```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[ ] #Load dataset
    import pandas as pd
    mushroom_df = pd.read_csv("/content/drive/MyDrive/mushrooms.csv")
```

```
[ ] #Preprocessing
    mushroom_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
```

```
[ ] #Check for null
    print(mushroom_df.isnull().sum())
```

```
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
```

```
[ ] mushroom_df.head()
```

	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	ring-type	spore-print-color	po
0	p	x	s	n	t	p	f	c	n	k	...	s	w	w	p	w	o	p	k	
1	e	x	s	y	t	a	f	c	b	k	...	s	w	w	p	w	o	p	n	
2	e	b	s	w	t	l	f	c	b	n	...	s	w	w	p	w	o	p	n	
3	p	x	y	w	t	p	f	c	n	n	...	s	w	w	p	w	o	p	k	
4	e	x	s	g	f	n	f	w	b	k	...	s	w	w	p	w	o	e	n	

5 rows × 23 columns

[ ] 5 rows × 23 columns

[ ]
```
#mapping each letter to numerical value
from sklearn.preprocessing import LabelEncoder

mappings= list()
encoder= LabelEncoder()

for column in mushroom_df.columns:
    mushroom_df[column] = encoder.fit_transform(mushroom_df[column])
    mapping_dictionary = {index: label for index, label in enumerate(encoder.classes_)}
    mappings.append(mapping_dictionary)
mappings
```

```
[{0: 'e', 1: 'p'},
 {0: 'b', 1: 'c', 2: 'f', 3: 'k', 4: 's', 5: 'x'},
 {0: 'f', 1: 'g', 2: 's', 3: 'y'},
 {0: 'b',
  1: 'c',
  2: 'e',
  3: 'g',
  4: 'n',
  5: 'p',
  6: 'r',
  7: 'u',
  8: 'w',
  9: 'y'},
 {0: 'f', 1: 't'},
```

[ ]
```
 {0: 'a', 1: 'c', 2: 'f', 3: 'l', 4: 'm', 5: 'n', 6: 'p', 7: 's', 8: 'y'},
 {0: 'a', 1: 'f'},
 {0: 'c', 1: 'w'},
 {0: 'b', 1: 'n'},
 {0: 'b',
  1: 'e',
  2: 'g',
  3: 'h',
  4: 'k',
  5: 'n',
  6: 'o',
  7: 'p',
  8: 'r',
  9: 'u',
  10: 'w',
  11: 'y'},
 {0: 'e', 1: 't'},
 {0: '?', 1: 'b', 2: 'c', 3: 'e', 4: 'r'},
 {0: 'f', 1: 'k', 2: 's', 3: 'y'},
 {0: 'f', 1: 'k', 2: 's', 3: 'y'},
 {0: 'b', 1: 'c', 2: 'e', 3: 'g', 4: 'n', 5: 'o', 6: 'p', 7: 'w', 8: 'y'},
 {0: 'b', 1: 'c', 2: 'e', 3: 'g', 4: 'n', 5: 'o', 6: 'p', 7: 'w', 8: 'y'},
 {0: 'p'},
 {0: 'n', 1: 'o', 2: 'w', 3: 'y'},
 {0: 'n', 1: 'o', 2: 't'},
 {0: 'e', 1: 'f', 2: 'l', 3: 'n', 4: 'p'},
 {0: 'b', 1: 'h', 2: 'k', 3: 'n', 4: 'o', 5: 'r', 6: 'u', 7: 'w', 8: 'y'},
 {0: 'a', 1: 'c', 2: 'n', 3: 's', 4: 'v', 5: 'y'},
 {0: 'd', 1: 'g', 2: 'l', 3: 'm', 4: 'p', 5: 'u', 6: 'w'}]
```

[ ]
```
mushroom_df.head()
```

	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	ring-type	spore-print-color	po
0	1	5	2	4	1	6	1	0	1	4	...	2	7	7	0	2	1	4	2	
1	0	5	2	9	1	0	1	0	0	4	...	2	7	7	0	2	1	4	3	
2	0	0	2	8	1	3	1	0	0	5	...	2	7	7	0	2	1	4	3	
3	1	5	3	8	1	6	1	0	1	5	...	2	7	7	0	2	1	4	2	
4	0	5	2	3	0	5	1	1	0	4	...	2	7	7	0	2	1	0	3	

5 rows × 23 columns

[ ]
```
#Define target (y) and features (x)
x = mushroom_df.drop('class', axis=1)
y = mushroom_df['class']

#To see shape
x.shape, y.shape
```

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

```
[ ] #Scaling x values
    from sklearn.preprocessing import StandardScaler
    scaler= StandardScaler()
    x=pd.DataFrame(scaler.fit_transform(x), columns=x.columns)
    x
```

	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color
0	1.029712	0.140128	-0.198250	1.185917	0.881938	0.162896	-0.438864	1.494683	-0.228998	-1.144806	...	0.586385	0.622441	0.631991	0.0	0.142037
1	1.029712	0.140128	1.765874	1.185917	-1.970316	0.162896	-0.438864	-0.669038	-0.228998	-1.144806	...	0.586385	0.622441	0.631991	0.0	0.142037
2	-2.087047	0.140128	1.373049	1.185917	-0.544189	0.162896	-0.438864	-0.669038	0.053477	-1.144806	...	0.586385	0.622441	0.631991	0.0	0.142037
3	1.029712	0.953270	1.373049	1.185917	0.881938	0.162896	-0.438864	1.494683	0.053477	-1.144806	...	0.586385	0.622441	0.631991	0.0	0.142037
4	1.029712	0.140128	-0.591075	-0.843230	0.406562	0.162896	2.278612	-0.669038	-0.228998	0.873511	...	0.586385	0.622441	0.631991	0.0	0.142037
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8119	-0.216992	0.140128	-0.198250	-0.843230	0.406562	-6.138869	-0.438864	-0.669038	1.748325	-1.144806	...	0.586385	-0.429288	-0.416681	0.0	-3.979055
8120	1.029712	0.140128	-0.198250	-0.843230	0.406562	-6.138869	-0.438864	-0.669038	1.748325	-1.144806	...	0.586385	-0.429288	-0.416681	0.0	-8.100146
8121	-0.840343	0.140128	-0.198250	-0.843230	0.406562	-6.138869	-0.438864	-0.669038	0.053477	-1.144806	...	0.586385	-0.429288	-0.416681	0.0	-3.979055
8122	-0.216992	0.953270	-0.198250	-0.843230	1.832689	0.162896	-0.438864	1.494683	-1.358896	0.873511	...	-0.893053	0.622441	0.631991	0.0	0.142037
8123	1.029712	0.140128	-0.198250	-0.843230	0.406562	-6.138869	-0.438864	-0.669038	1.748325	-1.144806	...	0.586385	-0.429288	-0.416681	0.0	-3.979055

```
[10] #Train, test splits
     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, random_state=0)

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

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

```
[11] #Decision Tree Algorithm

     from sklearn.tree import DecisionTreeClassifier
     from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, recall_score, precision_score

     # Train Decision Tree model
     dt_model = DecisionTreeClassifier(random_state=0)
     dt_model.fit(x_train, y_train)

     # Prediction
     y_pred_dt = dt_model.predict(x_test)

     # Evaluation
     print("Decision Tree Accuracy:", accuracy_score(y_test, y_pred_dt))
     print("Sensitivity (Recall):", recall_score(y_test, y_pred_dt))
     print("Specificity:", precision_score(y_test, y_pred_dt))
     print("\nClassification Report:\n\n", classification_report(y_test, y_pred_dt))
     print("\nConfusion Matrix:\n\n", confusion_matrix(y_test, y_pred_dt))
```

```
Decision Tree Accuracy: 1.0
Sensitivity (Recall): 1.0
Specificity: 1.0

Classification Report:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       852
           1       1.00      1.00      1.00       773

    accuracy                           1.00      1625
   macro avg       1.00      1.00      1.00      1625
weighted avg       1.00      1.00      1.00      1625


Confusion Matrix:

 [[852   0]
 [  0 773]]
```

```python
#KNN Algorithm
from sklearn.neighbors import KNeighborsClassifier

# Tain the KNN model
knn_model = KNeighborsClassifier()
knn_model.fit(x_train, y_train)

# Prediction
y_pred_knn = knn_model.predict(x_test)

# Evaluation
print("KNN Accuracy:", accuracy_score(y_test, y_pred_knn))
print("Sensitivity (Recall):", recall_score(y_test, y_pred_knn))
print("Specificity:", precision_score(y_test, y_pred_knn))
print("\nClassification Report:\n\n", classification_report(y_test, y_pred_knn))
print("\nConfusion Matrix:\n\n", confusion_matrix(y_test, y_pred_knn))
```

```
KNN Accuracy: 1.0
Sensitivity (Recall): 1.0
Specificity: 1.0

Classification Report:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       852
           1       1.00      1.00      1.00       773

    accuracy                           1.00      1625
   macro avg       1.00      1.00      1.00      1625
weighted avg       1.00      1.00      1.00      1625


Confusion Matrix:

 [[852   0]
 [  0 773]]
```

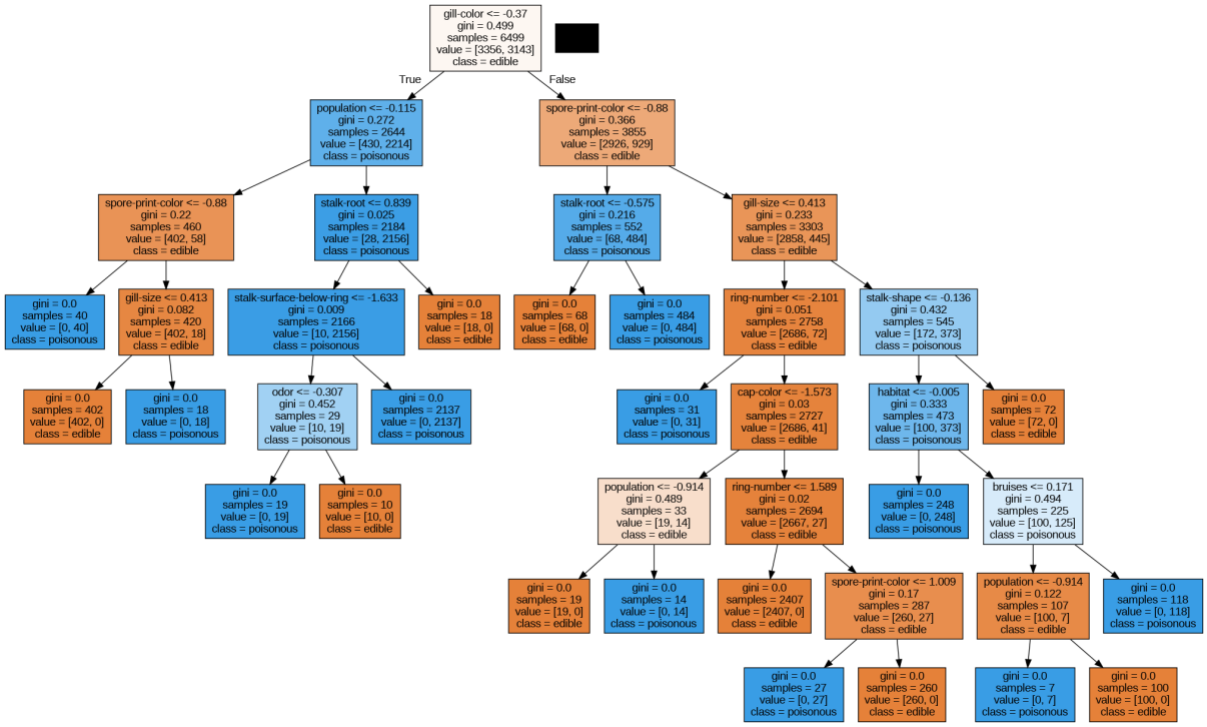
```python
[13] #drawing decision tree
from sklearn.tree import export_graphviz
from IPython.display import Image
import pydotplus

# Export Decision Tree as DOT file
dot_data = export_graphviz(
    dt_model,
    out_file=None,
    feature_names=x.columns,
    class_names=['edible', 'poisonous'],
    filled=True,
    rounded=False
)

#PyDotPlus graph from DOT data
graph = pydotplus.graph_from_dot_data(dot_data)

# Save
graph.write_png("/content/drive/MyDrive/decision_tree.png")

# Display
Image(graph.create_png())
```


gill-color <= -0.37
gini = 0.499
samples = 6499
value = [3356, 3143]
class = edible
True
False
population <= -0.115
gini = 0.272
samples = 2644
value = [430, 2214]
class = poisonous
spore-print-color <= -0.88
gini = 0.366
samples = 3855
value = [2926, 929]
class = edible
spore-print-color <= -0.88
gini = 0.22
samples = 460
value = [402, 58]
class = edible
stalk-root <= 0.839
gini = 0.025
samples = 2184
value = [28, 2156]
class = poisonous
stalk-root <= -0.575
gini = 0.216
samples = 552
value = [68, 484]
class = poisonous
gill-size <= 0.413
gini = 0.233
samples = 3303
value = [2858, 445]
class = edible
gini = 0.0
samples = 40
value = [0, 40]
class = poisonous
gill-size <= 0.413
gini = 0.082
samples = 420
value = [402, 18]
class = edible
stalk-surface-below-ring <= -1.633
gini = 0.009
samples = 2166
value = [10, 2156]
class = poisonous
gini = 0.0
samples = 18
value = [18, 0]
class = edible
gini = 0.0
samples = 68
value = [68, 0]
class = edible
gini = 0.0
samples = 484
value = [0, 484]
class = poisonous
ring-number <= -2.101
gini = 0.051
samples = 2758
value = [2686, 72]
class = edible
stalk-shape <= -0.136
gini = 0.432
samples = 545
value = [172, 373]
class = poisonous
gini = 0.0
samples = 402
value = [402, 0]
class = edible
gini = 0.0
samples = 18
value = [0, 18]
class = poisonous
odor <= -0.307
gini = 0.452
samples = 29
value = [10, 19]
class = poisonous
gini = 0.0
samples = 2137
value = [0, 2137]
class = poisonous
gini = 0.0
samples = 31
value = [0, 31]
class = poisonous
cap-color <= -1.573
gini = 0.03
samples = 2727
value = [2686, 41]
class = edible
habitat <= -0.005
gini = 0.333
samples = 473
value = [100, 373]
class = poisonous
gini = 0.0
samples = 72
value = [72, 0]
class = edible
gini = 0.0
samples = 19
value = [0, 19]
class = poisonous
gini = 0.0
samples = 10
value = [10, 0]
class = edible
population <= -0.914
gini = 0.489
samples = 33
value = [19, 14]
class = edible
ring-number <= 1.589
gini = 0.02
samples = 2694
value = [2667, 27]
class = edible
gini = 0.0
samples = 248
value = [0, 248]
class = poisonous
bruises <= 0.171
gini = 0.494
samples = 225
value = [100, 125]
class = poisonous
gini = 0.0
samples = 19
value = [19, 0]
class = edible
gini = 0.0
samples = 14
value = [0, 14]
class = poisonous
gini = 0.0
samples = 2407
value = [2407, 0]
class = edible
spore-print-color <= 1.009
gini = 0.17
samples = 287
value = [260, 27]
class = edible
population <= -0.914
gini = 0.122
samples = 107
value = [100, 7]
class = edible
gini = 0.0
samples = 118
value = [0, 118]
class = poisonous
gini = 0.0
samples = 27
value = [0, 27]
class = poisonous
gini = 0.0
samples = 260
value = [260, 0]
class = edible
gini = 0.0
samples = 7
value = [0, 7]
class = poisonous
gini = 0.0
samples = 100
value = [100, 0]
class = edible