In [1]:

```
# maha ebrahim mohammed
# 4051350
# IA8G
# Lab 7: K-means Clustering
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import KMeans
```

In [2]:

```
from sklearn import datasets
iris= datasets.load_iris()
df= pd.read_csv(iris.filename)
```

In [3]:

```
x = df.iloc[:, [0,1,2,3]].values
```

In [4]:

```
kmeans5= KMeans(n_clusters=5)
y_kmeans5=kmeans5.fit_predict(x)
print(y_kmeans5)
kmeans5.cluster_centers_
```

```
[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 2 4 2 2 2 4 2 4 4 2 4 2 4 2 2 4 2 4 2 4 2 2
 2 2 2 2 2 4 4 4 4 2 4 2 2 2 4 4 4 2 4 4 4 4 4 2 4 4 0 2 3 0 0 3 4 3 0 3 0
 0 0 2 0 0 0 3 3 2 0 2 3 2 0 3 2 2 0 3 3 3 0 2 2 3 0 0 2 0 0 0 2 0 0 0 2 0
 0 2]
```

Out[4]:

```
array([[6.52916667, 3.05833333, 5.50833333, 2.1625    ],
       [5.006     , 3.428     , 1.462     , 0.246     ],
       [6.20769231, 2.85384615, 4.74615385, 1.56410256],
       [7.475     , 3.125     , 6.3       , 2.05      ],
       [5.508     , 2.6       , 3.908     , 1.204     ]])
```

In [5]:

```
kmeans2= KMeans(n_clusters=2)
y_kmeans2=kmeans2.fit_predict(x)
print(y_kmeans2)
kmeans2.cluster_centers_
```

```
[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 0 0 0 0 0 0 0 1 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 1 0 0 0 0 1 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 0]
```

Out[5]:

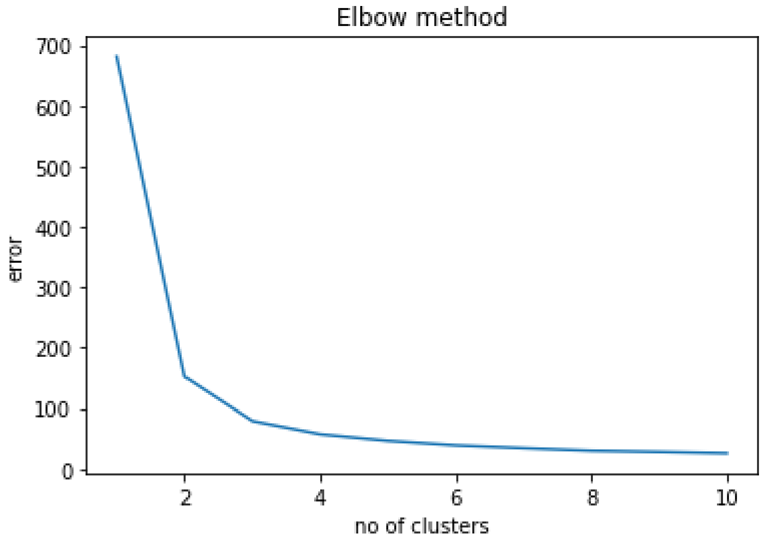
```
array([[6.30103093, 2.88659794, 4.95876289, 1.69587629],
       [5.00566038, 3.36981132, 1.56037736, 0.29056604]])
```

In [7]:

```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

Error =[]
for i in range(1,11):
 kmeans= KMeans(n_clusters=i).fit(x)
 kmeans.fit(x)
 Error.append(kmeans.inertia_)
plt.plot(range(1,11),Error)
plt.title('Elbow method')
plt.xlabel('no of clusters')
plt.ylabel('error')
plt.show()
```

```
C:\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:881: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are le
ss chunks than available threads. You can avoid it by setting the environmen
t variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:881: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are le
ss chunks than available threads. You can avoid it by setting the environmen
t variable OMP_NUM_THREADS=1.
  warnings.warn(
```

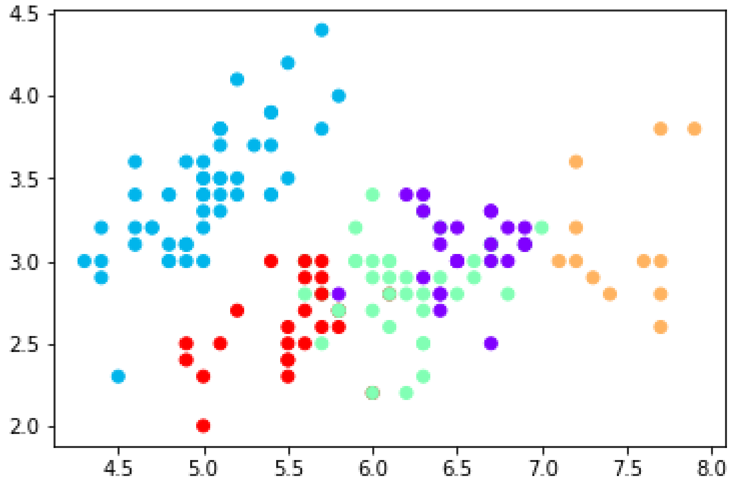


In [8]:

```
plt.scatter(x[:,0],x[:,1],c=y_kmeans5, cmap= 'rainbow')
```

Out[8]:

```
<matplotlib.collections.PathCollection at 0x27714b63be0>
```

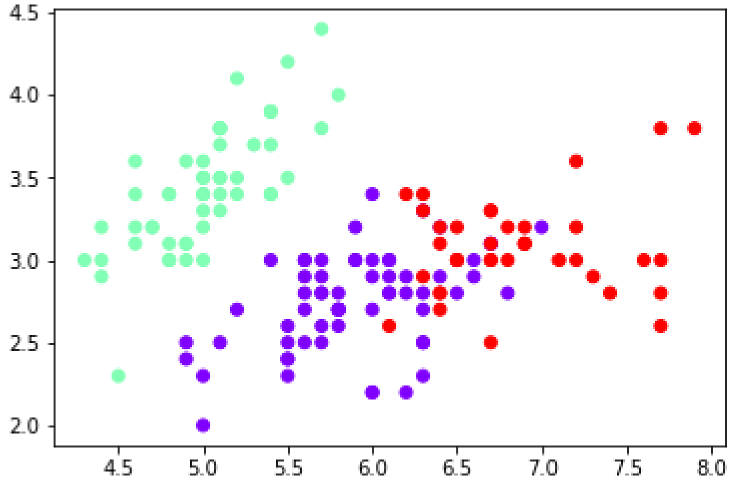


In [9]:

```
kmeans3= KMeans(n_clusters=3)
y_kmeans3=kmeans3.fit_predict(x)
plt.scatter(x[:,0],x[:,1],c=y_kmeans3, cmap='rainbow')
```

Out[9]:

```
<matplotlib.collections.PathCollection at 0x27714bdad00>
```



Lab 8: Association Rules

In [10]:

```
datasets=[['drink','nuts','diaper'],
['drink','coffee','diaper'],
['drink','diaper','eggs'],
['nuts','eggs','milk'],
['nuts','coffee','diaper','eggs','milk']]
```

In [11]:

```
pip install mlxtend
```

```
Collecting mlxtend
  Downloading mlxtend-0.19.0-py2.py3-none-any.whl (1.3 MB)
Requirement already satisfied: joblib>=0.13.2 in c:\anaconda3\lib\site-packa
ges (from mlxtend) (1.1.0)
Requirement already satisfied: scipy>=1.2.1 in c:\anaconda3\lib\site-package
s (from mlxtend) (1.7.1)
Requirement already satisfied: setuptools in c:\anaconda3\lib\site-packages
(from mlxtend) (58.0.4)
Requirement already satisfied: scikit-learn>=0.20.3 in c:\anaconda3\lib\site
-packages (from mlxtend) (0.24.2)
Requirement already satisfied: numpy>=1.16.2 in c:\anaconda3\lib\site-packag
es (from mlxtend) (1.20.3)
Requirement already satisfied: matplotlib>=3.0.0 in c:\anaconda3\lib\site-pa
ckages (from mlxtend) (3.4.3)
Requirement already satisfied: pandas>=0.24.2 in c:\anaconda3\lib\site-packa
ges (from mlxtend) (1.3.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\anaconda3\lib\site
-packages (from matplotlib>=3.0.0->mlxtend) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\anaconda3\lib\site-package
s (from matplotlib>=3.0.0->mlxtend) (0.10.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\anaconda3\lib\site-pac
kages (from matplotlib>=3.0.0->mlxtend) (3.0.4)
Requirement already satisfied: pillow>=6.2.0 in c:\anaconda3\lib\site-packag
es (from matplotlib>=3.0.0->mlxtend) (8.4.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\anaconda3\lib\site-pa
ckages (from matplotlib>=3.0.0->mlxtend) (1.3.1)
Requirement already satisfied: six in c:\anaconda3\lib\site-packages (from c
ycler>=0.10->matplotlib>=3.0.0->mlxtend) (1.16.0)
Requirement already satisfied: pytz>=2017.3 in c:\anaconda3\lib\site-package
s (from pandas>=0.24.2->mlxtend) (2021.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\anaconda3\lib\site
-packages (from scikit-learn>=0.20.3->mlxtend) (2.2.0)
Installing collected packages: mlxtend
Successfully installed mlxtend-0.19.0
Note: you may need to restart the kernel to use updated packages.
```

In [12]:

```
from mlxtend.frequent_patterns import apriori
```

In [13]:

```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
TranEncod = TransactionEncoder()
te_ary = TranEncod.fit(datasets).transform(datasets)
df = pd.DataFrame(te_ary, columns=TranEncod.columns_)
df
```

Out[13]:

	coffee	diaper	drink	eggs	milk	nuts
0	False	True	True	False	False	True
1	True	True	True	False	False	False
2	False	True	True	True	False	False
3	False	False	False	True	True	True
4	True	True	False	True	True	True

In [14]:

```
apriori(df, min_support=0.6)
```

Out[14]:

	support	itemsets
0	0.8	(1)
1	0.6	(2)
2	0.6	(3)
3	0.6	(5)
4	0.6	(1, 2)

In [15]:

```
apriori(df, min_support=0.6, use_colnames=True)
```

Out[15]:

	support	itemsets
0	0.8	(diaper)
1	0.6	(drink)
2	0.6	(eggs)
3	0.6	(nuts)
4	0.6	(drink, diaper)

In [16]:

```
frequent_itemsets = apriori(df, min_support=0.5, use_colnames=True)
frequent_itemsets['length'] = frequent_itemsets['itemsets'].apply(lambda x: len(x))
frequent_itemsets
```

Out[16]:

	support	itemsets	length
0	0.8	(diaper)	1
1	0.6	(drink)	1
2	0.6	(eggs)	1
3	0.6	(nuts)	1
4	0.6	(drink, diaper)	2

In [17]:

```
frequent_itemsets[ (frequent_itemsets['length'] == 2) &
                   (frequent_itemsets['support'] >= 0.5) ]
```

Out[17]:

	support	itemsets	length
4	0.6	(drink, diaper)	2

In [18]:

```
frequent_itemsets[frequent_itemsets['itemsets']=={'diaper','drink'}]
```

Out[18]:

	support	itemsets	length
4	0.6	(drink, diaper)	2

In [ ]:

```

```