**CODE:**

#Imports

import pandas as pd

import numpy as np

import seaborn as sns

from sklearn.cluster import KMeans

from sklearn.preprocessing import LabelEncoder

from sklearn.linear\_model import LogisticRegression

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

orig = pd.read\_csv('../input/mushrooms.csv')

#Shuffles the orig DataFrame

#orig = orig.sample(frac=1)

orig.head()

X = orig.drop(['class'], axis=1)

y = orig['class']

for attr in X.columns:

print('\n\*', attr, '\*')

print(X[attr].value\_counts())

X.drop(['veil-type'], axis=1, inplace=True)

for attr in X.columns:

fig, ax =plt.subplots(1,2)

sns.countplot(X[X['stalk-root']=='?'][attr], ax=ax[0]).set\_title('stalk-root = ?')

sns.countplot(X[X['stalk-root']!='?'][attr], ax=ax[1]).set\_title('stalk-root != ?')

fig.show()

#For columns with only two values

for col in X.columns:

if len(X[col].value\_counts()) == 2:

le = LabelEncoder()

X[col] = le.fit\_transform(X[col])

X.head()

X = pd.get\_dummies(X)

X.head()

#New

#train\_X, val\_X, train\_y, val\_y = train\_test\_split(X, y, test\_size=0.0)

#New (used to show train\_X is indeed the same as X, albeit, shuffled)

#print(len(X))

#print(len(train\_X), len(train\_y))

#print(len(val\_X), len(val\_y))

kmeans = KMeans(n\_clusters=2, random\_state=None)

#Old

kmeans.fit(X)

#New

#kmeans.fit(train\_X)

#Old

clusters = kmeans.predict(X)

#New

#clusters = kmeans.predict(train\_X)

clusters

cluster\_df = pd.DataFrame()

cluster\_df['cluster'] = clusters

#Old

cluster\_df['class'] = y

#New

#cluster\_df['class'] = train\_y

sns.factorplot(col='cluster', y=None, x='class', data=cluster\_df, kind='count')