Title: Mushroom classification

Code:

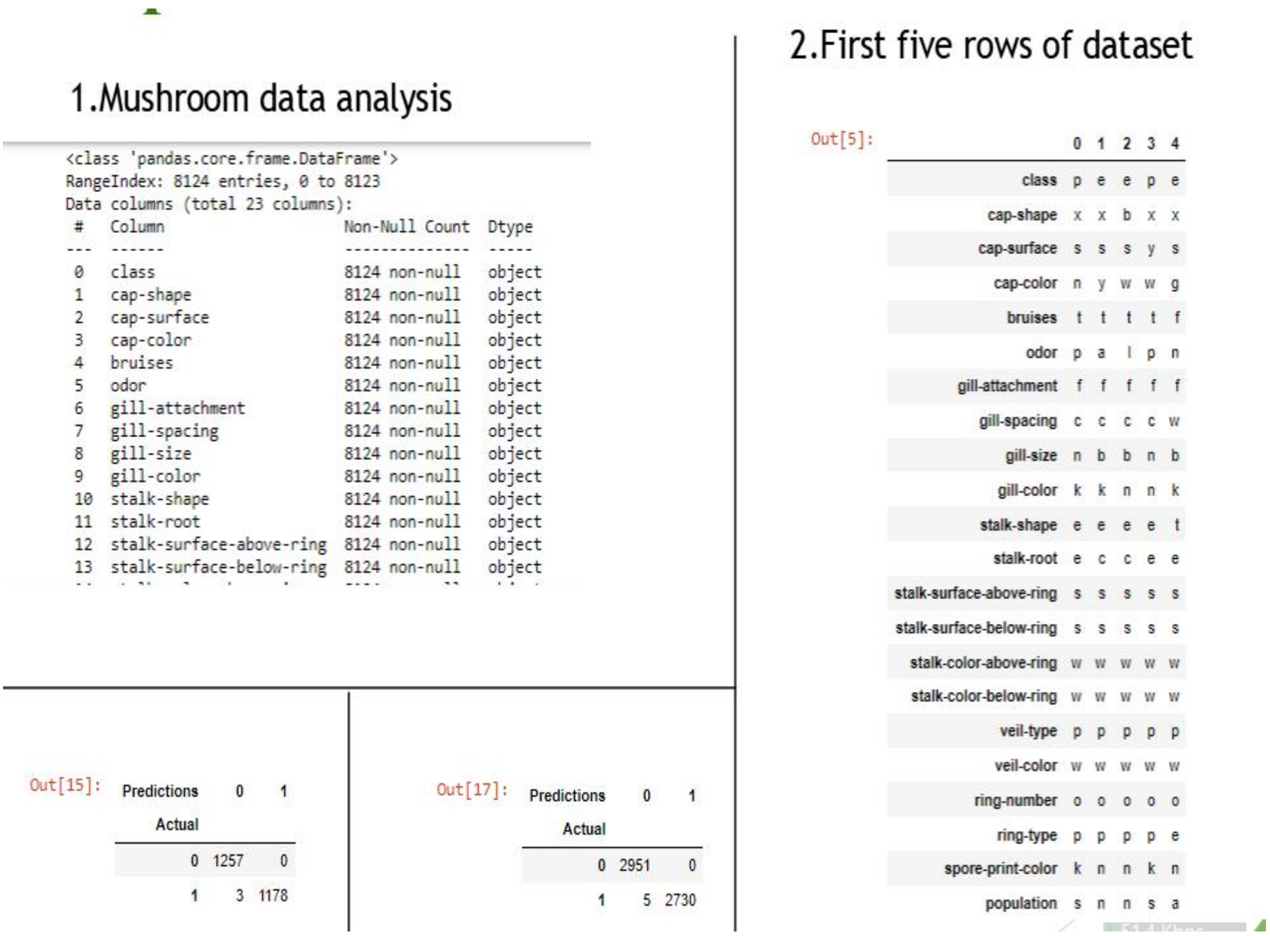
```
import numpy as np # linear algebra
import pandas as pd
import os
for dirname, _, filenames in os.walk('/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
mushroom = pd.read_csv('mushrooms.csv')
mushroom.info()
mushroom.head(5).transpose()
mushroom.isna().sum()
mushroom.duplicated().sum()
for i in mushroom.columns:
    print(mushroom[str(i)].value_counts())
    print('\n')
mushroom_en = pd.get_dummies(mushroom, drop_first = True)
mushroom_en.shape
mushroom_en.columns
X_feature = list(mushroom_en.columns)
X_feature.remove('class_p')
X = mushroom_en[X_feature]
X.columns
Y = mushroom_en['class_p']
Y.head()
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,Y, test_size =
0.30, random_state = 42)
print("X_train.shape",X_train.shape)
print("X_test.shape",X_test.shape)
print("y_train.shape",y_train.shape)
print("y_test.shape",y_test.shape)
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 21)
knn.fit(X_train, y_train)

#predicting for test data
knn_test_pred = knn.predict(X_test)
#evaluate the model
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, knn_test_pred))
pd.crosstab(y_test, knn_test_pred, rownames = ['Actual'], colnames=
['Predictions'])
#error on train data
knn_train_pred = knn.predict(X_train)
print(accuracy_score(y_train, knn_train_pred))
```

```
pd.crosstab(y_train, knn_train_pred, rownames = ['Actual'], colnames =
['Predictions'])
acc = []

# running KNN algorithm for 3 to 50 nearest neighbours(odd numbers) and
# storing the accuracy values
for i in range(3,50,2):
    neigh = KNeighborsClassifier(n_neighbors=i)
    neigh.fit(X_train, y_train)
    train_acc = np.mean(neigh.predict(X_train) == y_train)
    test_acc = np.mean(neigh.predict(X_test) == y_test)
    acc.append([train_acc, test_acc])
import matplotlib.pyplot as plt # library to do visualizations
# train accuracy plot
plt.plot(np.arange(3,50,2),[i[0] for i in acc],"ro-")
```

Output:



1.Mushroom data analysis

```
<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
```

2.First five rows of dataset

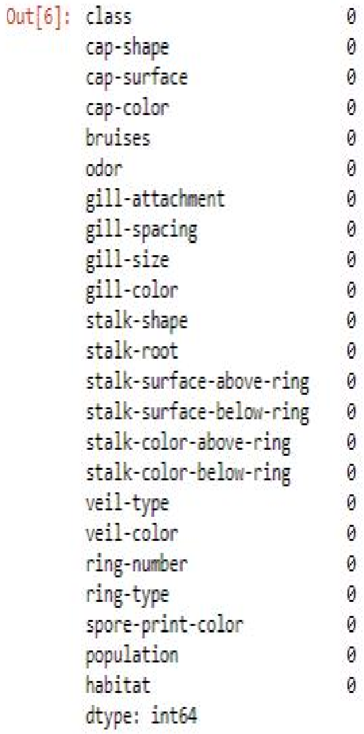
Out[5]:

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

```
Out[15]: Predictions     0     1
              Actual
                   0  1257     0
                   1     3  1178
```

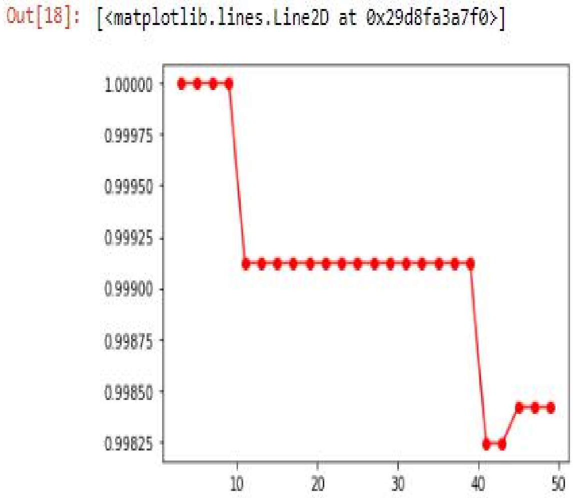
```
Out[17]: Predictions     0     1
              Actual
                   0  2951     0
                   1     5  2730
```

3. Checking for null values



```
Out[6]: 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
```

4. Accuracy graph



```
Out[18]: [<matplotlib.lines.Line2D at 0x29d8fa3a7f0>]
```