K\_NEAREST\_NEIGHBOURS:

**FOR ZOO.csv dataset:**

**Implement a KNN model to classify the animals in to categorie**

**Coding:**

# -\*- coding: utf-8 -\*-

"""

Created on Wed Nov 4 17:50:51 2020

@author: sunil

"""

import pandas as pd

import numpy as np

zoo=pd.read\_csv("Zoo.csv")

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

train,test=train\_test\_split(zoo,test\_size=0.2,random\_state=0)

from sklearn.neighbors import KNeighborsClassifier as KNC

neigh=KNC(n\_neighbors=3)

neigh.fit(train.iloc[:,1:17],train.iloc[:,17])

#training accuracy

train\_accuracy=np.mean(neigh.predict(train.iloc[:,1:17])==train.iloc[:,17])

test\_accuracy=np.mean(neigh.predict(test.iloc[:,1:17])==test.iloc[:,17])

neigh=KNC(n\_neighbors=5)

neigh.fit(train.iloc[:,1:17],train.iloc[:,17])

train\_accuracy=np.mean(neigh.predict(train.iloc[:,1:17])==train.iloc[:,17])

test\_accuracy=np.mean(neigh.predict(test.iloc[:,1:17])==test.iloc[:,17])

emp=[]

for i in range(3,50,2):

neigh=KNC(n\_neighbors=i)

neigh.fit(train.iloc[:,1:17],train.iloc[:,17])

train\_accuracy=np.mean(neigh.predict(train.iloc[:,1:17])==train.iloc[:,17])

test\_accuracy=np.mean(neigh.predict(test.iloc[:,1:17])==test.iloc[:,17])

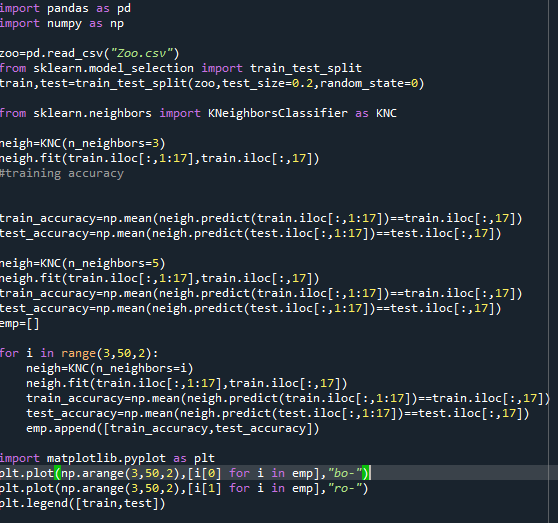
emp.append([train\_accuracy,test\_accuracy])

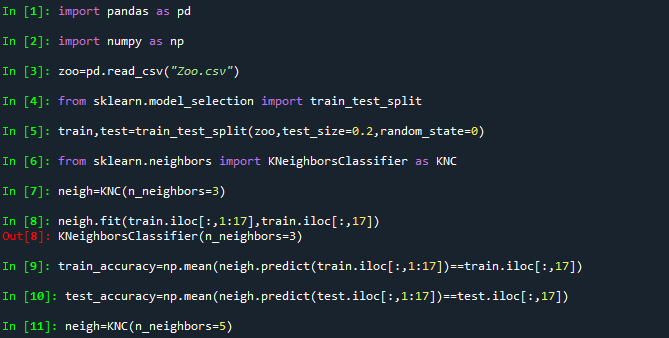
import matplotlib.pyplot as plt

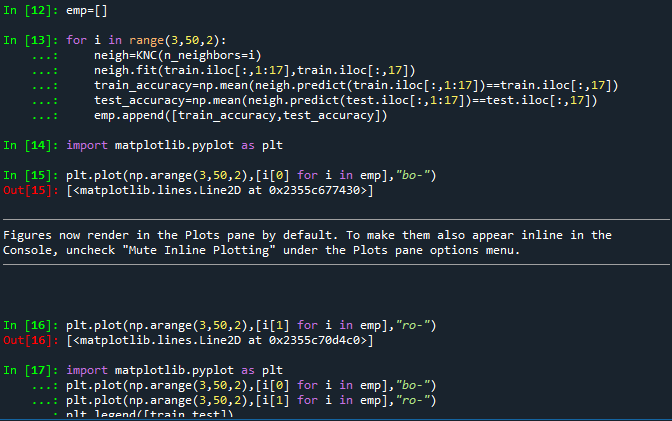
plt.plot(np.arange(3,50,2),[i[0] for i in emp],"bo-")

plt.plot(np.arange(3,50,2),[i[1] for i in emp],"ro-")

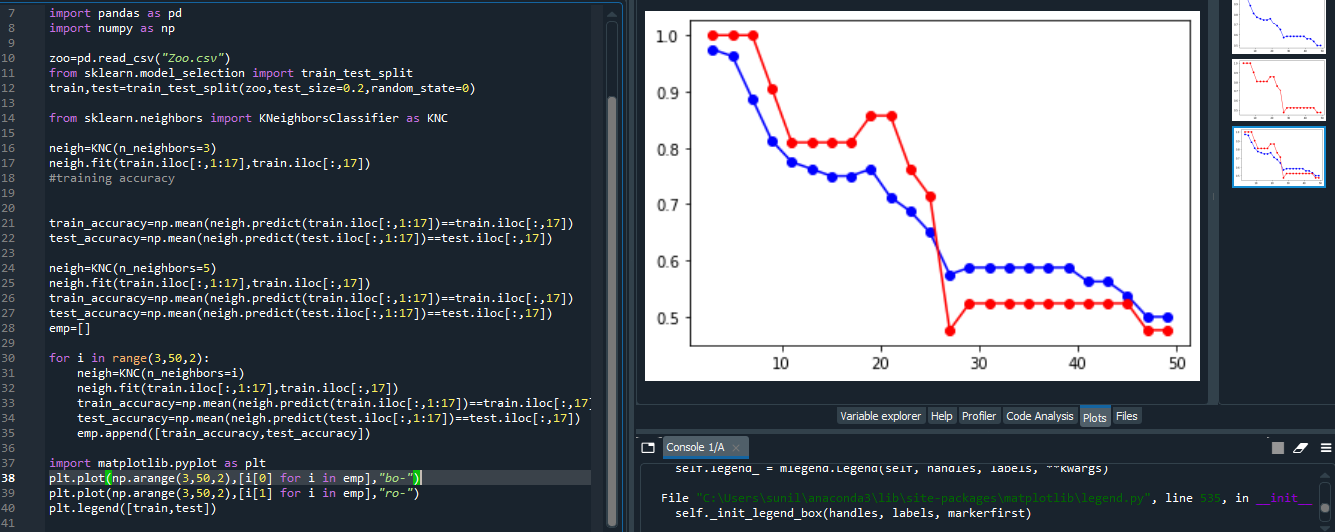
plt.legend([train,test])

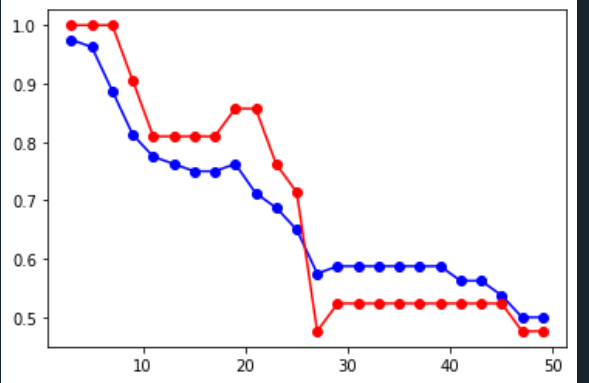






ZOO DATASET:





**FOR “GLASS.csv” dataset.**

**Prepare a model for glass classification using KNN**

**Coding:**

# -\*- coding: utf-8 -\*-

"""

Created on Wed Nov 4 17:50:51 2020

@author: sunil

"""

import pandas as pd

import numpy as np

zoo=pd.read\_csv("Zoo.csv")

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

train,test=train\_test\_split(zoo,test\_size=0.2,random\_state=0)

from sklearn.neighbors import KNeighborsClassifier as KNC

neigh=KNC(n\_neighbors=3)

neigh.fit(train.iloc[:,1:17],train.iloc[:,17])

#training accuracy

train\_accuracy=np.mean(neigh.predict(train.iloc[:,1:17])==train.iloc[:,17])

test\_accuracy=np.mean(neigh.predict(test.iloc[:,1:17])==test.iloc[:,17])

neigh=KNC(n\_neighbors=5)

neigh.fit(train.iloc[:,1:17],train.iloc[:,17])

train\_accuracy=np.mean(neigh.predict(train.iloc[:,1:17])==train.iloc[:,17])

test\_accuracy=np.mean(neigh.predict(test.iloc[:,1:17])==test.iloc[:,17])

emp=[]

for i in range(3,50,2):

neigh=KNC(n\_neighbors=i)

neigh.fit(train.iloc[:,1:17],train.iloc[:,17])

train\_accuracy=np.mean(neigh.predict(train.iloc[:,1:17])==train.iloc[:,17])

test\_accuracy=np.mean(neigh.predict(test.iloc[:,1:17])==test.iloc[:,17])

emp.append([train\_accuracy,test\_accuracy])

import matplotlib.pyplot as plt

plt.plot(np.arange(3,50,2),[i[0] for i in emp],"bo-")

plt.plot(np.arange(3,50,2),[i[1] for i in emp],"ro-")

plt.legend([train,test])

