Mushroom Classification -By Nithin Reddy

```
import pandas as pd
```

```
data=pd.read_csv("https://www.dropbox.com/s/81ggs49w6255qb5/MushroomClassification.csv?dl=1")
```

```
data
```

stalk- stalk- stalk-

```
data.columns
```

```
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')
```

```
data["class"].unique()
```

```
array(['p', 'e'], dtype=object)
```

```
data["class"]=data["class"].map({"p":0,"e":1})
```

```
8121    e    f    s    n    f    n    a    c    b    n  ...    s    o    o
```

```
data["cap-shape"].unique()
```

```
array(['x', 'b', 's', 'f', 'k', 'c'], dtype=object)
```

8124 rows × 23 columns

```
data["cap-shape"]=data["cap-shape"].map({"x":0,"b":1,"s":2,"f":3,"k":4,"c":5})
```

```
data["cap-surface"].unique()
```

```
array(['s', 'y', 'f', 'g'], dtype=object)
```

```
data["cap-surface"]=data["cap-surface"].map({"s":0,"y":1,"f":2,"g":3})
```

```
data['cap-color'].unique()
```

```
array(['n', 'y', 'w', 'g', 'e', 'p', 'b', 'u', 'c', 'r'], dtype=object)
```

```
data["cap-color"]=data["cap-color"].map({"n":0,"y":1,"w":2,"g":3,"e":4,"p":5,"b":6,"u":7,"c":8,"r":9})
```

```
data['bruises'].unique()
```

```
array(['t', 'f'], dtype=object)
```

```
data["bruises"]=data["bruises"].map({"t":0,"f":1})
```

```
data["odor"].unique()
```

```
array(['p', 'a', 'l', 'n', 'f', 'c', 'y', 's', 'm'], dtype=object)
```

```
data["odor"]=data["odor"].map({"p":0,"a":1,"l":2,"n":3,"f":4,"c":5,"y":6,"s":7,"m":8})
```

```
data['gill-attachment'].unique()
```

```
array(['f', 'a'], dtype=object)
```

```
data['gill-attachment']=data['gill-attachment'].map({"f":0,"a":1})
```

```
data['gill-spacing'].unique()
```

```
array(['c', 'w'], dtype=object)
```

```
data['gill-spacing']=data['gill-spacing'].map({"c":0,"w":1})
```

```
data['gill-size'].unique()
```

```
array(['n', 'b'], dtype=object)
```

```
data['gill-size']=data['gill-size'].map({"n":0,"b":1})
```

```
data['gill-color'].unique()
```

```
array(['k', 'n', 'g', 'p', 'w', 'h', 'u', 'e', 'b', 'r', 'y', 'o'],
      dtype=object)
```

```
data['gill-color']=data['gill-color'].map({"k":0,"n":1,"g":2,"p":3,"w":4,"h":5,"u":6,"e":7,"b":8,"r":9,"y":10,"o":11})
```

```
data['stalk-shape'].unique()
```

```
array(['e', 't'], dtype=object)
```

```
data['stalk-shape']=data['stalk-shape'].map({"e":0,"t":1})
```

```
data['stalk-root'].unique()
```

```
array(['e', 'c', 'b', 'r', '?'], dtype=object)
```

```
data['stalk-root']=data['stalk-root'].map({"e":0,"c":1,"b":2,"r":3,"?":4})
```

```
data['stalk-surface-above-ring'].unique()
```

```
array(['s', 'f', 'k', 'y'], dtype=object)
```

```
data['stalk-surface-above-ring']=data['stalk-surface-above-ring'].map({"s":0,"f":1,"k":2,"y":3})
```

```
data['stalk-surface-below-ring'].unique()
```

```
array(['s', 'f', 'y', 'k'], dtype=object)
```

```
data['stalk-surface-below-ring']=data['stalk-surface-below-ring'].map({"s":0,"f":1,"k":2,"y":3})
```

```
data['stalk-color-above-ring'].unique()
```

```
array(['w', 'g', 'p', 'n', 'b', 'e', 'o', 'c', 'y'], dtype=object)
```

```
data['stalk-color-above-ring']=data['stalk-color-above-ring'].map({"w":0,"g":1,"p":2,"n":3,"b":4,"e":5,"o":6,"c":7,"y":8})
```

```
data['stalk-color-below-ring'].unique()
```

```
array(['w', 'p', 'g', 'b', 'n', 'e', 'y', 'o', 'c'], dtype=object)
```

```
data['stalk-color-below-ring']=data['stalk-color-below-ring'].map({"w":0,"p":1,"g":2,"b":3,"n":4,"e":5,"y":6,"o":7,"c":8})
```

```
data['veil-type'].unique()
```

```
array(['p'], dtype=object)
```

```
data['veil-type']=data['veil-type'].map({"p":0})
```

```
data['veil-color'].unique()
```

```
array(['w', 'n', 'o', 'y'], dtype=object)
```

```
data['veil-color']=data['veil-color'].map({"w":0,"n":1,"o":2,"y":3})
```

```
data['ring-number'].unique()
```

```
array(['o', 't', 'n'], dtype=object)
```

```
data['ring-number']=data['ring-number'].map({"o":0,"t":1,"n":2})
```

```
data['ring-type'].unique()
```

```
array(['p', 'e', 'l', 'f', 'n'], dtype=object)
```

```
data['ring-type']=data['ring-type'].map({"p":0,"e":1,"l":2,"f":3,"n":4})
```

```
data['spore-print-color'].unique()
```

```
array(['k', 'n', 'u', 'h', 'w', 'r', 'o', 'y', 'b'], dtype=object)
```

```
data['spore-print-color']=data['spore-print-color'].map({"k":0,"n":1,"u":2,"h":3,"w":4,"r":5,"o":6,"y":7,"b":8})
```

```
data['population'].unique()
```

```
array(['s', 'n', 'a', 'v', 'y', 'c'], dtype=object)
```

```
data['population']=data['population']
```

```
data["population"]=data["population"].map({"s":0,"n":1,"a":2,"v":3,"y":4,"c":5})
```

```
data["habitat"].unique()
```

```
array(['u', 'g', 'm', 'd', 'p', 'w', 'l'], dtype=object)
```

```
data["habitat"]=data["habitat"].map({"u":0,"g":1,"m":2,"d":3,"p":4,"w":5,"l":6})
```

```
data
```

	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	ve ty
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
1	1	0	0	1	0	1	0	0	1	0	...	0	0	0	
2	1	1	0	2	0	2	0	0	1	1	...	0	0	0	
3	0	0	1	2	0	0	0	0	0	1	...	0	0	0	
4	1	0	0	3	1	3	0	1	1	0	...	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
8119	1	4	0	0	1	3	1	0	1	10	...	0	6	7	
8120	1	0	0	0	1	3	1	0	1	10		0	6	7	
8123	1	0	0	0	1	3	1	0	1	10	...	0	6	7	

```
x=data.drop("class",axis=1)
```

```
y=data["class"]
```

```
from sklearn.linear_model import LogisticRegression
```

```
model=LogisticRegression()
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,shuffle=True)
```

```
x_train.shape,y_train.shape,x_test.shape,y_test.shape
```

```
((6499, 22), (6499,), (1625, 22), (1625,))
```

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

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to conve
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
LogisticRegression()
```

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

```
from sklearn.metrics import accuracy_score
```

```
accuracy_score(y_test,predictions)
```

```
0.9876923076923076
```

Colab paid products - Cancel contracts here

✓ 0s completed at 9:00 PM