MACHINE LEARNING (CSE 4020)

LAB UPLOAD 2

NAME: Harshit Kedia

REG. NO. : 15BCE0329

FACULTY: JAISAKTHI S M

Code:-

```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.cluster import KMeans
from sklearn import metrics
import pandas as pd
import numpy as np



iris = datasets.load_iris()
iris.data
iris.feature_names
iris.target
iris.target_names

x = pd.DataFrame(iris.data)
x.columns = ['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width']

y = pd.DataFrame(iris.target)
y.columns = ['Targets']




plt.figure(figsize=(14,7))



colormap = np.array(['red', 'lime', 'black'])

# Plot Sepal
plt.subplot(1, 2, 1)
```

```
plt.scatter(x.Sepal_Length, x.Sepal_Width, c=colormap[y.Targets], s=40)
plt.title('Sepal')

plt.subplot(1, 2, 2)
plt.scatter(x.Petal_Length, x.Petal_Width, c=colormap[y.Targets], s=40)
plt.title('Petal')
# K Means Cluster
model = KMeans(n_clusters=3)
model.fit(x)

model.labels_

plt.figure(figsize=(14,7))



colormap = np.array(['red', 'lime', 'black'])

# Plot the true Classifications
plt.subplot(1, 2, 1)
plt.scatter(x.Petal_Length, x.Petal_Width, c=colormap[y.Targets], s=40)
plt.title('Real Classification')

# Plot the predicted Classifications
plt.subplot(1, 2, 2)
plt.scatter(x.Petal_Length, x.Petal_Width, c=colormap[model.labels_], s=40)
plt.title('K Mean Classification')

predY = np.choose(model.labels_, [1, 0, 2]).astype(np.int64)



print("the sample labels(species) are:")
print (model.labels_)
print("the k-means labels(species) are:")
print (predY)

plt.figure(figsize=(14,7))
```

```
colormap = np.array(['red', 'lime', 'black'])



plt.subplot(1, 2, 1)
plt.scatter(x.Petal_Length, x.Petal_Width, c=colormap[y.Targets], s=40)
plt.title('Real Classification')

plt.subplot(1, 2, 2)
plt.scatter(x.Petal_Length, x.Petal_Width, c=colormap[predY], s=40)
plt.title('K Mode Classification')

 # Performance Metrics

print("the accuracy is:")
a=metrics.accuracy_score(y, predY)
print(a)

print("the error rate is:")
print(1-a)

print("the confusion matrix is:")
b=metrics.confusion_matrix(y, predY)
print(b)

print("the f1 score is:")
d=metrics.f1_score(y, predY)
print(d)

print("the adjusted rand score is:")
c=metrics.adjusted_rand_score(y,predY)
print(c)
```

OUTPUT:-

Spyder (Python 3.5)

File Edit Search Source Run Debug Consoles Tools View Help

C:\Users\HP

IPython console

Console 2/A

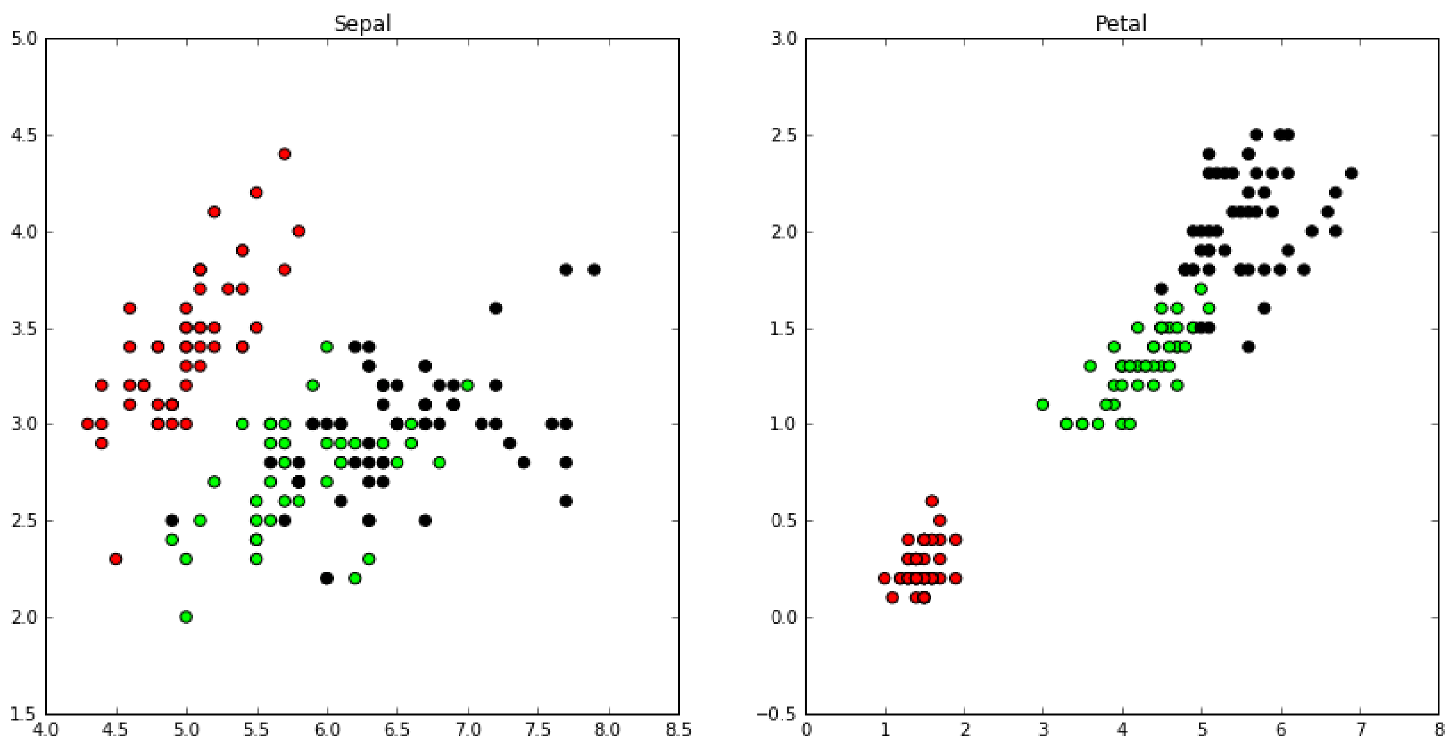
```
Python 3.5.2 |Anaconda 4.1.1 (64-bit)| (default, Jul  5 2016, 11:41:13) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 4.2.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
%guiref   -> A brief reference about the graphical user interface.

In [1]: runfile('C:/Users/HP/kmeans-from-scratch-and-sklearn.py', wdir='C:/Users/HP')
the sample labels(species) are:
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 2 0 0 0 0 2 0 0 0 0
 0 0 2 2 0 0 0 0 2 0 2 0 2 0 0 2 2 0 0 0 0 0 2 0 0 0 0 2 0 0 0 2 0 0 0 2 0
 0 2]
the k-means labels(species) are:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1
 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1
 1 2]
the accuracy is:
0.44
the error rate is:
0.56
the confusion matrix is:
[[50  0  0]
 [ 0  2 48]
 [ 0 36 14]]
the f1 score is:
0.431818181818
the adjusted rand score is:
0.74
C:\Users\HP\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:756: DeprecationWarning: The default `weighted` averaging is deprecated, and from version 0.18, use of precision,
recall or F-score with multiclass or multilabel data or pos_label=None will result in an exception. Please set an explicit value for `average`, one of (None, 'micro', 'macro', 'weighted',
'samples'). In cross validation use, for instance, scoring="f1_weighted" instead of scoring="f1".
  sample_weight=sample_weight)
```

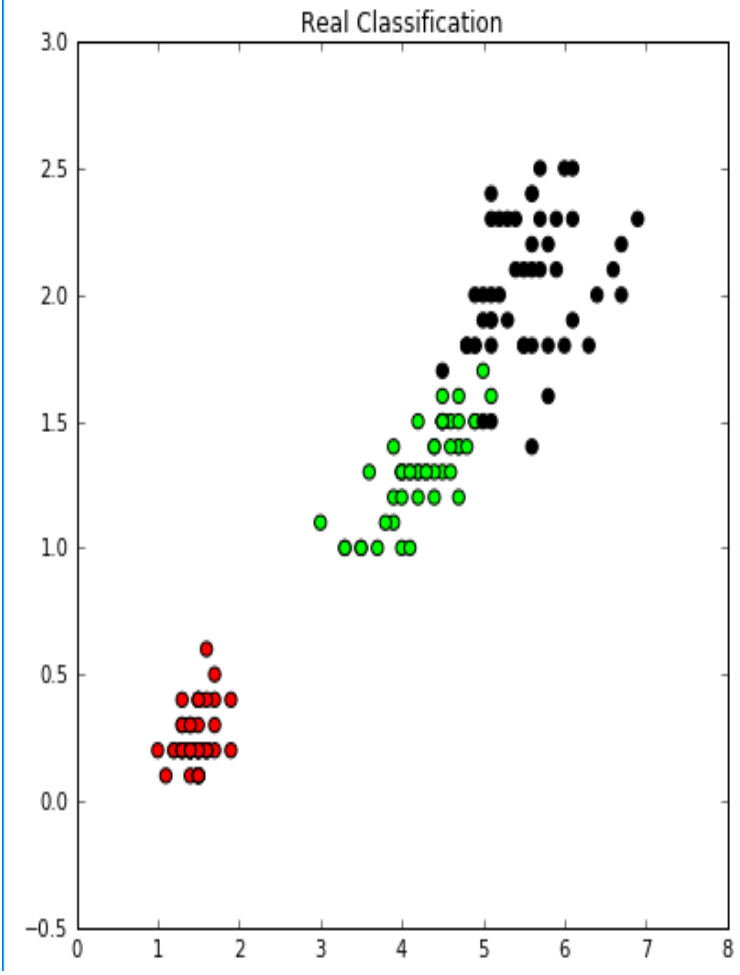
IPython console | Editor - C:\Users\HP\kmeans-from-scratch-and-sklearn.py

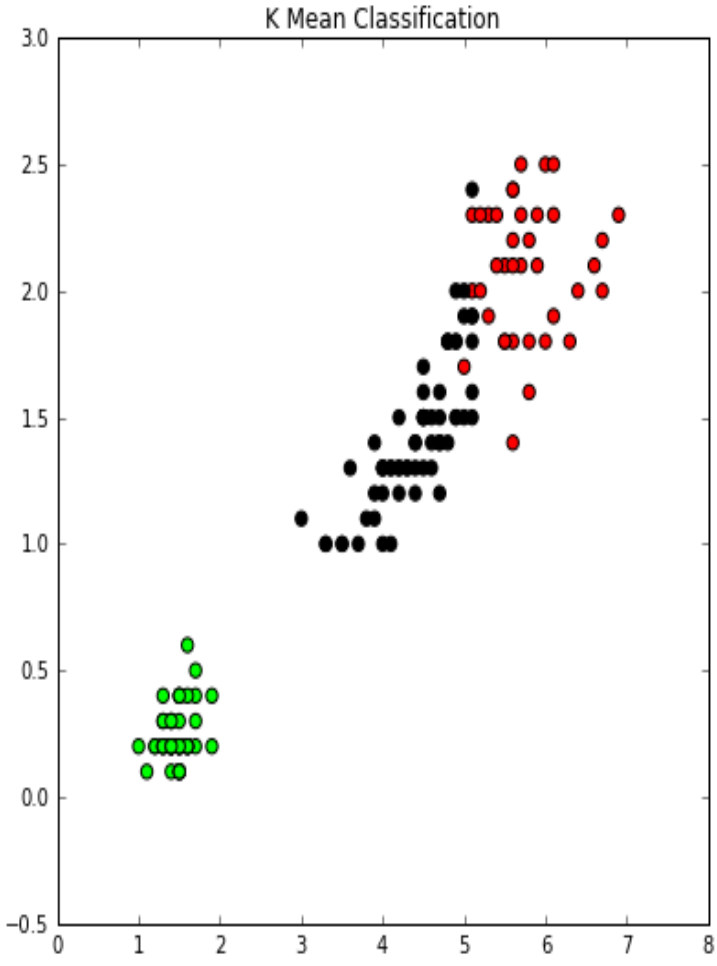
Permissions: RW End-of-lines: LF Encoding: ASCII Line: 1 Column: 1 Memory: 54 %

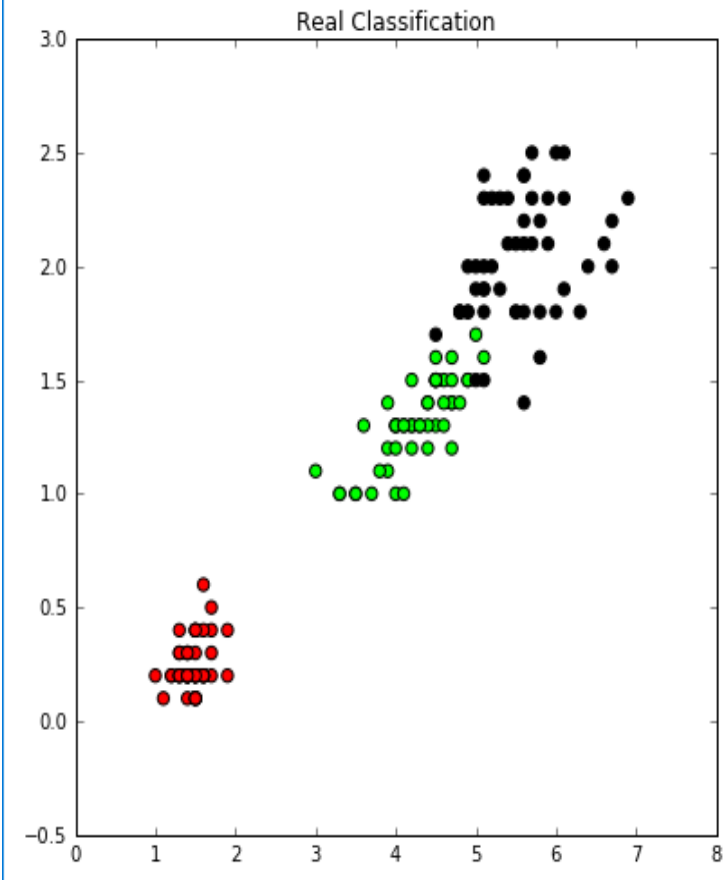
Console 2/A

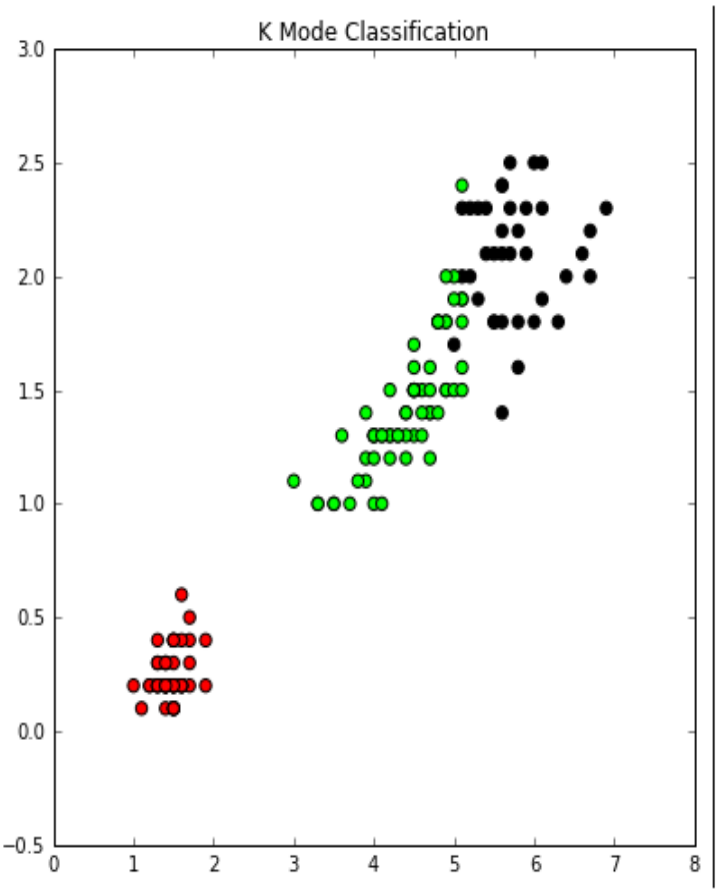

```
 [ 0 36 14]]
the f1 score is:
0.431818181818
the adjusted rand score is:
0.74
C:\Users\HP\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:756: DeprecationWarning: The default `weighted` aver
recall or F-score with multiclass or multilabel data or pos_label=None will result in an exception. Please set an explicit val
'samples'). In cross validation use, for instance, scoring="f1_weighted" instead of scoring="f1".
  sample_weight=sample_weight)
```


Real Classification


K Mean Classification


Real Classification


K Mode Classification