.warn(
```

```
In [67]: if p[0]==1:
          print('Poissonous')
        else:
          print('Edible')
```

Poissonous

```
In [70]: def show_entry_fields():
          p1=int(e1.get())
          p2=int(e2.get())
          p3=int(e3.get())
          p4=int(e4.get())
          p5=int(e5.get())
          p6=int(e6.get())
          p7=int(e7.get())
          p8=int(e8.get())
          p9=int(e9.get())
          p10=int(e10.get())
          p11=int(e11.get())

          p12=int(e12.get())
          p13=int(e13.get())
          p14=int(e14.get())
          p15=int(e15.get())
          p16=int(e16.get())
          p17=int(e17.get())
          p18=int(e18.get())
          p19=int(e19.get())
          p20=int(e20.get())
          p21=int(e21.get())
          p22=int(e22.get())

          model = joblib.load('Mushroom_prediction')
          result=model.predict(pca1.transform([[p1,p2,p3,p4,p5,p6,
          p7,p8,p9,p10,p11,p12,p13,p14,p15,
          p16,p17,p18,p19,p20,p21,p22]]))

          if result[0] == 0:
              Label(master, text="Edible").grid(row=31)
          else:
              Label(master, text="Poisonous").grid(row=31)

          master = Tk()
          master.title("Mushroom Classification Using Machine Learning")

          label = Label(master, text = "Mushroom Classification Using Machine Learning"
              , bg = "black", fg = "white"). \
              grid(row=0,columnspan=2)

          Label(master,text="cap-shape :(cap-shape: bell=0,conical=1,convex=5,flat=2, knob=3) ").grid(
          Label(master, text="cap-surface:(fibrous=0,grooves=1,scaly=3,smooth=2)").grid(
          Label(master, text="cap-color:(brown=4,buff=0,cinnamon=1,gray=3,green=r, \
```

```

pink=5,purple=6,red=2,white=7,yellow=8)").grid(row=3)
Label(master, text="bruises:(bruises=1,no=0)").grid(row=4)
Label(master, text="odor:(almond=0,anise=3,creosote=1,fishy=8,foul=2,\
musty=4,none=5,pungent=6,spicy=7 \
)").grid(row=5)
Label(master, text="gill-attachment:(attached=0,descending=1,free=2,notched=3)'\
Label(master, text="gill-spacing:(close=0,crowded=2,distant=1 \
)").grid(row=7)
Label(master, text="gill-size:(road=0,narrow=1)").grid(row=8)
Label(master, text="gill-color:(black=4,brown=5,buff=0,chocolate=3,gray=2,green=2,orange=5,pink=6,red=2,white=7,yellow=8)").grid(row=9)
Label(master, text="stalk-shape:(enlarging=0,tapering=1)").grid(row=10)
Label(master, text="stalk-root:( bulbous=0,club=1,cup=5,equal=2,rhizomorphs=4, \
rooted=3,missing=6)").grid(row=11)

Label(master, text="stalk-surface-above-ring:(fibrous=0,scaly=3,silky=1,smooth=2)").grid(row=12)
Label(master, text="stalk-surface-below-ring:(fibrous=0,scaly=3,silky=1,smooth=2)").grid(row=13)
Label(master, text="stalk-color-above-ring:(brown=4,buff=0,cinnamon=1,gray=3, \
orange=5,pink=6,red=2,white=7,yellow=8)").grid(row=14)
Label(master, text="stalk-color-below-ring:(brown=4,buff=0,cinnamon=1,gray=3, \
orange=5,pink=6,red=2,white=7,yellow=8)").grid(row=15)
Label(master, text="veil-type:(partial=0,universal=1)").grid(row=16)
Label(master, text="veil-color:(brown=0,orange=1,white=2,yellow=3)").grid(row=17)
Label(master, text="ring-number:(none=0,one=1,two=2)").grid(row=18)
Label(master, text="ring-type:(cobwebby=0,evanescent=1,flaring=2,large=3, \
none=4,pendant=5,sheathing=6,zone=7)").grid(row=19)
Label(master, text="spore-print-color:(black=2,brown=3,buff=0,chocolate=1, \
green=5,orange=4,purple=6,white=7,yellow=8 \
)").grid(row=20)

Label(master, text="population:(abundant=0,clustered=1,numerous=2,scattered=3, \
# several=4,solitary=5)").grid(row=21)
Label(master, text="habitat:(grasses=1,leaves=2,meadows=3,paths=4,urban=5, \
# waste=6,woods=0)").grid(row=22)

e1 = Entry(master)
e2 = Entry(master)
e3 = Entry(master)
e4 = Entry(master)
e5 = Entry(master)
e6 = Entry(master)
e7 = Entry(master)
e8 = Entry(master)
e9 = Entry(master)
e10 = Entry(master)
e11 = Entry(master)

e12 = Entry(master)
e13 = Entry(master)
e14 = Entry(master)
e15 = Entry(master)
e16 = Entry(master)
e17 = Entry(master)
e18 = Entry(master)
e19 = Entry(master)
e20 = Entry(master)
e21 = Entry(master)
e22 = Entry(master)

```

```

e1.grid(row=1, column=1)
e2.grid(row=2, column=1)
e3.grid(row=3, column=1)
e4.grid(row=4, column=1)
e5.grid(row=5, column=1)
e6.grid(row=6, column=1)
e7.grid(row=7, column=1)
e8.grid(row=8, column=1)
e9.grid(row=9, column=1)
e10.grid(row=10, column=1)
e11.grid(row=11, column=1)

```

```

e12.grid(row=12, column=1)
e13.grid(row=13, column=1)
e14.grid(row=14, column=1)
e15.grid(row=15, column=1)
e16.grid(row=16, column=1)
e17.grid(row=17, column=1)
e18.grid(row=18, column=1)
e19.grid(row=19, column=1)
e20.grid(row=20, column=1)
e21.grid(row=21, column=1)
e22.grid(row=22, column=1)

```

```

Button(master, text='Predict', command=show_entry_fields).grid()

```

```

mainloop()

```

C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but PCA was fitted with feature names  
warnings.warn(

#### Mushroom Classification Using Machine Learning

##### Mushroom Classification Using Machine Learning

cap-shape:(cap-shape: bell=0,conical=1,convex=5,flat=2, knobbed=3,sunken=4)	5
cap-surface:(fibrous=0,grooves=1,scaly=3,smooth=2)	4
cap-color:(brown=4,buff=0,cinnamon=1,gray=3,green=r, pink=5,purple=6,red=2,white=7,yellow=8)	4
bruises:(bruises=1,no=0)	1
odor:(almond=0,anise=3,creosote=1,fishy=8,foul=2,musty=4,none=5,pungent=6,spicy=7)	6
gill-attachment:(attached=0,descending=1,free=2,notched=3)	1
gill-spacing:(close=0,crowded=2,distant=1)	0
gill-size:(road=0,narrow=1)	1
gill-color:(black=4,brown=5,buff=0,chocolate=3,gray=2,green=8,orange=6,pink=7,purple=9,red=1,white=10,yellow=11)	4
stalk-shape:(enlarging=0,tapering=1)	0
stalk-root:( bulbous=0,club=1,cup=5,equal=2,rhizomorphs=4, rooted=3,missing=6)	3
stalk-surface-above-ring:(fibrous=0,scaly=3,silky=1,smooth=2)	2
stalk-surface-below-ring:(fibrous=0,scaly=3,silky=1,smooth=2)	2
stalk-color-above-ring:(brown=4,buff=0,cinnamon=1,gray=3, orange=5,pink=6,red=2,white=7,yellow=8)	7
stalk-color-below-ring:(brown=4,buff=0,cinnamon=1,gray=3, orange=5,pink=6,red=2,white=7,yellow=8)	7
veil-type:(partial=0,universal=1)	0
veil-color:(brown=0,orange=1,white=2,yellow=3)	2
ring-number:(none=0,one=1,two=2)	1
ring-type:(cobwebby=0,evanescent=1,flaring=2,large=3,none=4,pendant=5,sheathing=6,zone=7)	4
spore-print-color:(black=2,brown=3,buff=0,chocolate=1, green=5,orange=4,purple=6,white=7,yellow=8)	2
population:(abundant=0,clustered=1,numerous=2,scattered=3, # several=4,solitary=5)	3
habitat:(grasses=1,leaves=2,meadows=3,paths=4,urban=5,# waste=6,woods=0)	5

Poisonous

CONNECT WITH ME:

[LinkedIn](#) [GitHub](#) [kaggle](#) [Medium](#)

---

PRASADMJADHAV2