Assignment 4

Name: Swapnil Ghosh
Roll: 001911001067
Dept: IT

Partition based: K-means

IRIS PLANT DATASET

In [1]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
```

In [ ]:
```
iris=load_iris()   #loading iris dataset from sklearn.datasets
iris
```

In [3]:
```
df=pd.DataFrame(data=iris.data, columns=['sepal length','sepal width','petal length','petal width'])
df
```

Out[3]:

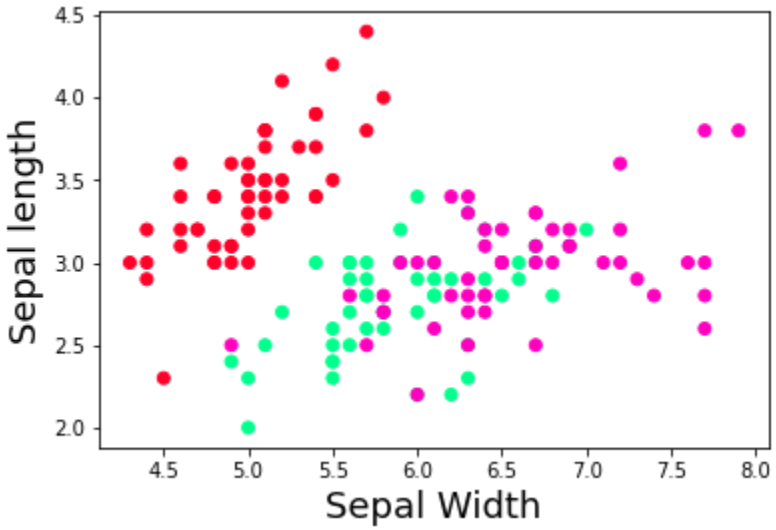
	sepal length	sepal width	petal length	petal width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [4]:
```
x=iris.data
```

In [5]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=iris.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('Sepal Width', fontsize=18)
plt.ylabel('Sepal length', fontsize=18)
```

Out[5]:
```
Text(0, 0.5, 'Sepal length')
```

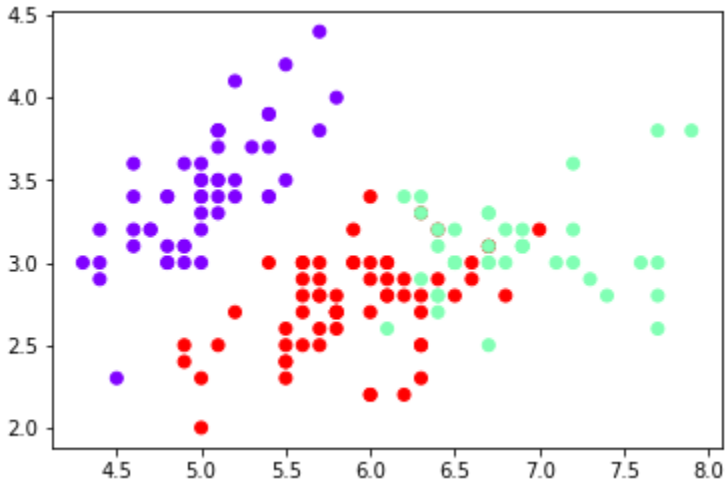


In [6]:
```
kmeans = KMeans(init="random", n_clusters=3, n_init=10, max_iter=300, random_state=42)
y = kmeans.fit_predict(x)
```

In [7]:
```
print("K-Means Cluster Centers")
print(kmeans.cluster_centers_)
print("Cluster Labels")
print(kmeans.labels_)
```

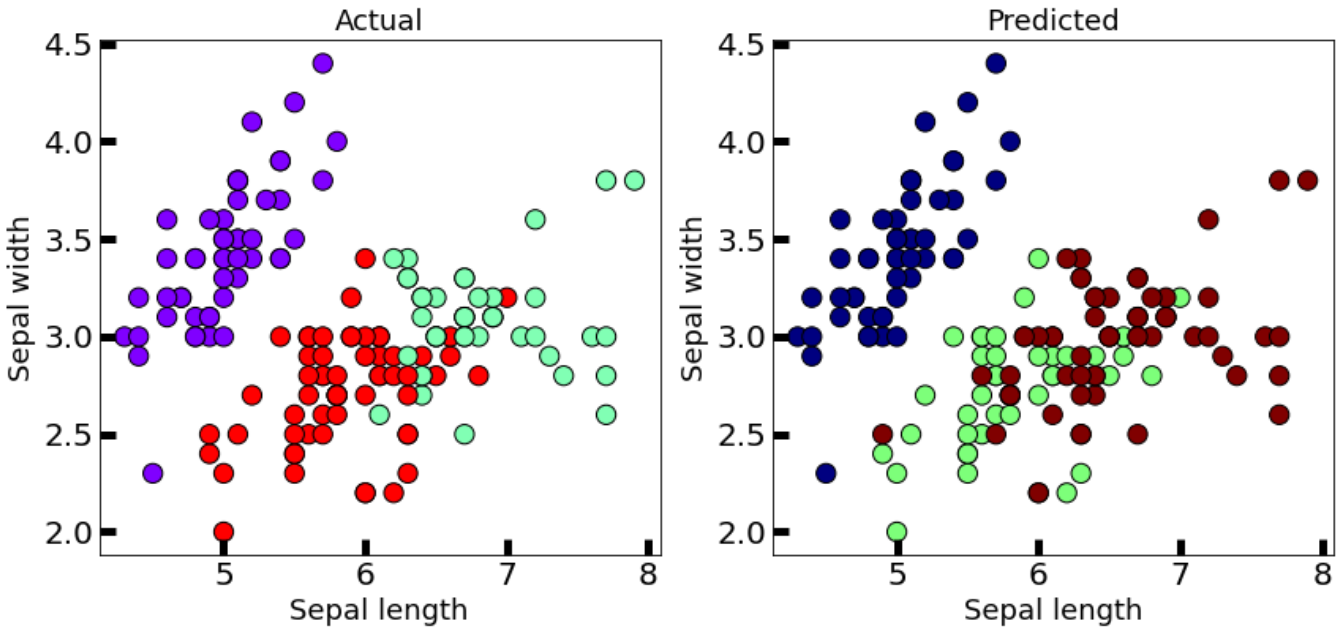
```
K-Means Cluster Centers
[[5.006      3.428      1.462      0.246     ]
 [6.85       3.07368421 5.74210526 2.07105263]
 [5.9016129  2.7483871  4.39354839 1.43387097]]
Cluster Labels
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 1 2 2 2 2 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 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 2 1 1 1 2 1 1 1 2 1 1 1 2 1
 1 2]
```

In [8]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=kmeans.labels_, cmap=
'rainbow') #try using cmap='rainbow'
plt.show()
```



In [9]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['sepal length'], y=df['sepal width'], c=y, cmap='rainbow',
edgecolor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['sepal length'], y=df['sepal width'], c=iris.target, cmap=
'jet',edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[9]:
```
Text(0.5, 1.0, 'Predicted')
```



In [10]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, kmeans.labels_)
```

```
The silhouette score is :
```

Out[10]:
```
0.5528190123564091
```

In [11]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, kmeans.labels_)
```

```
The calinski harabasz score is :
```

Out[11]:
```
561.62775662962
```

In [12]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, kmeans.labels_)
```

```
The davies bouldin score is :
```

Out[12]:
```
0.6619715465007511
```

In [13]:
```
print("It is observed that TSS=SSE+SSB is a constant. Hence we will calculate t
he TSS ans substract SSE from it to get SSB")

print("The value of SSE is: ")
print(kmeans.inertia_)



# Finding the overall centroid of the data points
centers = kmeans.cluster_centers_
center_x = []
for center in centers:
  center_x.append(center[0])
center_x
overall_center = sum(center_x)/len(center_x)

tss = 0
for i in range(len(df)):
  a = df.iloc[i][0] - overall_center
  b = pow(a,2)
  tss = tss+b



print("The value of SSB is: ")
print(tss - kmeans.inertia_)
```

```
It is observed that TSS=SSE+SSB is a constant. Hence we will calculate the TSS a
ns substract SSE from it to get SSB
The value of SSE is:
78.85144142614601
The value of SSB is:
24.18035246910202
```

WINE DATASET

In [14]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_wine
```

In [ ]:
```
wine=load_wine()   #loading iris dataset from sklearn.datasets
wine
```

In [16]:
```
x=wine.data
```

In [17]:
```
df=pd.DataFrame(data=wine.data, columns=['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline'])
df
```

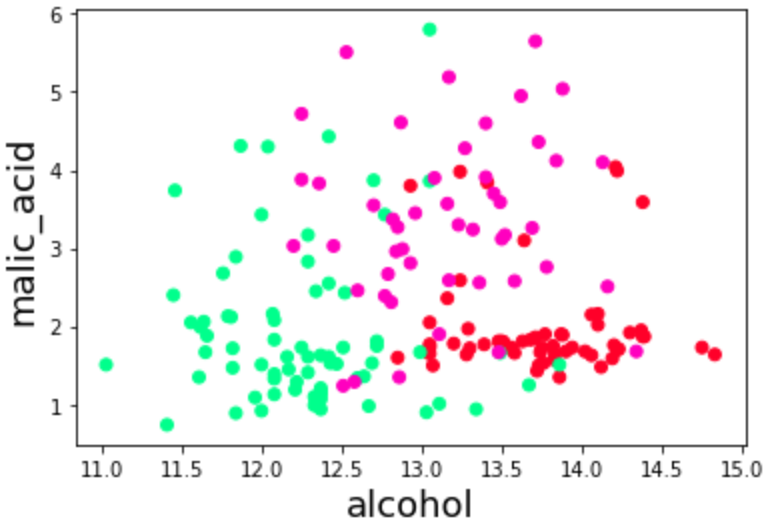
Out[17]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39
...	...	...	...	...	...	...	...	..
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56

178 rows × 13 columns

In [18]:
```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=wine.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('alcohol', fontsize=18)
plt.ylabel('malic_acid', fontsize=18)
```

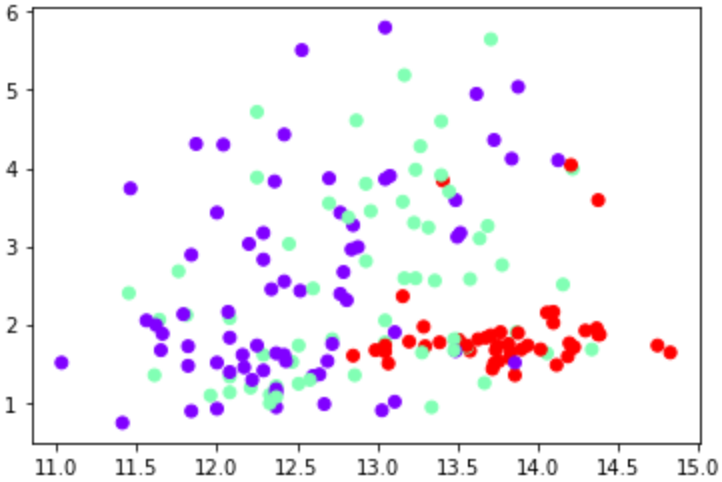
Out[18]:
```
Text(0, 0.5, 'malic_acid')
```



In [19]:
```
kmeans = KMeans(init="random", n_clusters=3, n_init=10, max_iter=300, random_state=42)
y = kmeans.fit_predict(x)
```

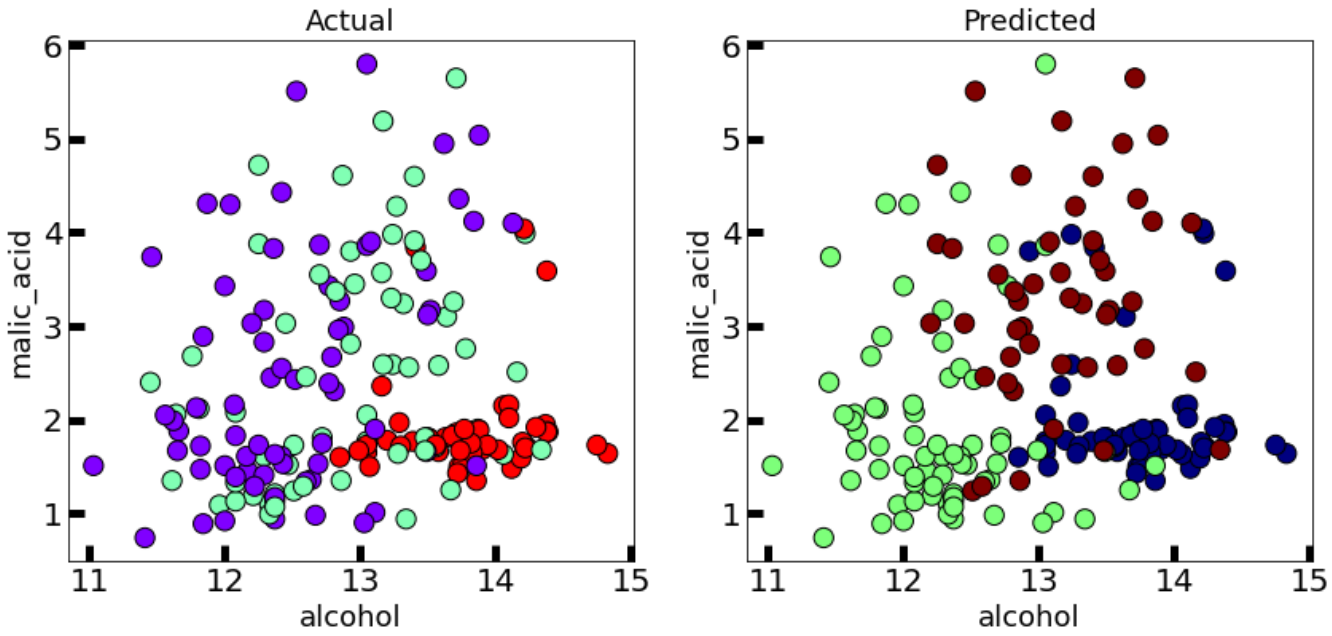
In [ ]:
```
print("K-Means Cluster Centers")
print(kmeans.cluster_centers_)
print("Cluster Labels")
print(kmeans.labels_)
```

In [ ]:
```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=kmeans.labels_, cmap='rainbow') #try using cmap='rainbow'
plt.show()
```



In [21]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['alcohol'], y=df['malic_acid'], c=y, cmap='rainbow',edgeco
lor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['alcohol'], y=df['malic_acid'], c=wine.target, cmap='jet',
edgecolor='k', s=150)
axes[0].set_xlabel('alcohol', fontsize=18)
axes[0].set_ylabel('malic_acid', fontsize=18)
axes[1].set_xlabel('alcohol', fontsize=18)
axes[1].set_ylabel('malic_acid', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[21]: Text(0.5, 1.0, 'Predicted')



In [22]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, kmeans.labels_)
```

```
The silhouette score is :
```

Out[22]: 0.5711381937868844

In [23]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, kmeans.labels_)
```

```
The calinski harabasz score is :
```

Out[23]: 561.815657860671

In [24]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, kmeans.labels_)
```

```
The davies bouldin score is :
```

Out[24]:
```
0.5342431775436273
```

In [25]:
```
print("It is observed that TSS=SSE+SSB is a constant. Hence we will calculate t
he TSS ans substract SSE from it to get SSB")

print("The value of SSE is: ")
print(kmeans.inertia_)



# Finding the overall centroid of the data points
centers = kmeans.cluster_centers_
center_x = []
for center in centers:
  center_x.append(center[0])
center_x
overall_center = sum(center_x)/len(center_x)

tss = 0
for i in range(len(df)):
  a = df.iloc[i][0] - overall_center
  b = pow(a,2)
  tss = tss+b



print("The value of SSB is: ")
print(tss - kmeans.inertia_)
```

```
It is observed that TSS=SSE+SSB is a constant. Hence we will calculate the TSS a
ns substract SSE from it to get SSB
The value of SSE is:
2370689.686782968
The value of SSB is:
-2370571.8053313834
```

Partition based: K-medoids

IRIS PLANT DATASET

In [ ]:
```
#importing libraries
!pip install scikit-learn-extra
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn_extra.cluster import KMedoids
from sklearn.datasets import load_iris
```

In [ ]:
```
iris=load_iris()   #loading iris dataset from sklearn.datasets
iris
```

In [28]:
```
df=pd.DataFrame(data=iris.data, columns=['sepal length','sepal width','petal length','petal width'])
df
```

Out[28]:

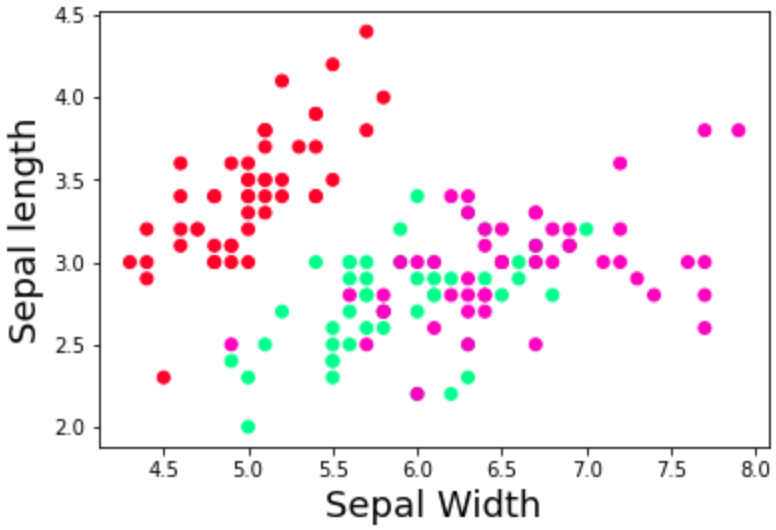
	sepal length	sepal width	petal length	petal width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [29]:
```
x=iris.data
```

In [30]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=iris.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('Sepal Width', fontsize=18)
plt.ylabel('Sepal length', fontsize=18)
```

Out[30]:
```
Text(0, 0.5, 'Sepal length')
```

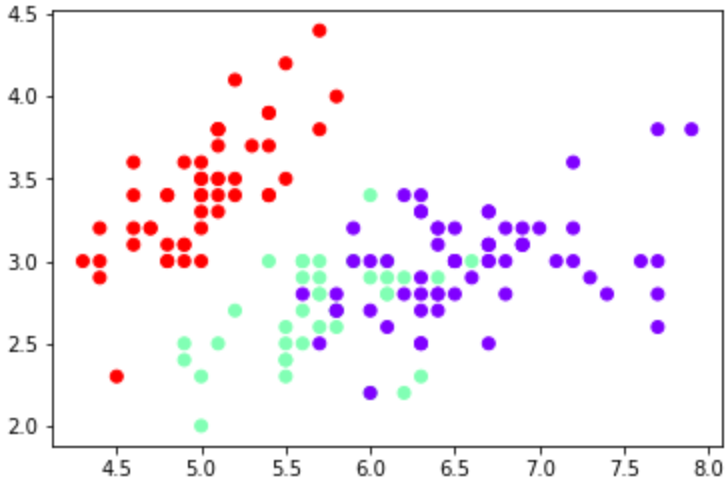


In [31]:
```
kmedoid = KMedoids(init="heuristic", n_clusters=3, max_iter=300, random_state=42)
y = kmedoid.fit_predict(x)
```

In [32]:
```
print("K-Medoids Cluster Centers")
print(kmedoid.cluster_centers_)
print("Cluster Labels")
print(kmedoid.labels_)
```

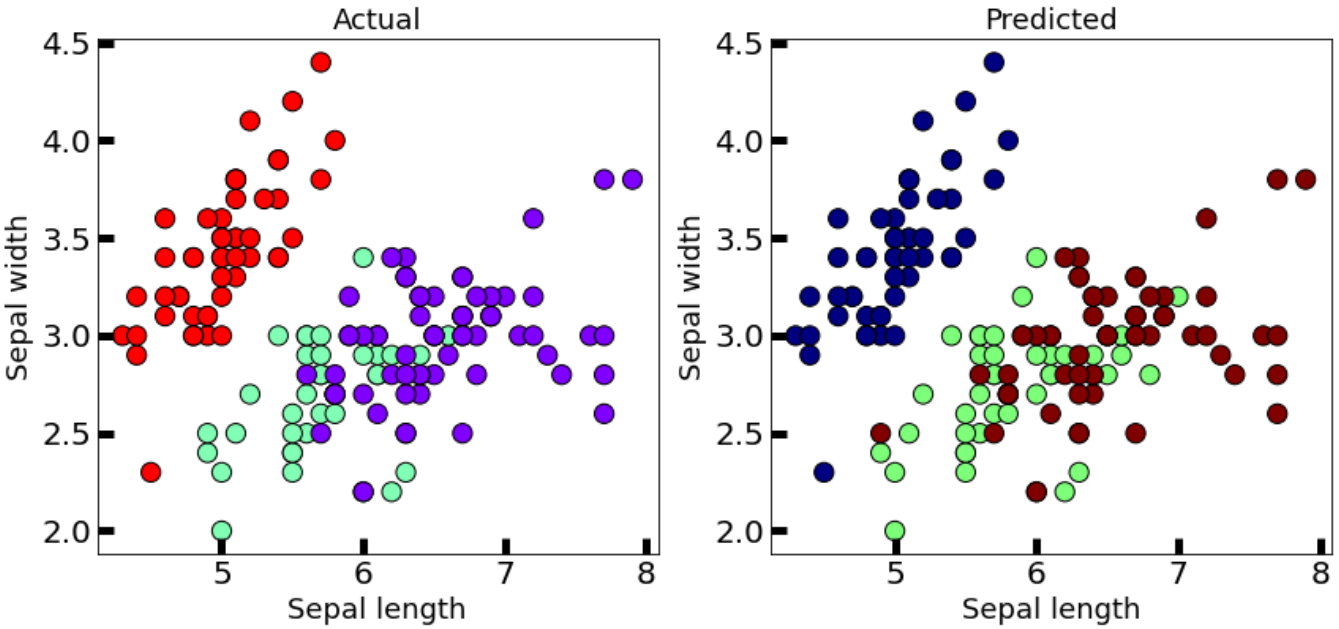
```
K-Medoids Cluster Centers
[[6.5 3.  5.2 2. ]
 [5.7 2.8 4.1 1.3]
 [5.  3.4 1.5 0.2]]
Cluster Labels
[2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 1 0 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 0 1
 1 1 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 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 0 0 0 0 0
 0 0]
```

In [33]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=kmedoid.labels_, cmap=
'rainbow') #try using cmap='rainbow'
plt.show()
```



In [34]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['sepal length'], y=df['sepal width'], c=y, cmap='rainbow',
edgecolor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['sepal length'], y=df['sepal width'], c=iris.target, cmap=
'jet',edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[34]:
```
Text(0.5, 1.0, 'Predicted')
```



In [35]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, kmedoid.labels_)
```

```
The silhouette score is :
```

Out[35]:
```
0.5201984013106979
```

In [36]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, kmedoid.labels_)
```

```
The calinski harabasz score is :
```

Out[36]:
```
521.5609065033622
```

In [37]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, kmedoid.labels_)
```

```
The davies bouldin score is :
```

Out[37]:
```
0.6686244410428932
```

In [38]:
```
print("It is observed that TSS=SSE+SSB is a constant. Hence we will calculate t
he TSS ans substract SSE from it to get SSB")

print("The value of SSE is: ")
print(kmedoid.inertia_)



# Finding the overall centroid of the data points
centers = kmedoid.cluster_centers_
center_x = []
for center in centers:
  center_x.append(center[0])
center_x
overall_center = sum(center_x)/len(center_x)

tss = 0
for i in range(len(df)):
  a = df.iloc[i][0] - overall_center
  b = pow(a,2)
  tss = tss+b



print("The value of SSB is: ")
print(tss - kmedoid.inertia_)
```

```
It is observed that TSS=SSE+SSB is a constant. Hence we will calculate the TSS a
ns substract SSE from it to get SSB
The value of SSE is:
98.86857318335639
The value of SSB is:
5.114760149976831
```

WINE DATASET

In [39]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn_extra.cluster import KMedoids
from sklearn.datasets import load_iris
```

In [ ]:
```
wine=load_wine()   #loading iris dataset from sklearn.datasets
wine
```

In [41]:
```
x=wine.data
```

In [42]:
```
df=pd.DataFrame(data=wine.data, columns=['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline'])
df
```

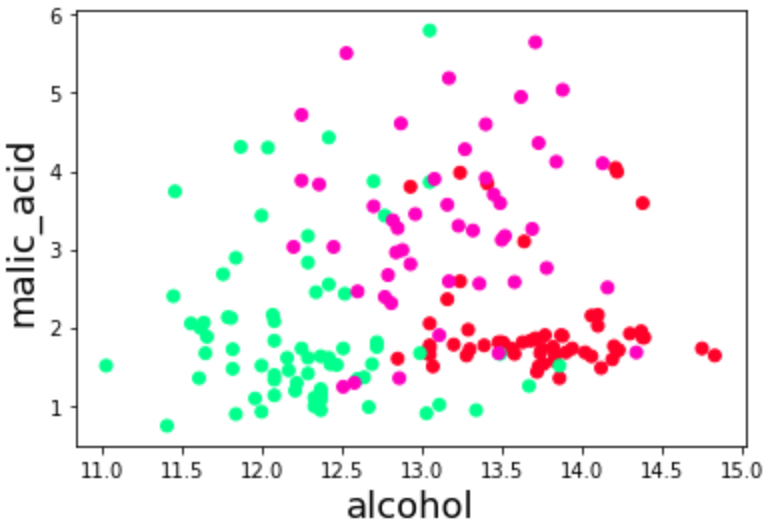
Out[42]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39
...	...	...	...	...	...	...	...	..
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56

178 rows × 13 columns

In [43]:
```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=wine.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('alcohol', fontsize=18)
plt.ylabel('malic_acid', fontsize=18)
```

Out[43]:
```
Text(0, 0.5, 'malic_acid')
```

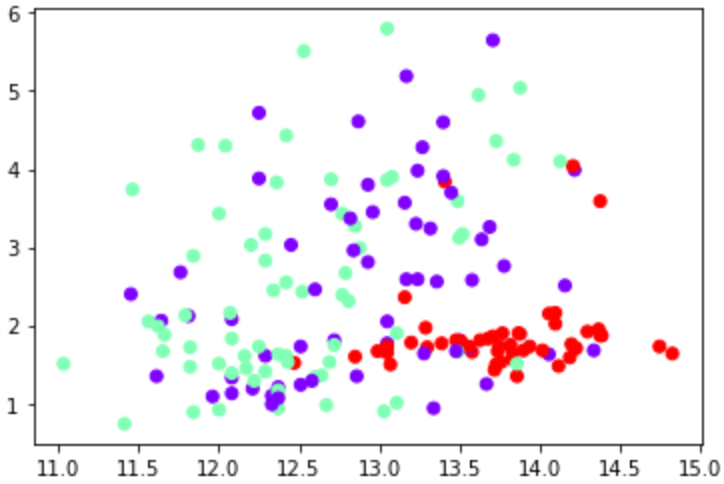


In [44]:
```
kmedoid = KMedoids(init="heuristic", n_clusters=3, max_iter=300, random_state=42)
y = kmedoid.fit_predict(x)
```

In [45]:
```
print("K-Medoids Cluster Centers")
print(kmedoid.cluster_centers_)
print("Cluster Labels")
print(kmedoid.labels_)
```

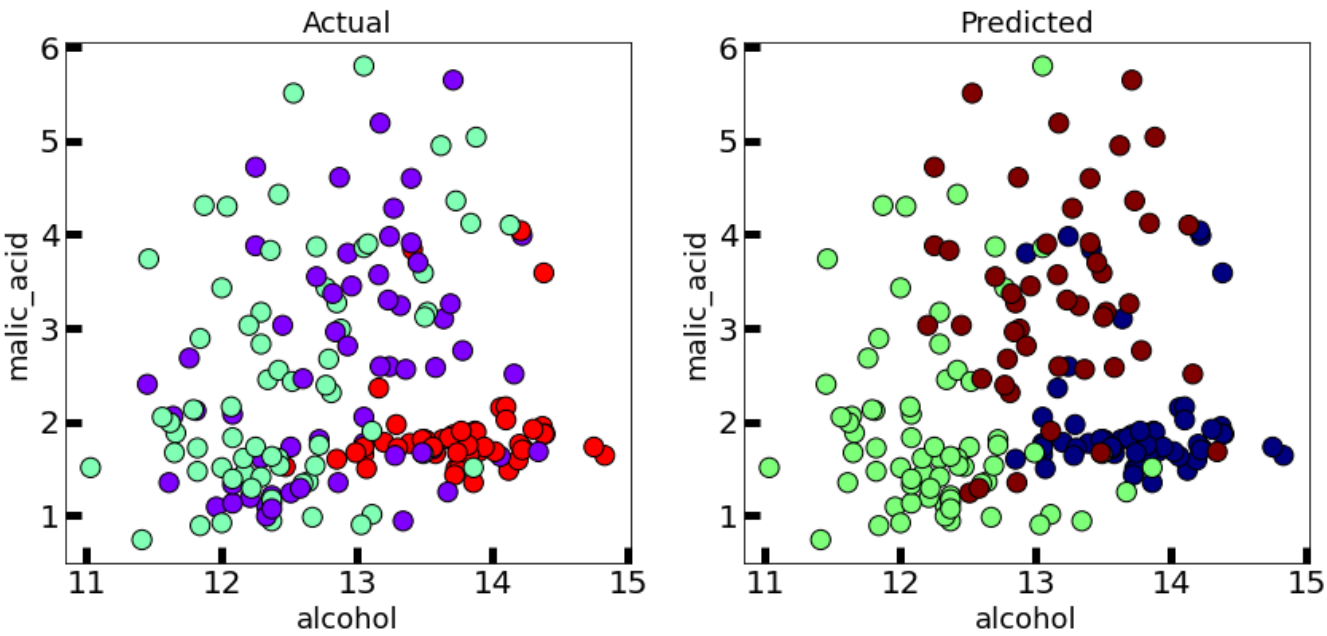
```
K-Medoids Cluster Centers
[[1.260e+01 2.460e+00 2.200e+00 1.850e+01 9.400e+01 1.620e+00 6.600e-01
  6.300e-01 9.400e-01 7.100e+00 7.300e-01 1.580e+00 6.950e+02]
 [1.349e+01 1.660e+00 2.240e+00 2.400e+01 8.700e+01 1.880e+00 1.840e+00
  2.700e-01 1.030e+00 3.740e+00 9.800e-01 2.780e+00 4.720e+02]
 [1.383e+01 1.570e+00 2.620e+00 2.000e+01 1.150e+02 2.950e+00 3.400e+00
  4.000e-01 1.720e+00 6.600e+00 1.130e+00 2.570e+00 1.130e+03]]
Cluster Labels
[2 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 2 2 0 0 2 2 2 2 2 2 2 2 2 0
 2 2 0 0 2 2 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 1 0 1 1 0 1 1 0 0 0 1 1 2
 0 1 1 1 0 1 1 0 0 1 1 1 1 1 0 0 1 1 1 1 1 2 0 1 0 1 0 1 1 1 0 1 1 1 1 0 1
 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 0 1 1 0 0 0 1 1 0 0 1 0
 0 1 1 1 1 0 0 0 1 0 0 0 1 0 1 0 0 1 0 0 0 0 1 1 0 0 0 0 0 1]
```

In [46]:
```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=kmedoid.labels_, cmap='rainbow') #try using cmap='rainbow'
plt.show()
```



In [47]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['alcohol'], y=df['malic_acid'], c=y, cmap='rainbow',edgecolor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['alcohol'], y=df['malic_acid'], c=wine.target, cmap='jet', edgecolor='k', s=150)
axes[0].set_xlabel('alcohol', fontsize=18)
axes[0].set_ylabel('malic_acid', fontsize=18)
axes[1].set_xlabel('alcohol', fontsize=18)
axes[1].set_ylabel('malic_acid', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=20)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=20)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[47]:
```
Text(0.5, 1.0, 'Predicted')
```



In [48]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, kmedoid.labels_)
```

```
The silhouette score is :
```

Out[48]:
```
0.5666480408636575
```

In [49]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, kmedoid.labels_)
```

```
The calinski harabasz score is :
```

Out[49]:
```
539.3792353535451
```

In [50]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, kmedoid.labels_)
```

```
The davies bouldin score is :
```

Out[50]:
```
0.529239412600317
```

In [51]:
```
print("It is observed that TSS=SSE+SSB is a constant. Hence we will calculate t
he TSS ans substract SSE from it to get SSB")

print("The value of SSE is: ")
print(kmedoid.inertia_)



# Finding the overall centroid of the data points
centers = kmedoid.cluster_centers_
center_x = []
for center in centers:
  center_x.append(center[0])
center_x
overall_center = sum(center_x)/len(center_x)

tss = 0
for i in range(len(df)):
  a = df.iloc[i][0] - overall_center
  b = pow(a,2)
  tss = tss+b



print("The value of SSB is: ")
print(tss - kmedoid.inertia_)
```

```
It is observed that TSS=SSE+SSB is a constant. Hence we will calculate the TSS a
ns substract SSE from it to get SSB
The value of SSE is:
16376.969320536637
The value of SSB is:
-16243.642776092192
```

Hierarchical: Dendrogram

IRIS PLANT DATASET

In [52]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
```

In [ ]:
```
iris=load_iris()   #loading iris dataset from sklearn.datasets
iris
```

In [54]:
```
df=pd.DataFrame(data=iris.data, columns=['sepal length','sepal width','petal length','petal width'])
df
```

Out[54]:

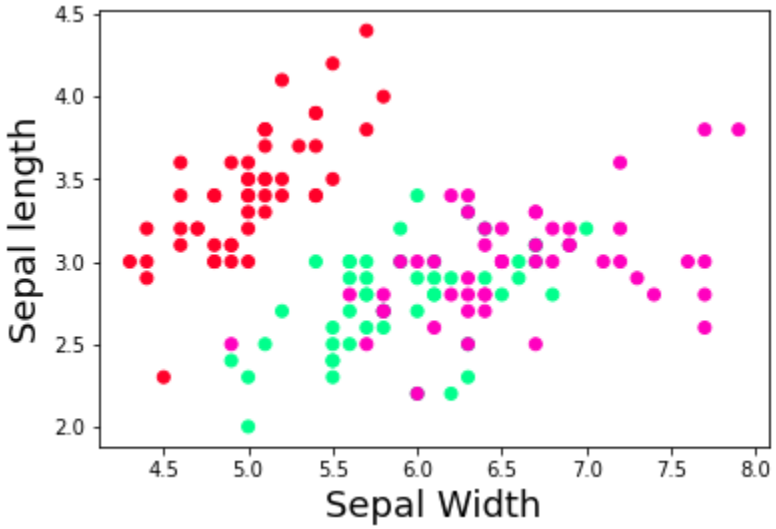
	sepal length	sepal width	petal length	petal width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

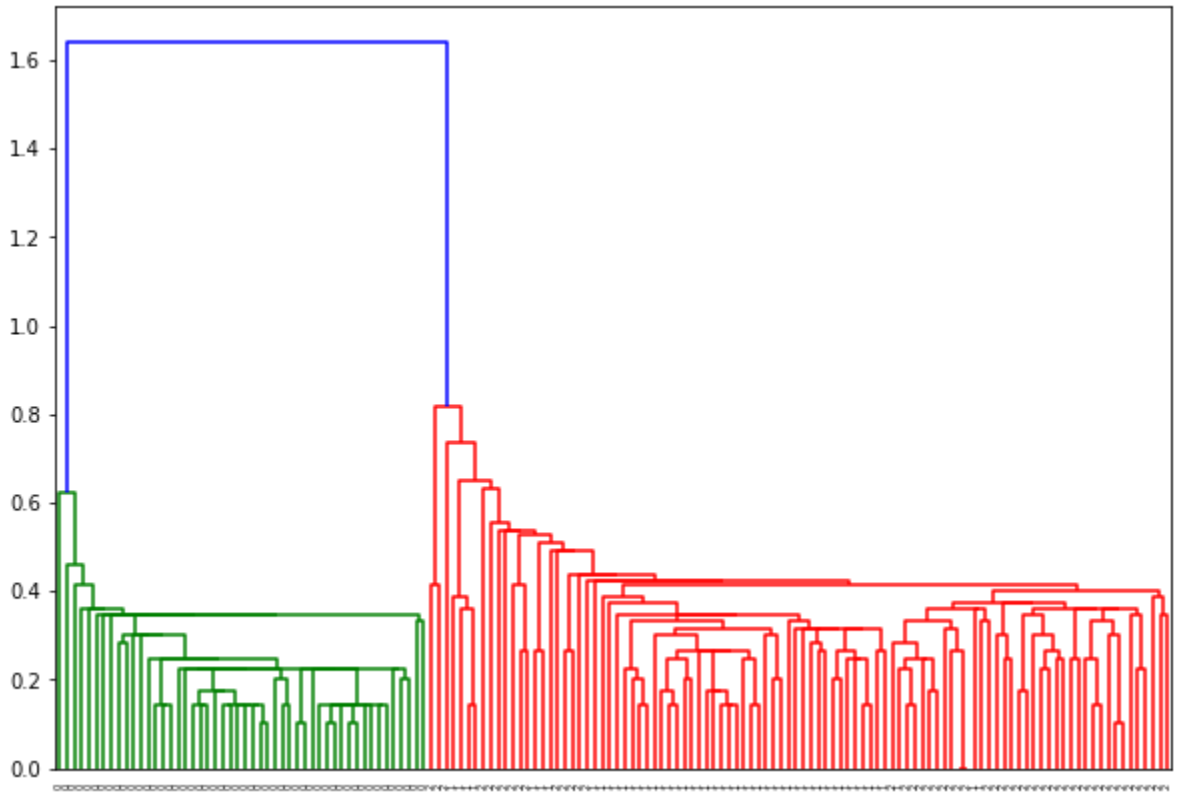
In [55]:
```
x=iris.data
```

In [56]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=iris.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('Sepal Width', fontsize=18)
plt.ylabel('Sepal length', fontsize=18)
```

Out[56]:
```
Text(0, 0.5, 'Sepal length')
```



In [57]:
```
from scipy.cluster.hierarchy import dendrogram, linkage

linked = linkage(x, 'single')
plt.figure(figsize=(10,7))

dendrogram(linked,
           orientation='top',
           labels=iris.target,
           distance_sort='descending',
           show_leaf_counts=True)

plt.show()
```



WINE DATASET

In [58]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_wine
```

In [ ]:
```
wine=load_wine()   #loading iris dataset from sklearn.datasets
wine
```

In [60]:
```
x=wine.data
```

In [61]:
```
df=pd.DataFrame(data=wine.data, columns=['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline'])
df
```

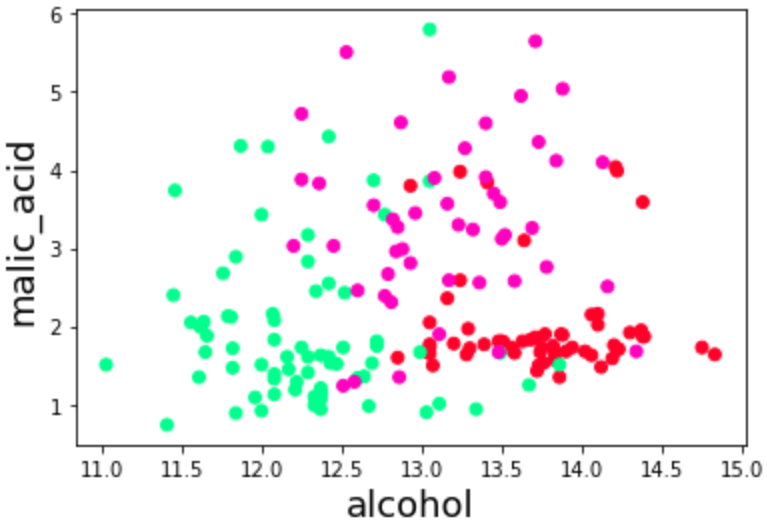
Out[61]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39
...	...	...	...	...	...	...	...	..
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56

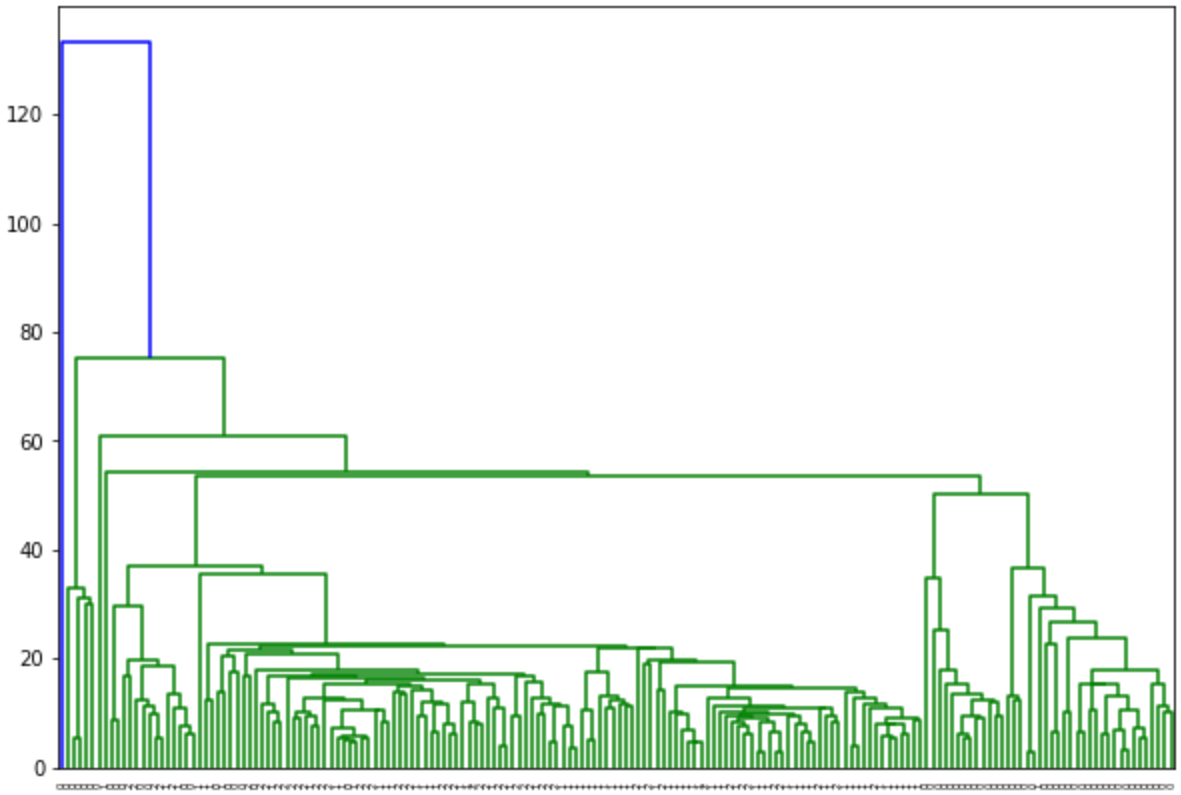
178 rows × 13 columns

In [62]:
```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=wine.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('alcohol', fontsize=18)
plt.ylabel('malic_acid', fontsize=18)
```

Out[62]:
```
Text(0, 0.5, 'malic_acid')
```



In [63]:
```
from scipy.cluster.hierarchy import dendrogram, linkage

linked = linkage(x, 'single')
plt.figure(figsize=(10,7))

dendrogram(linked,
           orientation='top',
           labels=wine.target,
           distance_sort='descending',
           show_leaf_counts=True)

plt.show()
```



Hierarchical: AGNES

IRIS PLANT DATASET

In [64]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
```

In [ ]:
```
iris=load_iris()   #loading iris dataset from sklearn.datasets
iris
```

In [66]:
```
df=pd.DataFrame(data=iris.data, columns=['sepal length','sepal width','petal length','petal width'])
df
```

Out[66]:

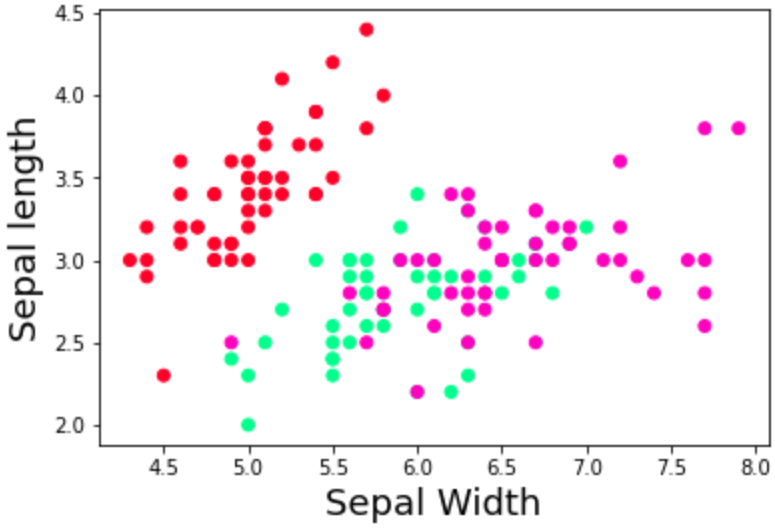
	sepal length	sepal width	petal length	petal width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [67]:
```
x=iris.data
```

In [68]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=iris.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('Sepal Width', fontsize=18)
plt.ylabel('Sepal length', fontsize=18)
```

Out[68]:
```
Text(0, 0.5, 'Sepal length')
```

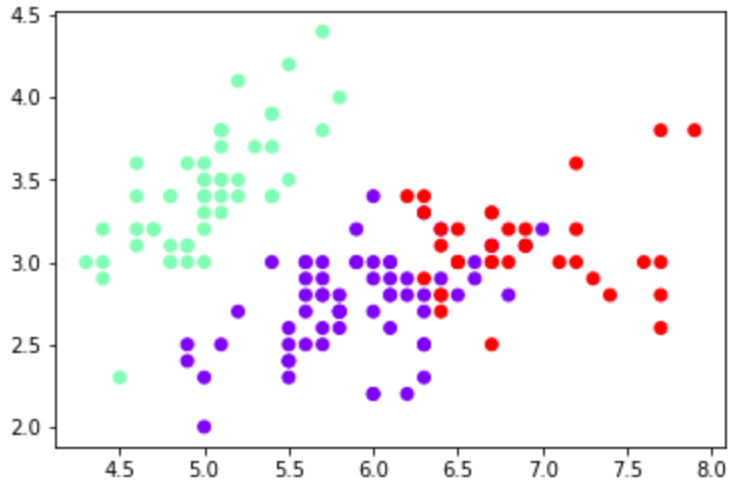


In [69]:
```
from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n_clusters=3, affinity='euclidean', linkage='ward')
y = cluster.fit_predict(x)
```

In [70]:
```
print("Cluster labels:")
print(cluster.labels_)
```

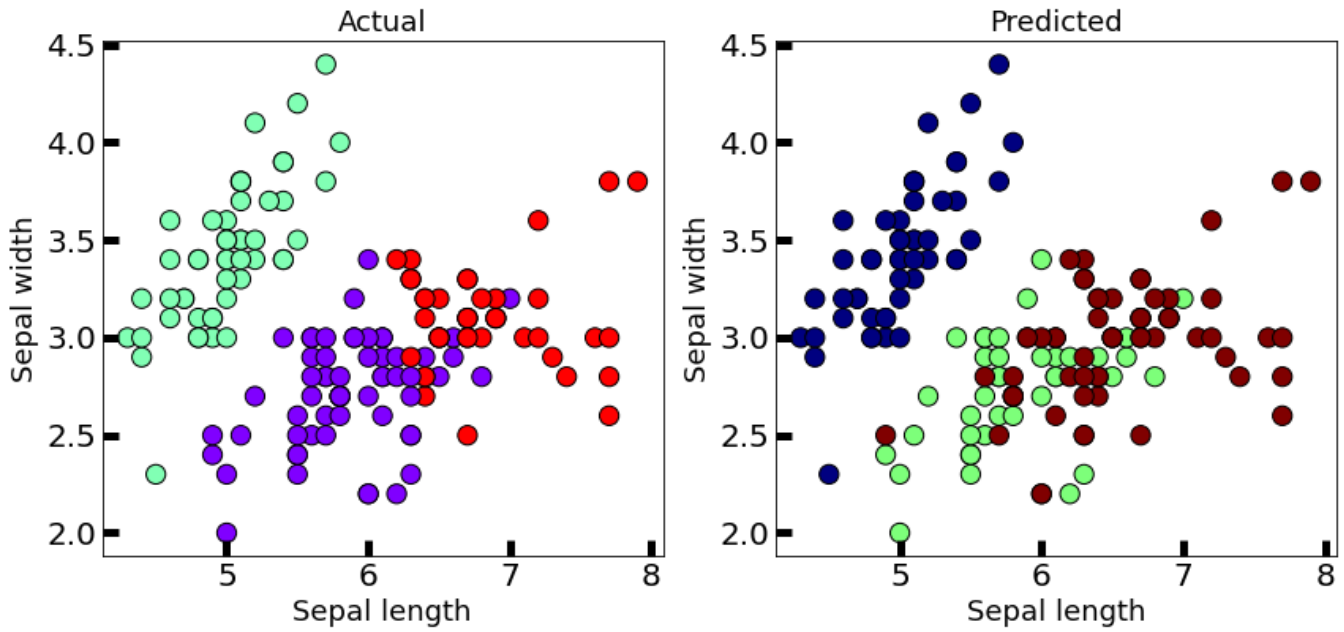
```
Cluster labels:
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 2 2 2 0 2 2 2 2
 2 2 0 0 2 2 2 2 0 2 0 2 0 2 2 0 0 2 2 2 2 2 0 0 2 2 2 0 2 2 2 0 2 2 2 0 2
 2 0]
```

In [71]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=cluster.labels_, cmap='rainbow') #try using cmap='rainbow'
plt.show()
```



In [72]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['sepal length'], y=df['sepal width'], c=y, cmap='rainbow',
edgecolor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['sepal length'], y=df['sepal width'], c=iris.target, cmap=
'jet',edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[72]:
```
Text(0.5, 1.0, 'Predicted')
```



In [73]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, cluster.labels_)
```

```
The silhouette score is :
```

Out[73]:
```
0.5543236611296415
```

In [74]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, cluster.labels_)
```

```
The calinski harabasz score is :
```

Out[74]:
```
558.0580408128307
```

In [75]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, cluster.labels_)
```

```
The davies bouldin score is :
```

Out[75]:
```
0.6562564540642065
```

WINE DATASET

In [76]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_wine
```

In [ ]:
```
wine=load_wine()   #loading iris dataset from sklearn.datasets
wine
```

In [78]:
```
x=wine.data
```

In [79]:
```
df=pd.DataFrame(data=wine.data, columns=['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline'])
df
```

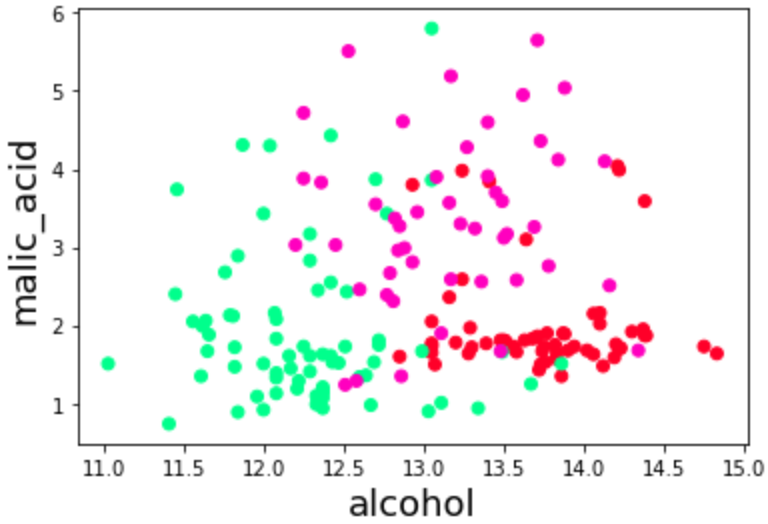
Out[79]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39
...	...	...	...	...	...	...	...	..
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56

178 rows × 13 columns

In [80]:
```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=wine.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('alcohol', fontsize=18)
plt.ylabel('malic_acid', fontsize=18)
```

Out[80]:
```
Text(0, 0.5, 'malic_acid')
```

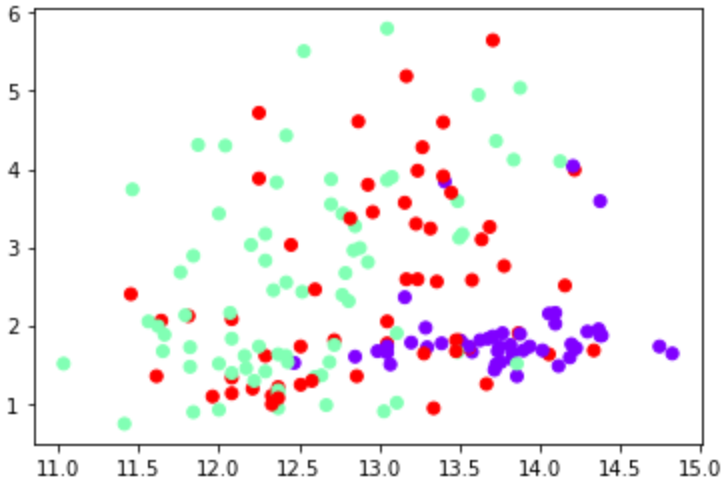


In [81]:
```
from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n_clusters=3, affinity='euclidean', linkage='ward')
y = cluster.fit_predict(x)
```

In [82]:
```
print("Cluster Labels")
print(cluster.labels_)
```

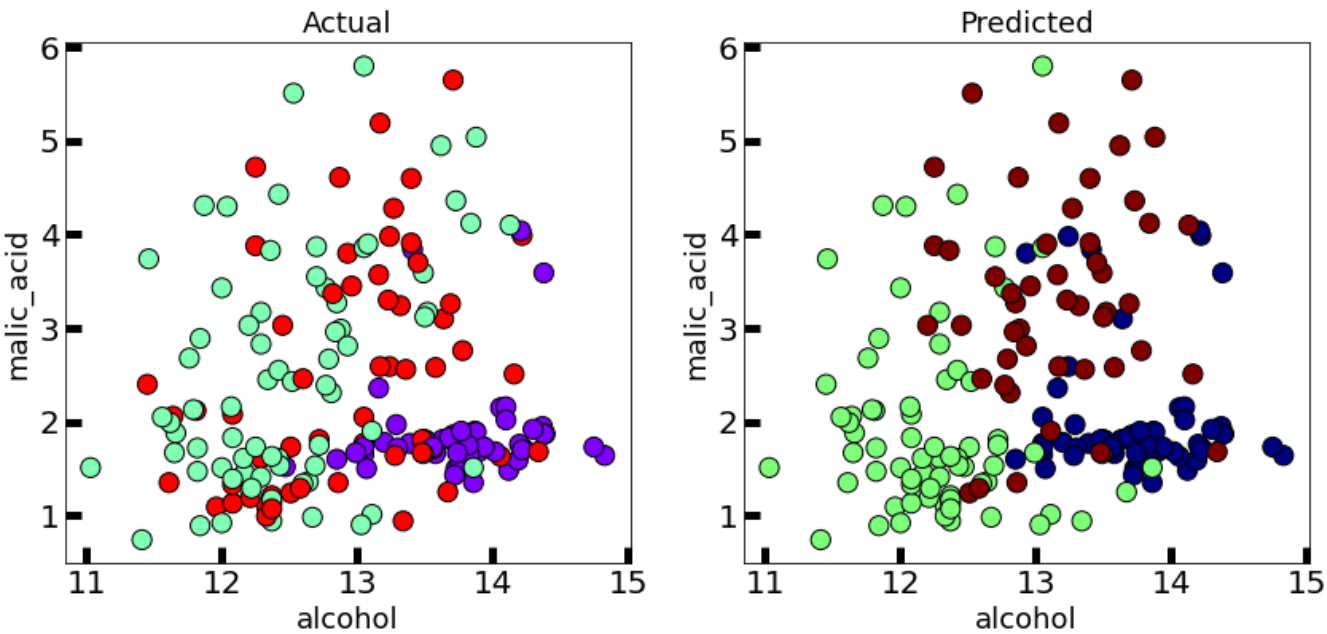
```
Cluster Labels
[0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 0 0 2 2 0 0 2 0 0 0 0 0 0 2 2
 0 0 2 2 0 0 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 1 2 1 1 2 1 1 2 2 2 1 1 0
 2 1 1 1 2 1 1 2 2 1 1 1 1 1 2 2 1 1 1 1 1 0 2 1 2 1 2 1 1 1 2 1 1 1 1 2 1
 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2 1 1 1 1 2 1 1 2 2 1 2
 2 1 1 1 1 2 2 2 1 2 2 2 1 2 1 2 2 1 2 2 2 2 1 1 2 2 2 2 2 1]
```

In [83]:
```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=cluster.labels_, cmap='rainbow') #try using cmap='rainbow'
plt.show()
```



In [84]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['alcohol'], y=df['malic_acid'], c=y, cmap='rainbow',edgecolor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['alcohol'], y=df['malic_acid'], c=wine.target, cmap='jet', edgecolor='k', s=150)
axes[0].set_xlabel('alcohol', fontsize=18)
axes[0].set_ylabel('malic_acid', fontsize=18)
axes[1].set_xlabel('alcohol', fontsize=18)
axes[1].set_ylabel('malic_acid', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=20)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=20)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[84]:
```
Text(0.5, 1.0, 'Predicted')
```



In [85]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, cluster.labels_)
```

```
The silhouette score is :
```

Out[85]:
```
0.5644796401732074
```

In [86]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, cluster.labels_)
```

```
The calinski harabasz score is :
```

Out[86]:
```
552.851711505718
```

In [87]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, cluster.labels_)
```

```
The davies bouldin score is :
```

Out[87]:
```
0.5357343073560216
```

Hierarchical: BIRCH

IRIS PLANT DATASET

In [88]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
```

In [ ]:
```
iris=load_iris()   #loading iris dataset from sklearn.datasets
iris
```

In [90]:
```
df=pd.DataFrame(data=iris.data, columns=['sepal length','sepal width','petal length','petal width'])
df
```

Out[90]:

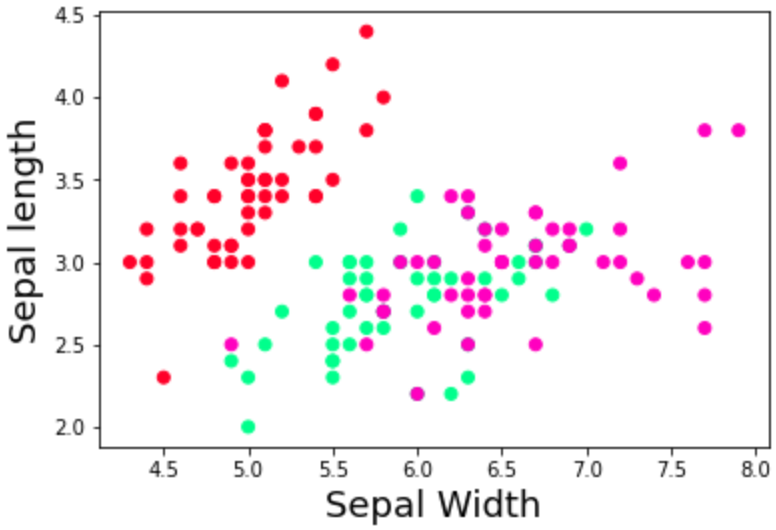
	sepal length	sepal width	petal length	petal width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [91]:
```
x=iris.data
```

In [92]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=iris.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('Sepal Width', fontsize=18)
plt.ylabel('Sepal length', fontsize=18)
```

Out[92]:
```
Text(0, 0.5, 'Sepal length')
```

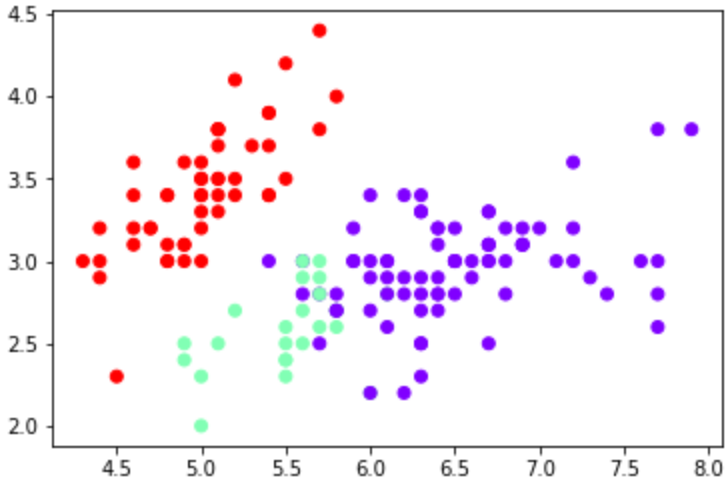


In [93]:
```
from sklearn.cluster import Birch

birch = Birch(n_clusters=3, compute_labels=True, branching_factor=50)
y = birch.fit_predict(x)
```

In [94]:
```
print("Cluster Labels")
print(cluster.labels_)
```

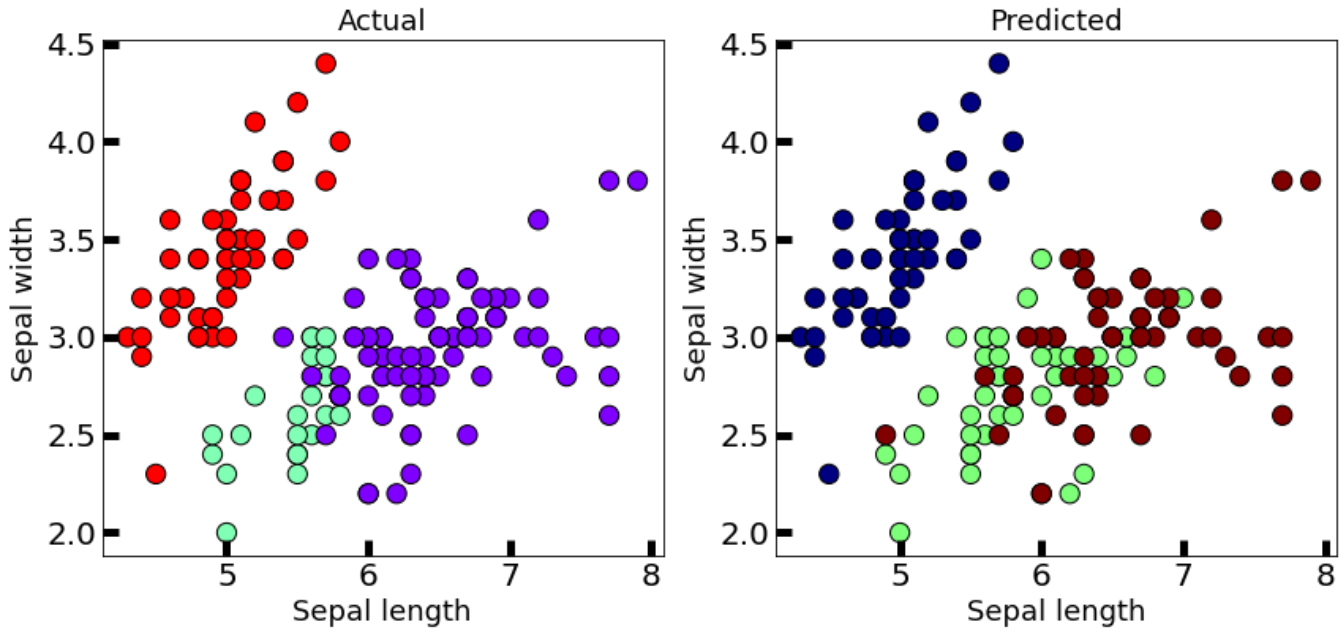
```
Cluster Labels
[0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 0 0 2 2 0 0 2 0 0 0 0 0 0 2 2
 0 0 2 2 0 0 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 1 2 1 1 2 1 1 2 2 2 1 1 0
 2 1 1 1 2 1 1 2 2 1 1 1 1 1 2 2 1 1 1 1 1 0 2 1 2 1 2 1 1 1 2 1 1 1 1 2 1
 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2 1 1 1 1 2 1 1 2 2 1 2
 2 1 1 1 1 2 2 2 1 2 2 2 1 2 1 2 2 1 2 2 2 2 1 1 2 2 2 2 2 1]
```

In [95]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=birch.labels_, cmap='r
ainbow') #try using cmap='rainbow'
plt.show()
```



In [96]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['sepal length'], y=df['sepal width'], c=y, cmap='rainbow',
edgecolor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['sepal length'], y=df['sepal width'], c=iris.target, cmap=
'jet',edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[96]:
```
Text(0.5, 1.0, 'Predicted')
```



In [97]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, birch.labels_)
```

```
The silhouette score is :
```

Out[97]:
```
0.5019524848046075
```

In [98]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, birch.labels_)
```

```
The calinski harabasz score is :
```

Out[98]:
```
458.47251055625765
```

In [99]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, birch.labels_)
```

```
The davies bouldin score is :
```

Out[99]:
```
0.6258305924331691
```

WINE DATASET

In [100]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_wine
```

In [ ]:
```
wine=load_wine()   #loading iris dataset from sklearn.datasets
wine
```

In [102]:
```
x=wine.data
```

In [103]:
```
df=pd.DataFrame(data=wine.data, columns=['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline'])
df
```

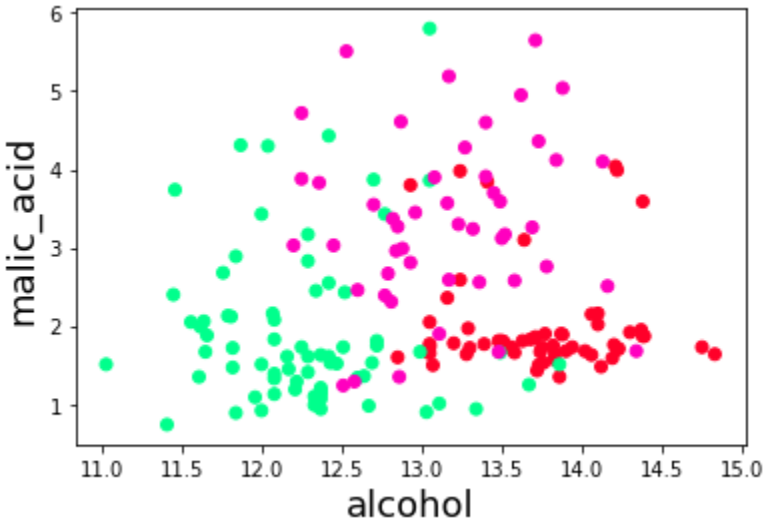
Out[103]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39
...	...	...	...	...	...	...	...	..
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56

178 rows × 13 columns

In [104]:
```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=wine.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('alcohol', fontsize=18)
plt.ylabel('malic_acid', fontsize=18)
```

Out[104]:
```
Text(0, 0.5, 'malic_acid')
```

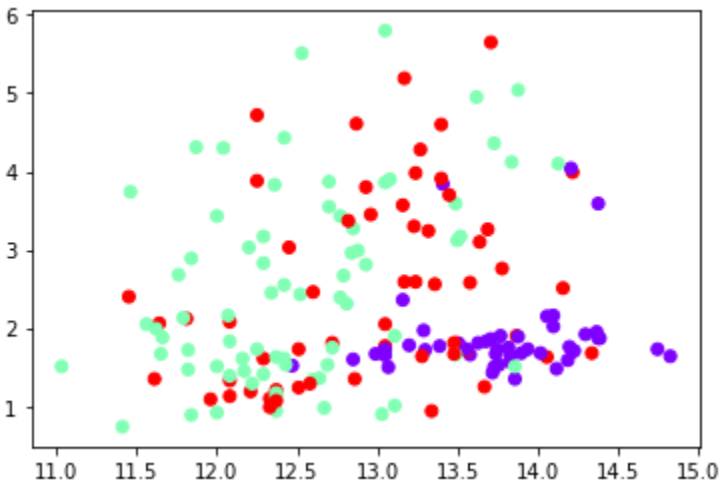


In [105]:
```
from sklearn.cluster import Birch

birch = Birch(n_clusters=3, compute_labels=True, branching_factor=50)
y = birch.fit_predict(x)
```

In [106]:
```
print("Cluster Labels")
print(cluster.labels_)
```

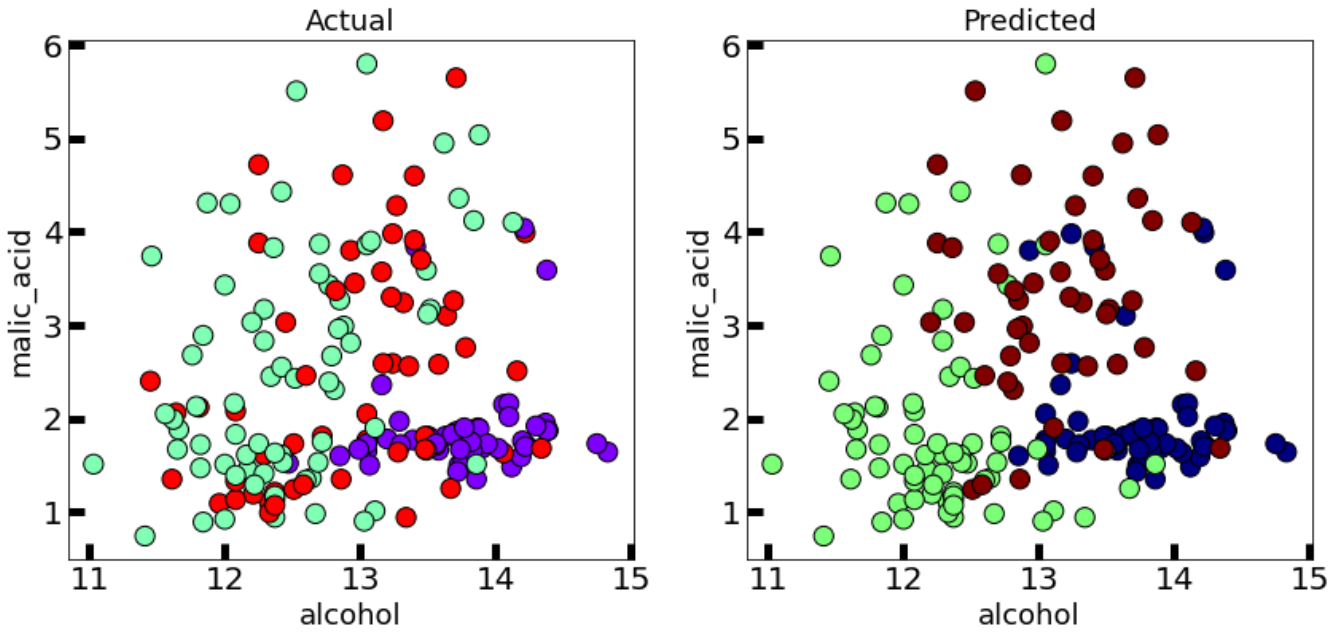
```
Cluster Labels
[0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 0 0 2 2 0 0 2 0 0 0 0 0 0 2 2
 0 0 2 2 0 0 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 1 2 1 1 2 1 1 2 2 2 1 1 0
 2 1 1 1 2 1 1 2 2 1 1 1 1 1 2 2 1 1 1 1 1 0 2 1 2 1 2 1 1 1 2 1 1 1 1 2 1
 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2 1 1 1 1 2 1 1 2 2 1 2
 2 1 1 1 1 2 2 2 1 2 2 2 1 2 1 2 2 1 2 2 2 2 1 1 2 2 2 2 2 1]
```

In [107]:
```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=birch.labels_, cmap='rainbow') #try using cmap='rainbow'
plt.show()
```



In [108]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['alcohol'], y=df['malic_acid'], c=y, cmap='rainbow',edgeco
lor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['alcohol'], y=df['malic_acid'], c=wine.target, cmap='jet',
edgecolor='k', s=150)
axes[0].set_xlabel('alcohol', fontsize=18)
axes[0].set_ylabel('malic_acid', fontsize=18)
axes[1].set_xlabel('alcohol', fontsize=18)
axes[1].set_ylabel('malic_acid', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[108]: Text(0.5, 1.0, 'Predicted')



In [109]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, birch.labels_)
```

```
The silhouette score is :
```

Out[109]: 0.5644796401732074

In [110]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, birch.labels_)
```

```
The calinski harabasz score is :
```

Out[110]: 552.851711505718

In [111]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, birch.labels_)
```

```
The davies bouldin score is :
```

Out[111]:
```
0.5357343073560216
```

Density based: DBSCAN

IRIS PLANT DATASET

In [112]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
```

In [ ]:
```
iris=load_iris()   #loading iris dataset from sklearn.datasets
iris
```

In [114]:
```
df=pd.DataFrame(data=iris.data, columns=['sepal length','sepal width','petal length','petal width'])
df
```

Out[114]:

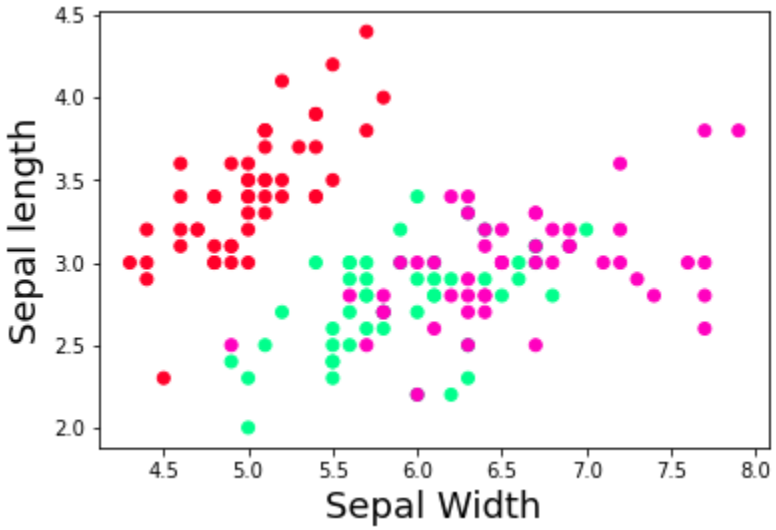
	sepal length	sepal width	petal length	petal width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [115]:
```
x=iris.data
```

In [116]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=iris.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('Sepal Width', fontsize=18)
plt.ylabel('Sepal length', fontsize=18)
```

Out[116]:
```
Text(0, 0.5, 'Sepal length')
```

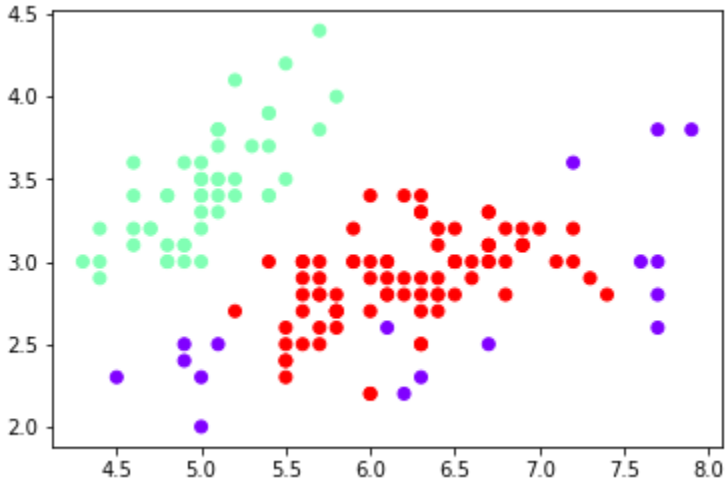


In [117]:
```
from sklearn.cluster import DBSCAN

dbscan = DBSCAN(eps=0.5, algorithm='auto', metric='euclidean')
y = dbscan.fit_predict(x)
```

In [118]:
```
print("Cluster Labels")
print(dbscan.labels_)
```

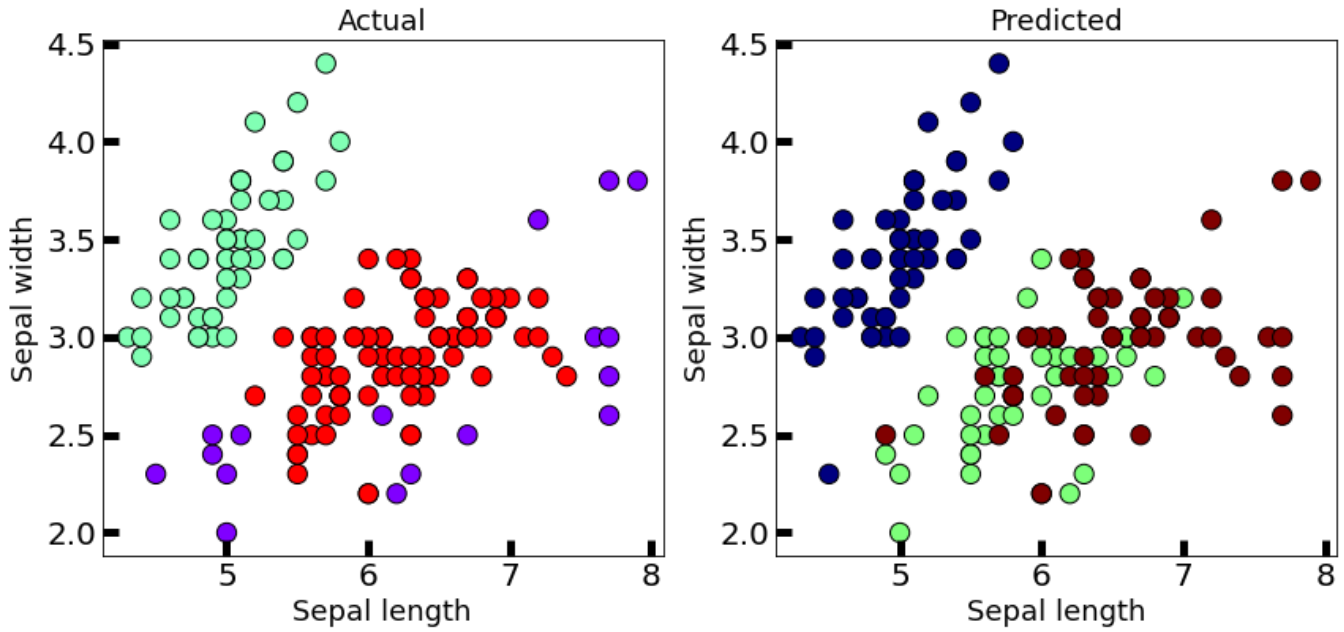
```
Cluster Labels
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0
  0  0  1  1  1  1  1  1  1 -1  1  1 -1  1  1  1  1  1  1  1 -1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1  1  1  1  1  1 -1  1  1
  1  1 -1  1  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]
```

In [119]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=dbscan.labels_, cmap=
'rainbow') #try using cmap='rainbow'
plt.show()
```



In [120]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['sepal length'], y=df['sepal width'], c=y, cmap='rainbow',
edgecolor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['sepal length'], y=df['sepal width'], c=iris.target, cmap=
'jet',edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[120]:
```
Text(0.5, 1.0, 'Predicted')
```



In [121]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, dbscan.labels_)
```

```
The silhouette score is :
```

Out[121]:
```
0.48603419703456835
```

In [122]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, dbscan.labels_)
```

```
The calinski harabasz score is :
```

Out[122]:
```
220.29751498443005
```

In [123]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, dbscan.labels_)
```

```
The davies bouldin score is :
```

Out[123]:
```
7.222448016359586
```

WINE DATASET

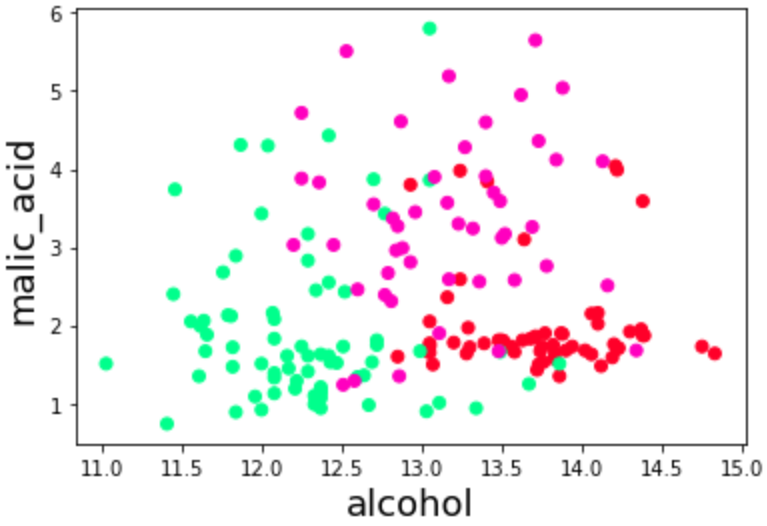
In [124]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_wine
```

In [ ]:
```
wine=load_wine()   #loading iris dataset from sklearn.datasets
wine
```

In [126]:
```
x=wine.data
```

In [127]:
```
df=pd.DataFrame(data=wine.data, columns=['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline'])
df

plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=wine.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('alcohol', fontsize=18)
plt.ylabel('malic_acid', fontsize=18)
```

Out[127]:
```
Text(0, 0.5, 'malic_acid')
```

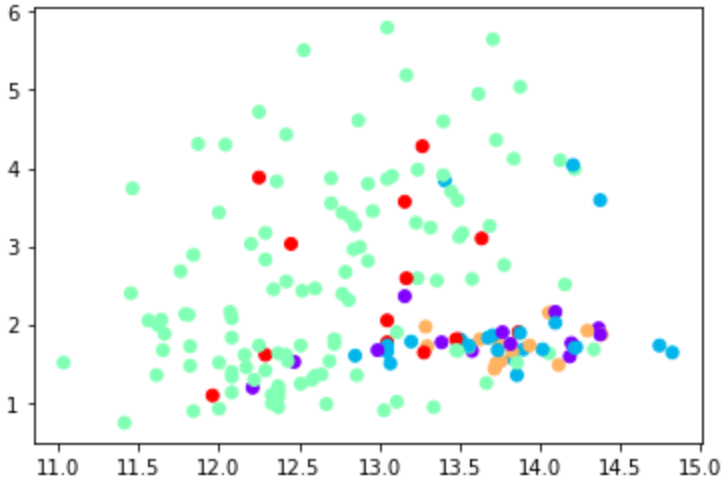


In [128]:
```
from sklearn.cluster import DBSCAN

dbscan = DBSCAN(eps=35, algorithm='auto', metric='euclidean')
y = dbscan.fit_predict(x)
```

In [129]:
```
print("Cluster Labels")
print(dbscan.labels_)
```

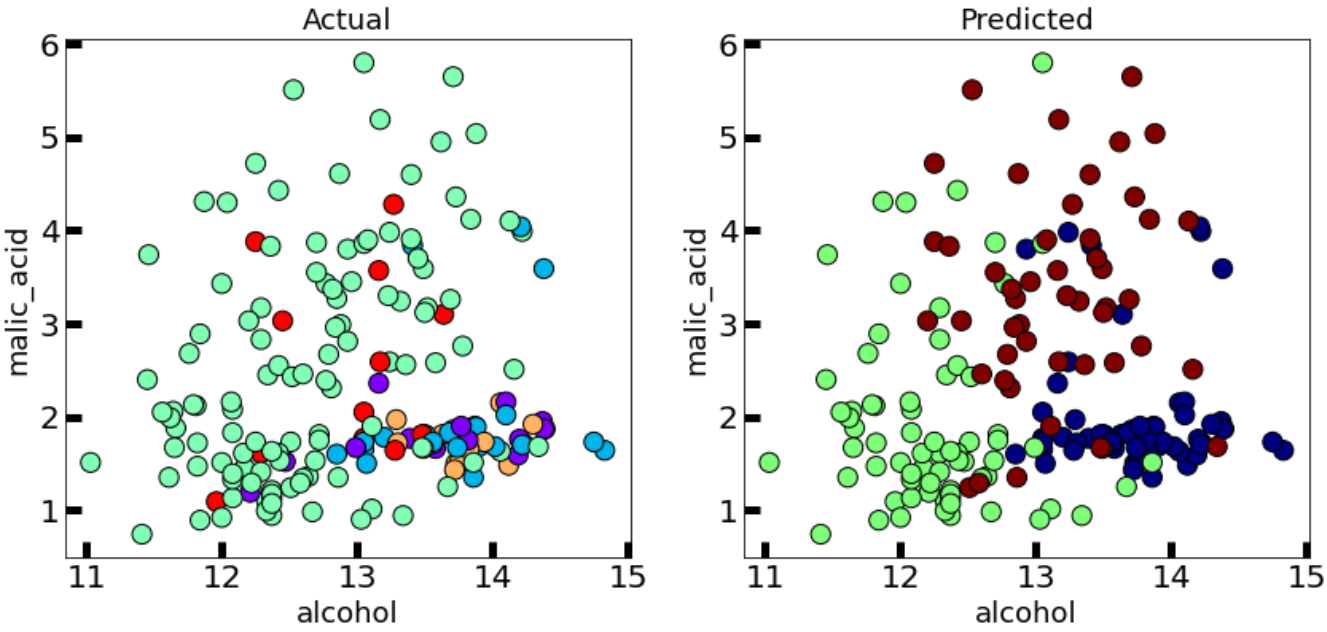
```
Cluster Labels
[ 0  0 -1 -1  1 -1  2  2  0  0 -1  2  2  0 -1  2  2  0 -1  3  1  1  0  0
  3  3 -1  2  3  0  2 -1  0  2  0  3  3  0  0  1  1  0  0  1  3  0  0  0
  0  2  0  2 -1 -1  0  0  0  2  2  1  1  1  1  1  1  1  1  1 -1  3  1
  1 -1  3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -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  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  3  3  1  1  1  1  1  1  1  1  1  1  1  3  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  3  3  1]
```

In [130]:
```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=dbscan.labels_, cmap='rainbow') #try using cmap='rainbow'
plt.show()
```



In [131]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['alcohol'], y=df['malic_acid'], c=y, cmap='rainbow',edgecolor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['alcohol'], y=df['malic_acid'], c=wine.target, cmap='jet', edgecolor='k', s=150)
axes[0].set_xlabel('alcohol', fontsize=18)
axes[0].set_ylabel('malic_acid', fontsize=18)
axes[1].set_xlabel('alcohol', fontsize=18)
axes[1].set_ylabel('malic_acid', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=20)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=20)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[131]:
```
Text(0.5, 1.0, 'Predicted')
```



In [132]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, dbscan.labels_)
```

```
The silhouette score is :
```

Out[132]:
```
0.4413295944891938
```

In [133]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, dbscan.labels_)
```

```
The calinski harabasz score is :
```

Out[133]:
```
208.9449395725058
```

In [134]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, dbscan.labels_)
```

```
The davies bouldin score is :
```

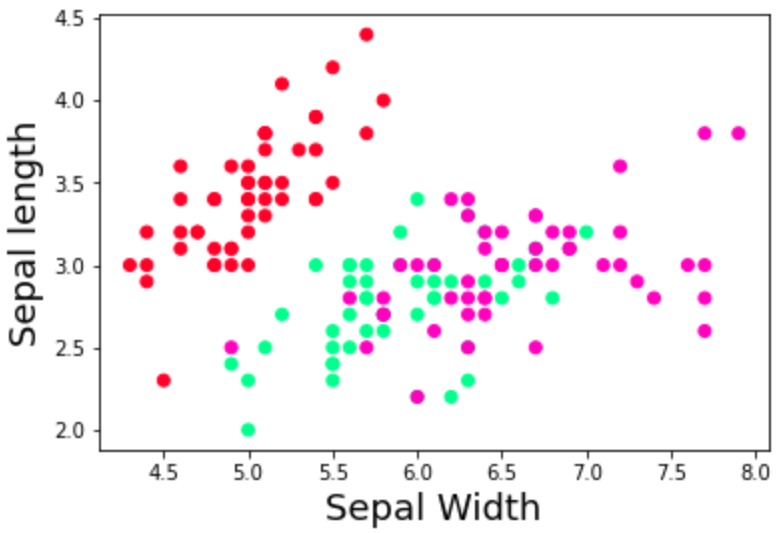
Out[134]:
```
7.812129203041904
```

Density based: OPTICS

IRIS PLANT DATASET

In [135]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris

iris=load_iris()   #loading iris dataset from sklearn.datasets
iris

df=pd.DataFrame(data=iris.data, columns=['sepal length','sepal width','petal length','petal width'])
df

x=iris.data

plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=iris.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('Sepal Width', fontsize=18)
plt.ylabel('Sepal length', fontsize=18)
```

Out[135]:
```
Text(0, 0.5, 'Sepal length')
```

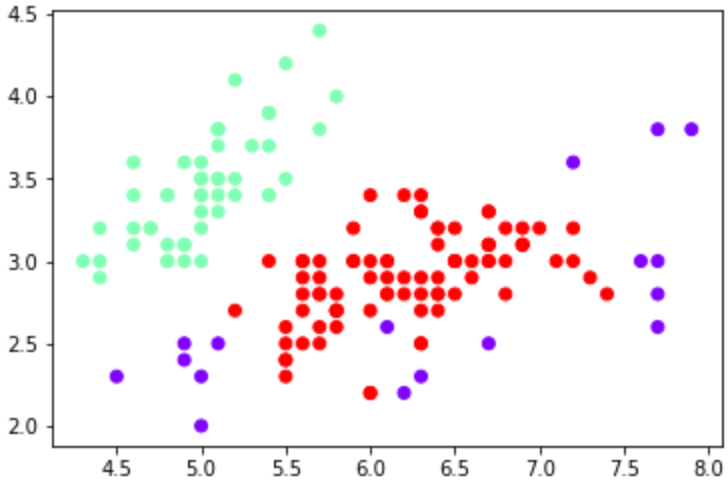


In [136]:
```
from sklearn.cluster import DBSCAN

dbscan = DBSCAN(eps=0.5, algorithm='auto', metric='euclidean')
y = dbscan.fit_predict(x)
```

In [137]:
```
print("Cluster Labels")
print(dbscan.labels_)
```

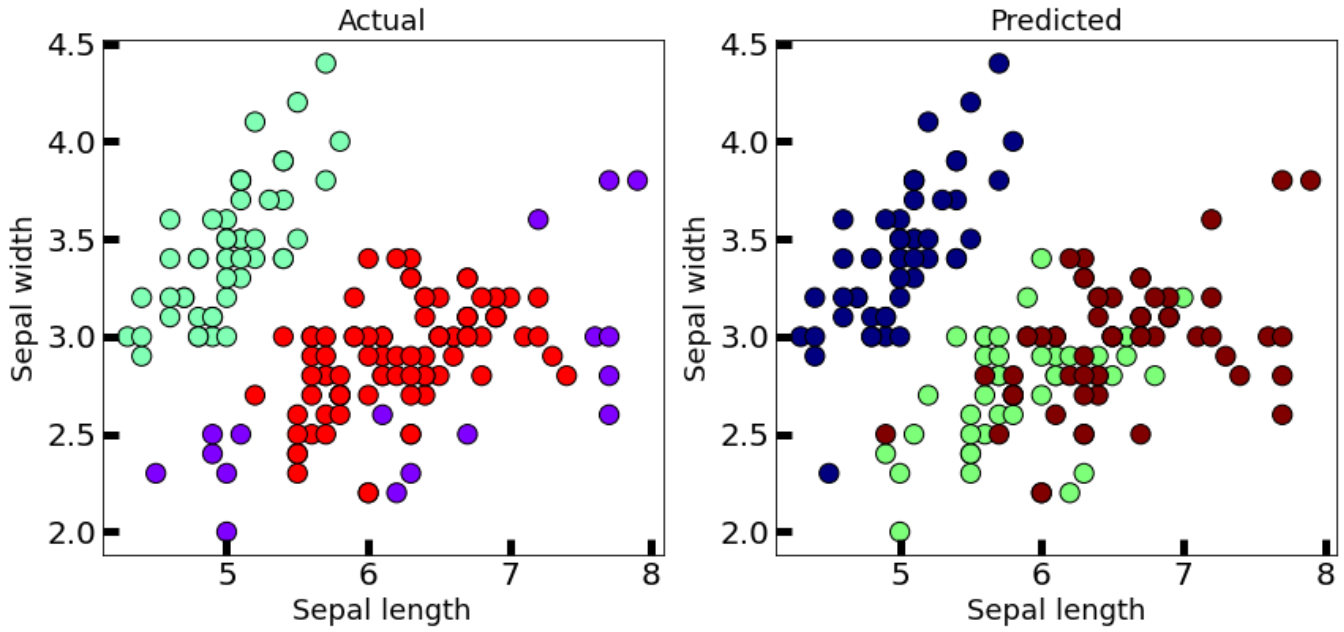
```
Cluster Labels
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0
  0  0  1  1  1  1  1  1  1 -1  1  1 -1  1  1  1  1  1  1  1 -1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1  1  1  1  1  1 -1  1  1
  1  1 -1  1  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]
```

In [138]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=dbscan.labels_, cmap=
'rainbow') #try using cmap='rainbow'
plt.show()
```



In [139]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['sepal length'], y=df['sepal width'], c=y, cmap='rainbow',
edgecolor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['sepal length'], y=df['sepal width'], c=iris.target, cmap=
'jet',edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[139]:
```
Text(0.5, 1.0, 'Predicted')
```



In [140]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, dbscan.labels_)
```

```
The silhouette score is :
```

Out[140]:
```
0.48603419703456835
```

In [141]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, dbscan.labels_)
```

```
The calinski harabasz score is :
```

Out[141]:
```
220.29751498443005
```

In [142]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, dbscan.labels_)
```

```
The davies bouldin score is :
```

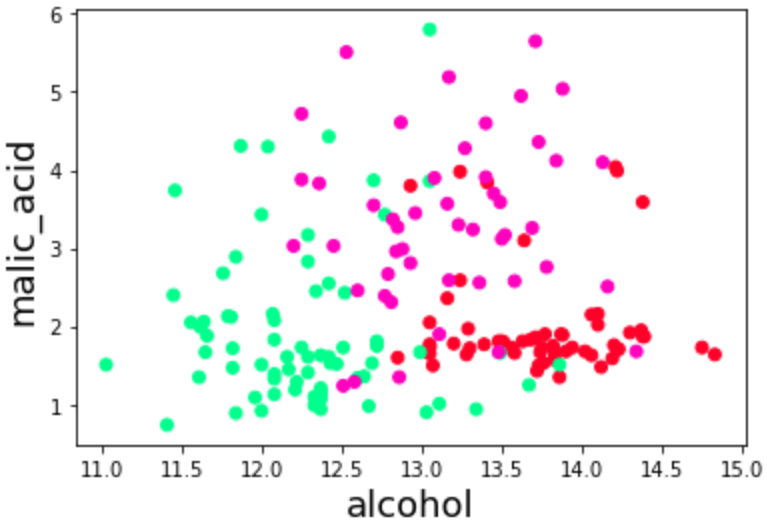
Out[142]:
```
7.222448016359586
```

WINE DATASET

In [143]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_wine
```

In [144]:
```
wine=load_wine()   #loading iris dataset from sklearn.datasets
wine

x=wine.data

df=pd.DataFrame(data=wine.data, columns=['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline'])
df

plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=wine.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('alcohol', fontsize=18)
plt.ylabel('malic_acid', fontsize=18)
```

Out[144]:
```
Text(0, 0.5, 'malic_acid')
```

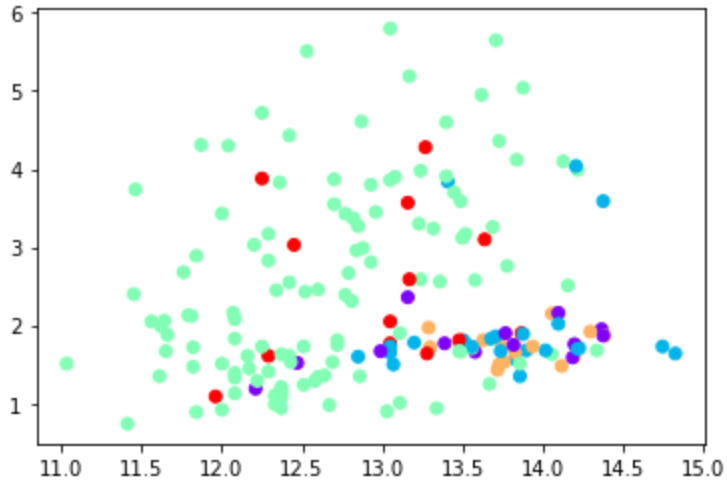


In [145]:
```
from sklearn.cluster import DBSCAN

dbscan = DBSCAN(eps=35, algorithm='auto', metric='euclidean')
y = dbscan.fit_predict(x)
```

In [146]:
```
print("Cluster Labels")
print(dbscan.labels_)
```

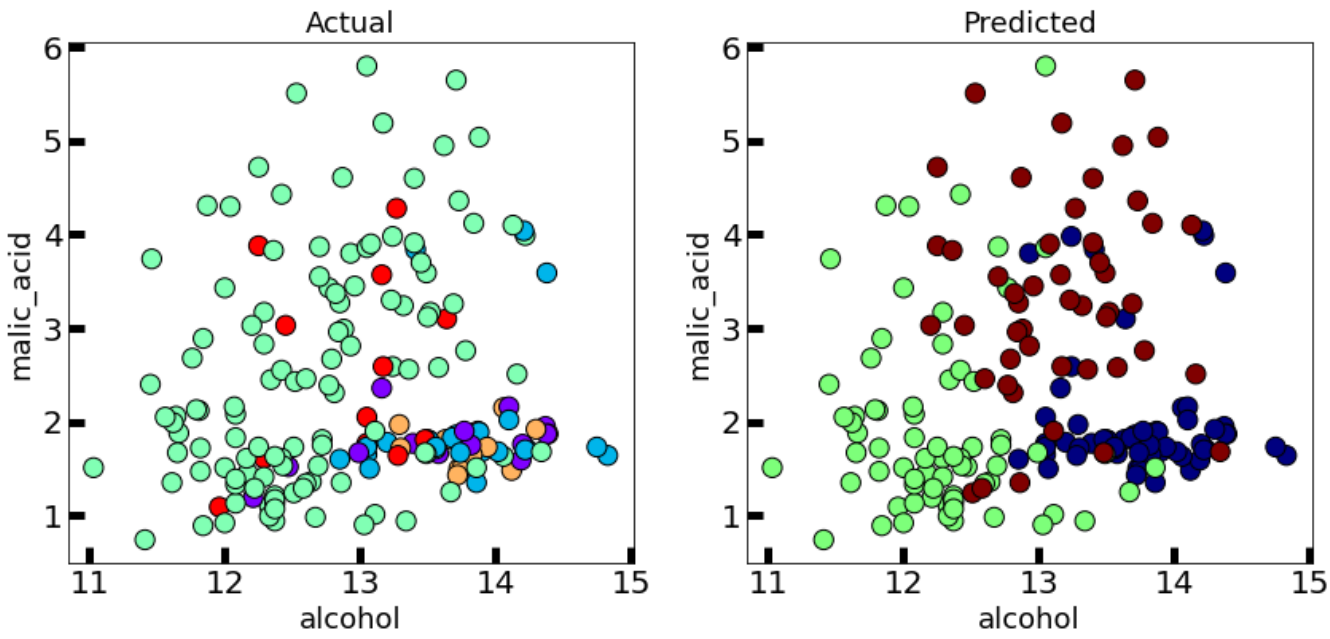
```
Cluster Labels
[ 0  0 -1 -1  1 -1  2  2  0  0 -1  2  2  0 -1  2  2  0 -1  3  1  1  0  0
  3  3 -1  2  3  0  2 -1  0  2  0  3  3  0  0  1  1  0  0  1  3  0  0  0
  0  2  0  2 -1 -1  0  0  0  2  2  1  1  1  1  1  1  1  1  1 -1  3  1
  1 -1  3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -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  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  3  3  1  1  1  1  1  1  1  1  1  1  1  3  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  3  3  1]
```

In [147]:
```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=dbscan.labels_, cmap='rainbo
w') #try using cmap='rainbow'
plt.show()
```



In [148]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['alcohol'], y=df['malic_acid'], c=y, cmap='rainbow',edgeco
lor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['alcohol'], y=df['malic_acid'], c=wine.target, cmap='jet',
edgecolor='k', s=150)
axes[0].set_xlabel('alcohol', fontsize=18)
axes[0].set_ylabel('malic_acid', fontsize=18)
axes[1].set_xlabel('alcohol', fontsize=18)
axes[1].set_ylabel('malic_acid', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[148]:
```
Text(0.5, 1.0, 'Predicted')
```



In [149]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, dbscan.labels_)
```

```
The silhouette score is :
```

Out[149]:
```
0.4413295944891938
```

In [150]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, dbscan.labels_)
```

```
The calinski harabasz score is :
```

Out[150]:
```
208.9449395725058
```

In [151]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, dbscan.labels_)
```

```
The davies bouldin score is :
```

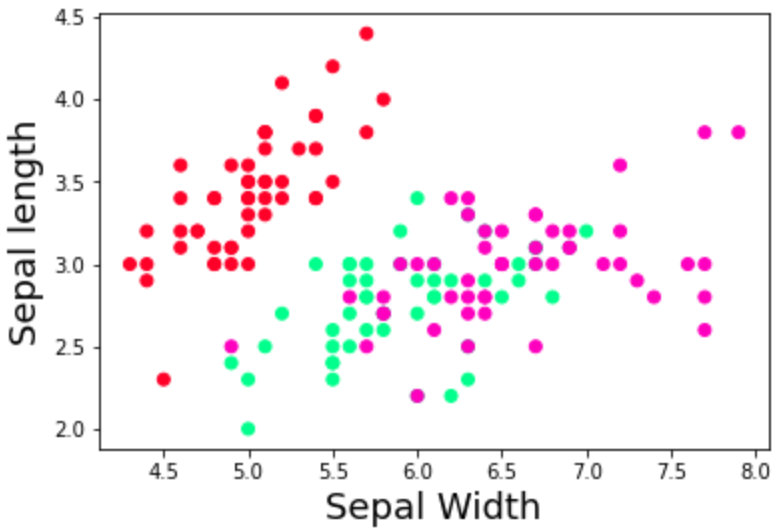
Out[151]:
```
7.812129203041904
```

K-means++

IRIS PLANT DATASET

In [168]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris

iris=load_iris()   #loading iris dataset from sklearn.datasets
iris

df=pd.DataFrame(data=iris.data, columns=['sepal length','sepal width','petal le
ngth','petal width'])
df

x=iris.data

plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=iris.target, cmap='gis
t_rainbow') #try using cmap='rainbow'

plt.xlabel('Sepal Width', fontsize=18)
plt.ylabel('Sepal length', fontsize=18)
```

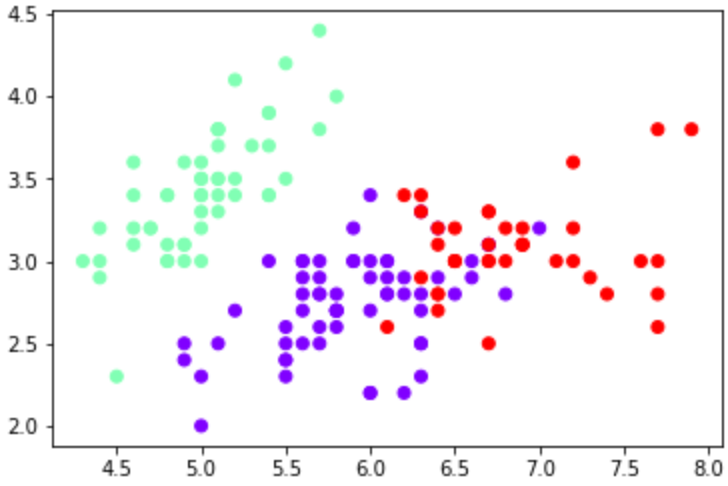
Out[168]:
```
Text(0, 0.5, 'Sepal length')
```



In [169]:
```
kmeans = KMeans(init='k-means++', n_clusters=3, n_init=10, max_iter=300, random
_state=42)
y = kmeans.fit_predict(x)


print("K-Means Cluster Centers")
print(kmeans.cluster_centers_)
print("Cluster Labels")
print(kmeans.labels_)
```

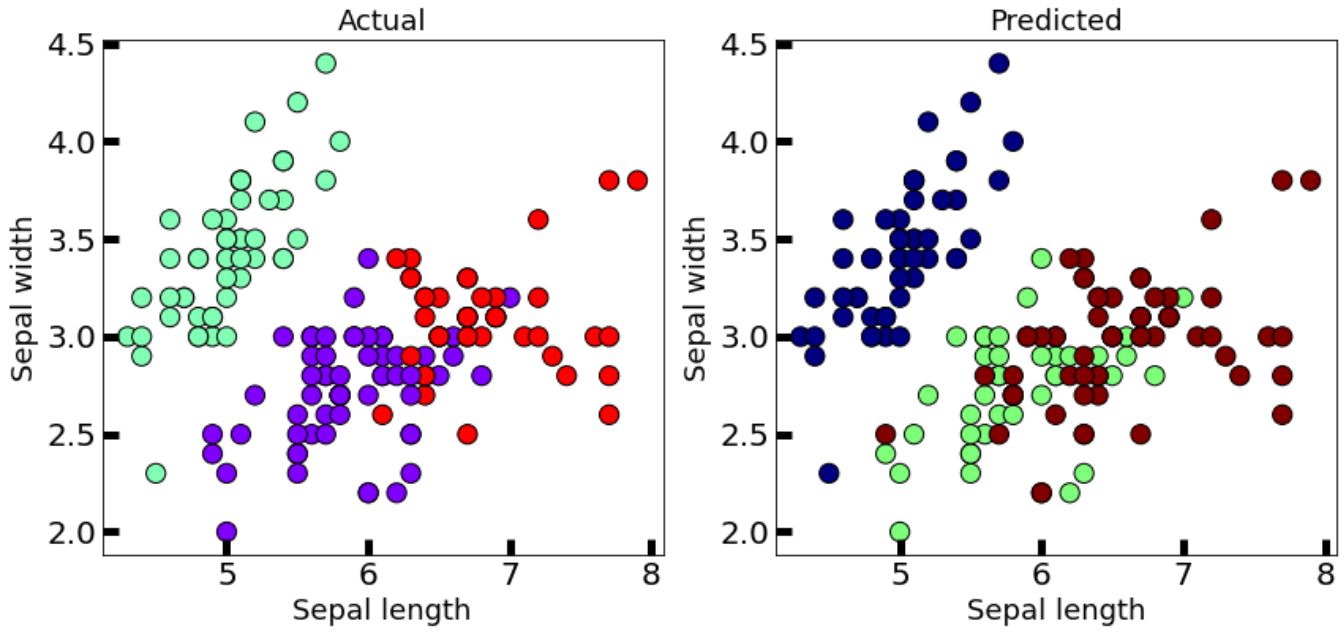
```
K-Means Cluster Centers
[[5.9016129  2.7483871  4.39354839 1.43387097]
 [5.006      3.428      1.462      0.246     ]
 [6.85       3.07368421 5.74210526 2.07105263]]
Cluster Labels
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 2 2 2 0 2 2 2 2
 2 2 0 0 2 2 2 2 0 2 0 2 0 2 2 0 0 2 2 2 2 2 0 2 2 2 2 0 2 2 2 0 2 2 2 0 2
 2 0]
```

In [170]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=kmeans.labels_, cmap=
'rainbow') #try using cmap='rainbow'
plt.show()
```



In [171]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['sepal length'], y=df['sepal width'], c=y, cmap='rainbow',
edgecolor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['sepal length'], y=df['sepal width'], c=iris.target, cmap=
'jet',edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[171]:
```
Text(0.5, 1.0, 'Predicted')
```



In [172]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, kmeans.labels_)
```

```
The silhouette score is :
```

Out[172]:
```
0.5528190123564091
```

In [173]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, kmeans.labels_)
```

```
The calinski harabasz score is :
```

Out[173]:
```
561.62775662962
```

In [174]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, kmeans.labels_)
```

```
The davies bouldin score is :
```

Out[174]:
```
0.6619715465007511
```

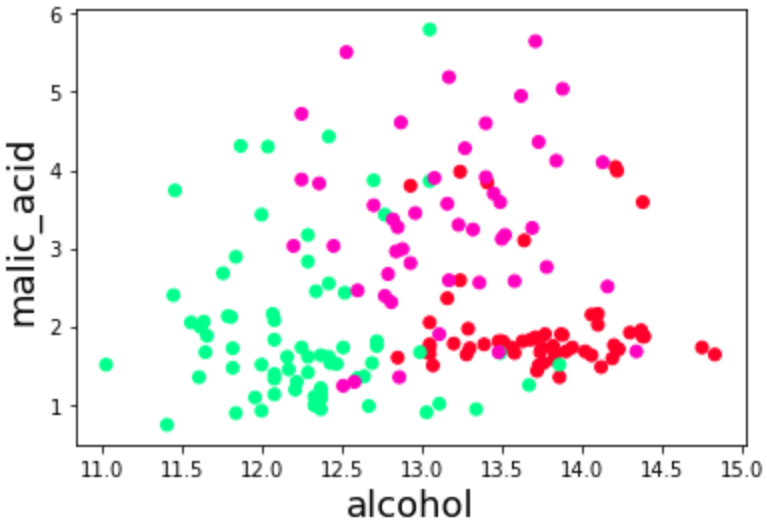
In [175]:
```
print("It is observed that TSS=SSE+SSB is a constant. Hence we will calculate t
he TSS ans substract SSE from it to get SSB")

print("The value of SSE is: ")
print(kmeans.inertia_)



# Finding the overall centroid of the data points
centers = kmeans.cluster_centers_
center_x = []
for center in centers:
  center_x.append(center[0])
center_x
overall_center = sum(center_x)/len(center_x)

tss = 0
for i in range(len(df)):
  a = df.iloc[i][0] - overall_center
  b = pow(a,2)
  tss = tss+b



print("The value of SSB is: ")
print(tss - kmeans.inertia_)
```

```
It is observed that TSS=SSE+SSB is a constant. Hence we will calculate the TSS a
ns substract SSE from it to get SSB
The value of SSE is:
78.85144142614601
The value of SSB is:
24.18035246910202
```

WINE DATASET

In [176]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_wine


wine=load_wine()   #loading iris dataset from sklearn.datasets
wine

x=wine.data

df=pd.DataFrame(data=wine.data, columns=['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline'])
df

plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=wine.target, cmap='gist_rain
bow') #try using cmap='rainbow'

plt.xlabel('alcohol', fontsize=18)
plt.ylabel('malic_acid', fontsize=18)
```

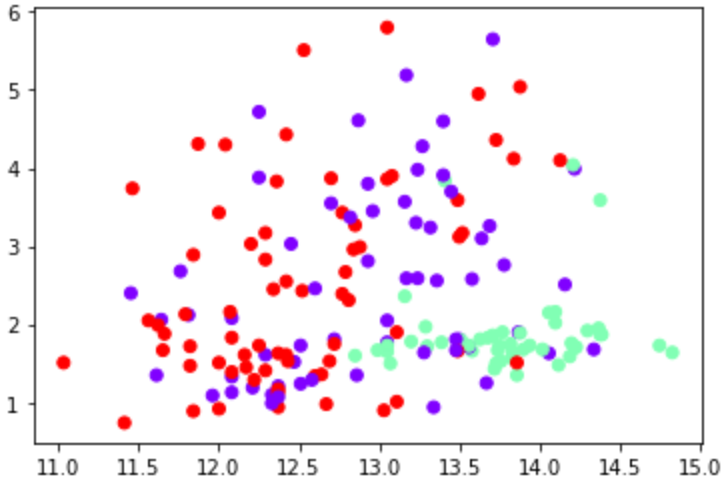
Out[176]:
```
Text(0, 0.5, 'malic_acid')
```



In [177]:
```
kmeans = KMeans(init='k-means++', n_clusters=3, n_init=10, max_iter=300, random
_state=42)
y = kmeans.fit_predict(x)


print("K-Means Cluster Centers")
print(kmeans.cluster_centers_)
print("Cluster Labels")
print(kmeans.labels_)
```

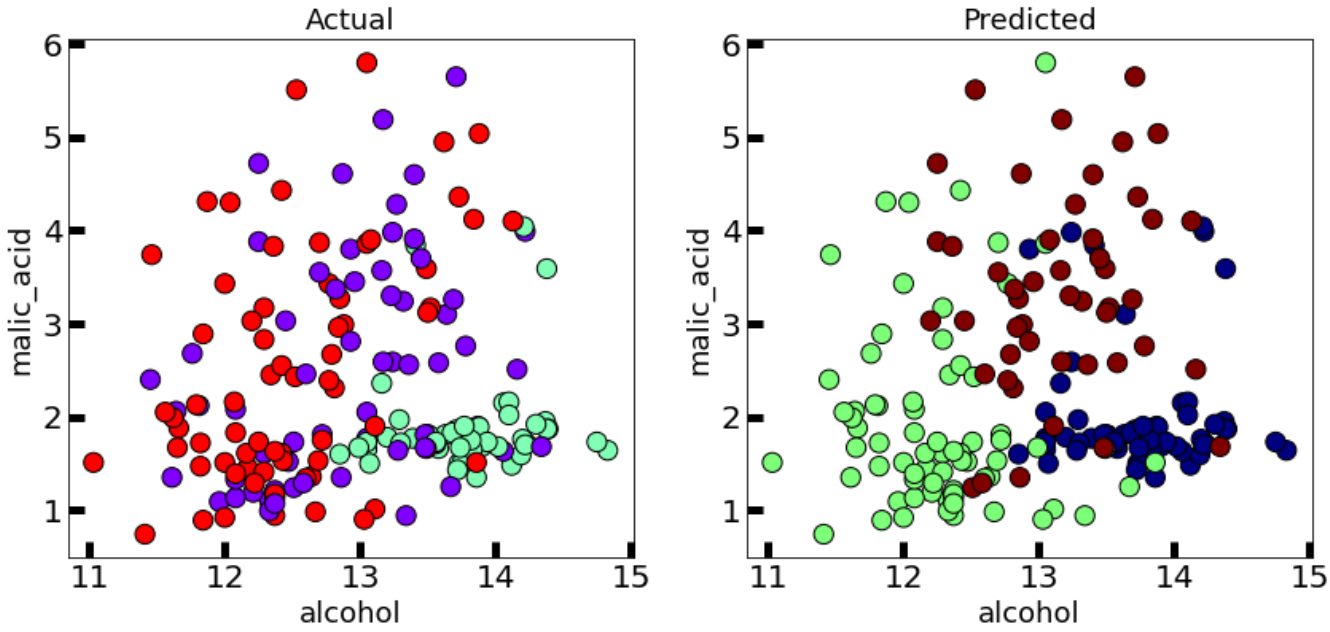
```
K-Means Cluster Centers
[[1.29298387e+01 2.50403226e+00 2.40806452e+00 1.98903226e+01
  1.03596774e+02 2.11112903e+00 1.58403226e+00 3.88387097e-01
  1.50338710e+00 5.65032258e+00 8.83967742e-01 2.36548387e+00
  7.28338710e+02]
 [1.38044681e+01 1.88340426e+00 2.42617021e+00 1.70234043e+01
  1.05510638e+02 2.86723404e+00 3.01425532e+00 2.85319149e-01
  1.91042553e+00 5.70255319e+00 1.07829787e+00 3.11404255e+00
  1.19514894e+03]
 [1.25166667e+01 2.49420290e+00 2.28855072e+00 2.08231884e+01
  9.23478261e+01 2.07072464e+00 1.75840580e+00 3.90144928e-01
  1.45188406e+00 4.08695651e+00 9.41159420e-01 2.49072464e+00
  4.58231884e+02]]
Cluster Labels
[1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0
 1 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 2 0 2 2 0 2 2 0 0 0 2 2 1
 0 2 2 2 0 2 2 0 0 2 2 2 2 2 0 0 2 2 2 2 2 0 0 2 0 2 0 2 2 2 0 2 2 2 2 0 2
 2 0 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 0 2 2 0 0 0 0 2 2 2 0 0 2 2 0 0 2 0
 0 2 2 2 2 2 0 0 0 2 0 0 0 2 0 2 0 0 2 0 0 0 0 2 2 0 0 0 0 0 2]
```

In [178]:
```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=kmeans.labels_, cmap='rainbo
w') #try using cmap='rainbow'
plt.show()
```



In [179]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['alcohol'], y=df['malic_acid'], c=y, cmap='rainbow',edgeco
lor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['alcohol'], y=df['malic_acid'], c=wine.target, cmap='jet',
edgecolor='k', s=150)
axes[0].set_xlabel('alcohol', fontsize=18)
axes[0].set_ylabel('malic_acid', fontsize=18)
axes[1].set_xlabel('alcohol', fontsize=18)
axes[1].set_ylabel('malic_acid', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[179]: Text(0.5, 1.0, 'Predicted')



In [180]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, kmeans.labels_)
```

```
The silhouette score is :
```

Out[180]: 0.5711381937868844

In [181]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, kmeans.labels_)
```

```
The calinski harabasz score is :
```

Out[181]: 561.815657860671

In [182]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, kmeans.labels_)
```

```
The davies bouldin score is :
```

Out[182]:
```
0.5342431775436273
```

In [183]:
```
print("It is observed that TSS=SSE+SSB is a constant. Hence we will calculate t
he TSS ans substract SSE from it to get SSB")

print("The value of SSE is: ")
print(kmeans.inertia_)



# Finding the overall centroid of the data points
centers = kmeans.cluster_centers_
center_x = []
for center in centers:
  center_x.append(center[0])
center_x
overall_center = sum(center_x)/len(center_x)

tss = 0
for i in range(len(df)):
  a = df.iloc[i][0] - overall_center
  b = pow(a,2)
  tss = tss+b



print("The value of SSB is: ")
print(tss - kmeans.inertia_)
```

```
It is observed that TSS=SSE+SSB is a constant. Hence we will calculate the TSS a
ns substract SSE from it to get SSB
The value of SSE is:
2370689.686782968
The value of SSB is:
-2370571.8053313834
```

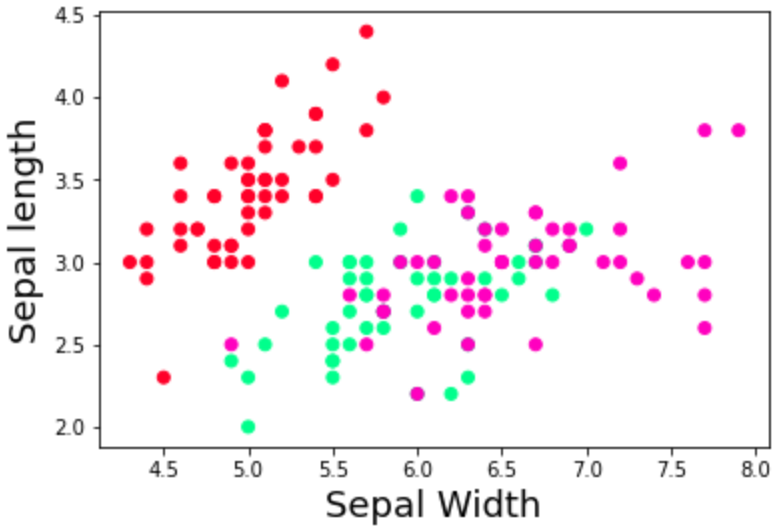
Bisecting K-means

IRIS PLANT DATASET

In [184]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
```

In [185]:
```
iris=load_iris()   #loading iris dataset from sklearn.datasets
iris

df=pd.DataFrame(data=iris.data, columns=['sepal length','sepal width','petal length','petal width'])
df

x=iris.data

plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=iris.target, cmap='gist_rainbow') #try using cmap='rainbow'

plt.xlabel('Sepal Width', fontsize=18)
plt.ylabel('Sepal length', fontsize=18)
```

Out[185]:
```
Text(0, 0.5, 'Sepal length')
```

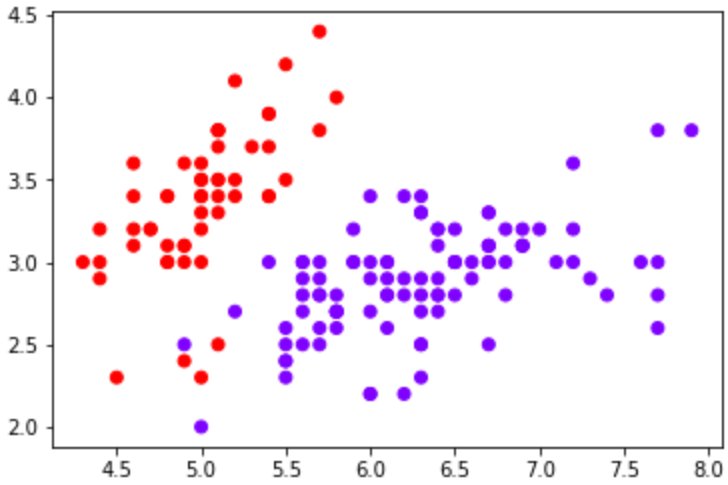


In [186]:
```
from sklearn.cluster import KMeans
import numpy as np

K = 2
current_clusters = 1
split = 0
while current_clusters != K:
    kmeans = KMeans(n_clusters=2).fit(x)
    current_clusters += 1
    split += 1
    cluster_centers = kmeans.cluster_centers_
    sse = [0]*2
    for point, label in zip(x, kmeans.labels_):
        sse[label] += np.square(point-cluster_centers[label]).sum()
    chosen_cluster = np.argmax(sse, axis=0)
    chosen_cluster_data = x[kmeans.labels_ == chosen_cluster]
    x = chosen_cluster_data
```

In [187]:
```
print("K-Means Cluster Centers")
print(kmeans.cluster_centers_)
print("Cluster Labels")
print(kmeans.labels_)
```

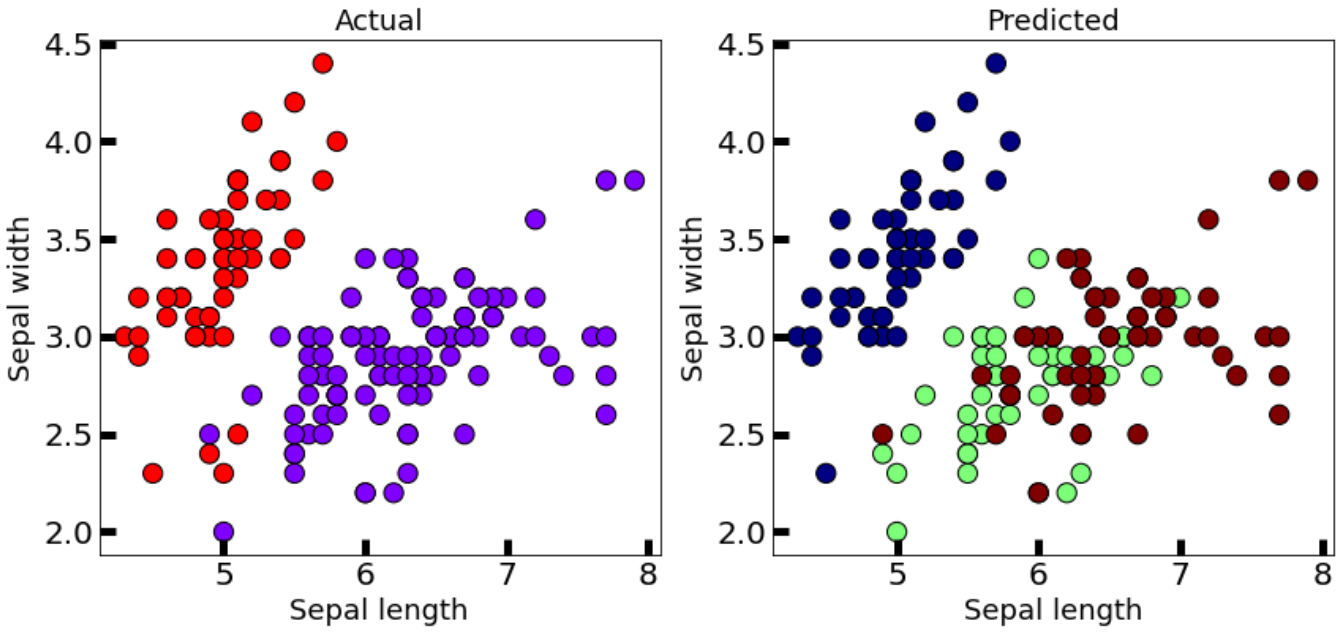
```
K-Means Cluster Centers
[[6.30103093 2.88659794 4.95876289 1.69587629]
 [5.00566038 3.36981132 1.56037736 0.29056604]]
Cluster Labels
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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]
```

In [188]:
```
plt.scatter(x=df['sepal length'], y=df['sepal width'] ,c=kmeans.labels_, cmap=
'rainbow') #try using cmap='rainbow'
plt.show()
```



In [189]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['sepal length'], y=df['sepal width'], c=kmeans.labels_, cm
ap='rainbow',edgecolor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['sepal length'], y=df['sepal width'], c=iris.target, cmap=
'jet',edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[189]:
```
Text(0.5, 1.0, 'Predicted')
```



In [190]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, kmeans.labels_[0:97])
```

```
The silhouette score is :
```

Out[190]:
```
0.3093066204565962
```

In [191]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, kmeans.labels_[0:97])
```

```
The calinski harabasz score is :
```

Out[191]:
```
61.177251757411454
```

In [192]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, kmeans.labels_[0:97])
```

```
The davies bouldin score is :
```

Out[192]:
```
1.0999710250083434
```

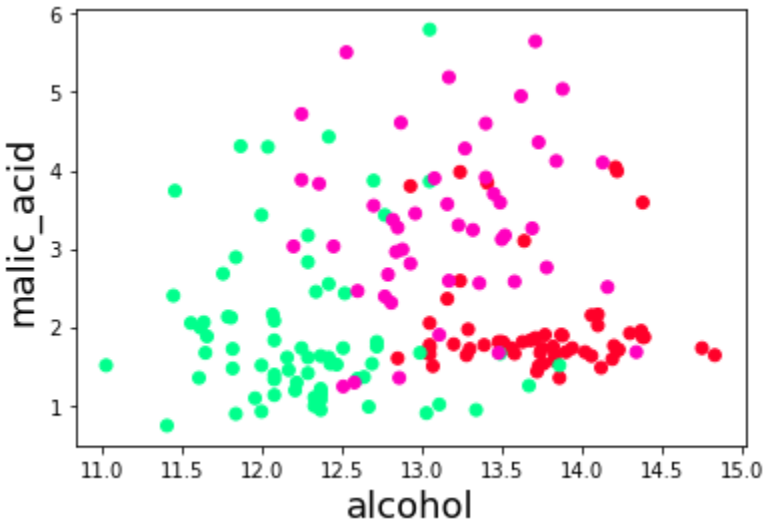
In [193]:
```
print("It is observed that TSS=SSE+SSB is a constant. Hence we will calculate t
he TSS ans substract SSE from it to get SSB")

print("The value of SSE is: ")
print(kmeans.inertia_)



# Finding the overall centroid of the data points
centers = kmeans.cluster_centers_
center_x = []
for center in centers:
  center_x.append(center[0])
center_x
overall_center = sum(center_x)/len(center_x)

tss = 0
for i in range(len(df)):
  a = df.iloc[i][0] - overall_center
  b = pow(a,2)
  tss = tss+b



print("The value of SSB is: ")
print(tss - kmeans.inertia_)
```

```
It is observed that TSS=SSE+SSB is a constant. Hence we will calculate the TSS a
ns substract SSE from it to get SSB
The value of SSE is:
152.34795176035792
The value of SSB is:
-44.76532060227591
```

WINE DATASET

In [ ]:
```
#importing libraries
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.datasets import load_wine


wine=load_wine()   #loading iris dataset from sklearn.datasets
wine

x=wine.data

df=pd.DataFrame(data=wine.data, columns=['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline'])
df

plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=wine.target, cmap='gist_rain
bow') #try using cmap='rainbow'

plt.xlabel('alcohol', fontsize=18)
plt.ylabel('malic_acid', fontsize=18)
```

Out[ ]:
```
Text(0, 0.5, 'malic_acid')
```



In [ ]:

```
from sklearn.cluster import KMeans
import numpy as np

K = 2
current_clusters = 1
split = 0
while current_clusters != K:
    kmeans = KMeans(n_clusters=2).fit(x)
    current_clusters += 1
    split += 1
    cluster_centers = kmeans.cluster_centers_
    sse = [0]*2
    for point, label in zip(x, kmeans.labels_):
        sse[label] += np.square(point-cluster_centers[label]).sum()
    chosen_cluster = np.argmax(sse, axis=0)
    chosen_cluster_data = x[kmeans.labels_ == chosen_cluster]
    x = chosen_cluster_data
```

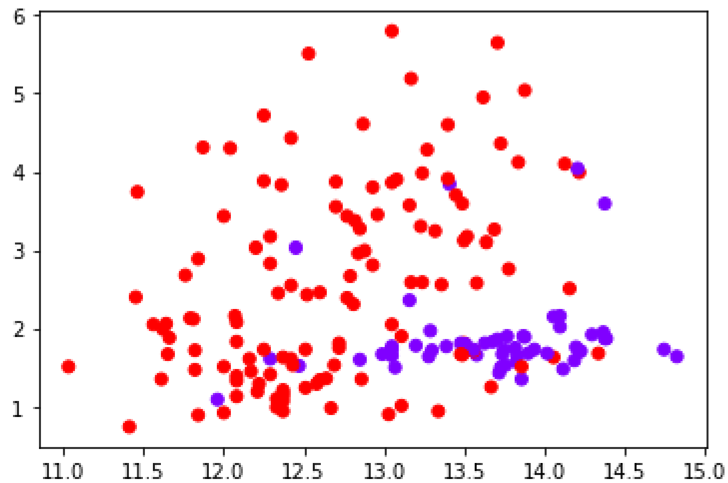
In [ ]:

```
print("K-Means Cluster Centers")
print(kmeans.cluster_centers_)
print("Cluster Labels")
print(kmeans.labels_)
```

```
K-Means Cluster Centers
[[1.36665455e+01 1.87072727e+00 2.42781818e+00 1.74527273e+01
  1.06290909e+02 2.81618182e+00 2.89654545e+00 2.92909091e-01
  1.89690909e+00 5.52036364e+00 1.06665455e+00 3.06672727e+00
  1.15172727e+03]
 [1.27028455e+01 2.54455285e+00 2.33910569e+00 2.04081301e+01
  9.68130081e+01 2.06211382e+00 1.64146341e+00 3.92682927e-01
  1.45406504e+00 4.85138211e+00 9.08617886e-01 2.40821138e+00
  5.65869919e+02]]
Cluster Labels
[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0
 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 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 1 1
 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

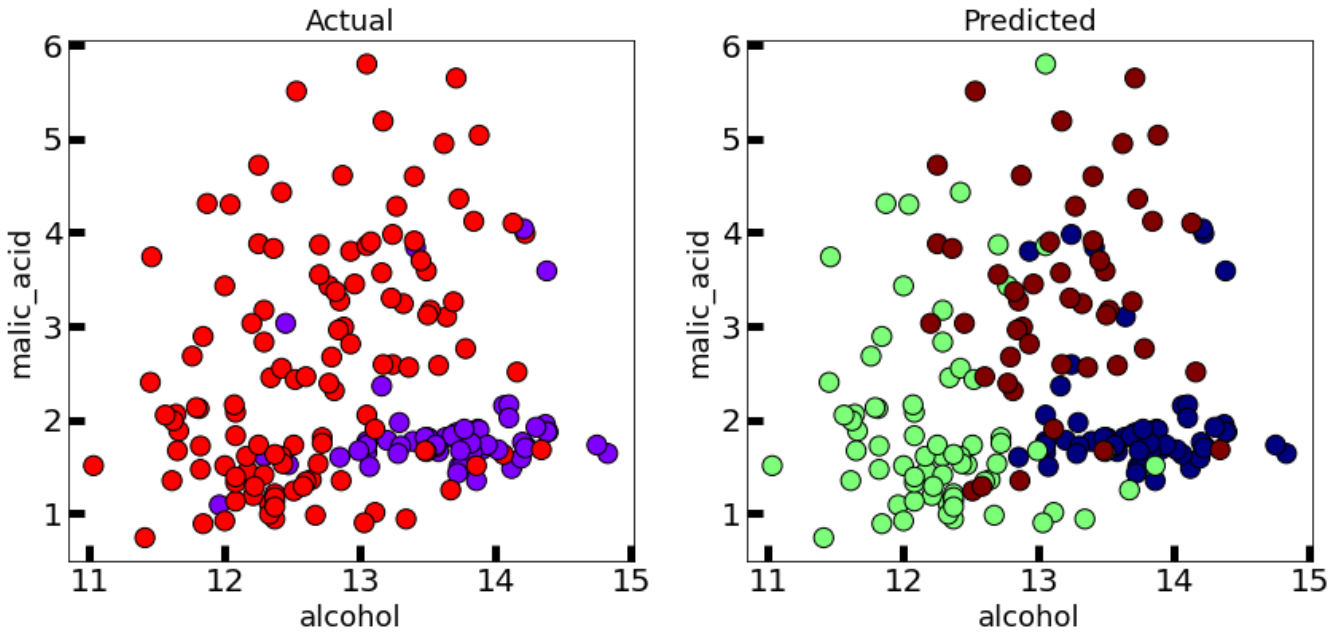
In [ ]:

```
plt.scatter(x=df['alcohol'], y=df['malic_acid'] ,c=kmeans.labels_, cmap='rainbo
w') #try using cmap='rainbow'
plt.show()
```



In [ ]:
```
fig, axes = plt.subplots(1, 2, figsize=(14,6))
axes[0].scatter(x=df['alcohol'], y=df['malic_acid'], c=kmeans.labels_, cmap='ra
inbow',edgecolor='k', s=150) #you can also try cmap='rainbow'
axes[1].scatter(x=df['alcohol'], y=df['malic_acid'], c=wine.target, cmap='jet',
edgecolor='k', s=150)
axes[0].set_xlabel('alcohol', fontsize=18)
axes[0].set_ylabel('malic_acid', fontsize=18)
axes[1].set_xlabel('alcohol', fontsize=18)
axes[1].set_ylabel('malic_acid', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[1].tick_params(direction='in', length=10, width=5, colors='k', labelsize=2
0)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

Out[ ]:
```
Text(0.5, 1.0, 'Predicted')
```



In [ ]:
```
from sklearn.metrics import silhouette_score
print("The silhouette score is :")
silhouette_score(x, kmeans.labels_[0:123])
```

```
The silhouette score is :
```

Out[ ]:
```
0.003840256949844042
```

In [ ]:
```
from sklearn.metrics import calinski_harabasz_score
print("The calinski harabasz score is :")
calinski_harabasz_score(x, kmeans.labels_[0:123])
```

```
The calinski harabasz score is :
```

Out[ ]:
```
1.0661966700358758
```

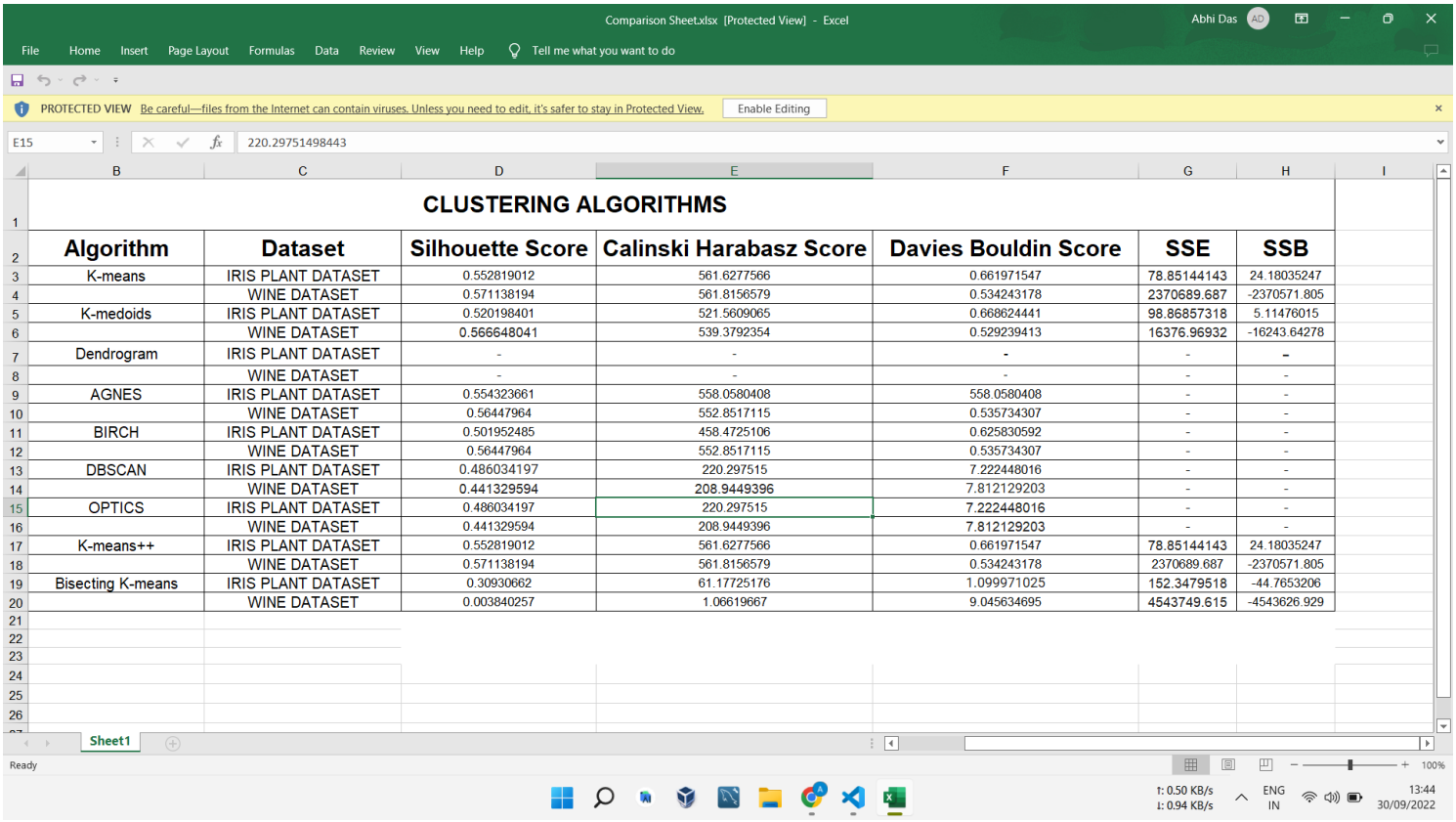
In [ ]:
```
from sklearn.metrics import davies_bouldin_score
print("The davies bouldin score is :")
davies_bouldin_score(x, kmeans.labels_[0:123])
```

```
The davies bouldin score is :
```

Out[ ]:
```
9.045634694626461
```

In [ ]:
```
print("It is observed that TSS=SSE+SSB is a constant. Hence we will calculate t
he TSS ans substract SSE from it to get SSB")

print("The value of SSE is: ")
print(kmeans.inertia_)



# Finding the overall centroid of the data points
centers = kmeans.cluster_centers_
center_x = []
for center in centers:
  center_x.append(center[0])
center_x
overall_center = sum(center_x)/len(center_x)

tss = 0
for i in range(len(df)):
  a = df.iloc[i][0] - overall_center
  b = pow(a,2)
  tss = tss+b



print("The value of SSB is: ")
print(tss - kmeans.inertia_)
```

```
It is observed that TSS=SSE+SSB is a constant. Hence we will calculate the TSS a
ns substract SSE from it to get SSB
The value of SSE is:
4543749.614531862
The value of SSB is:
-4543626.9290532945
```


Comparison Sheet.xlsx [Protected View] - Excel
Abhi Das
File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do
PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable Editing
E15
220.29751498443
CLUSTERING ALGORITHMS
Algorithm | Dataset | Silhouette Score | Calinski Harabasz Score | Davies Bouldin Score | SSE | SSB
K-means | IRIS PLANT DATASET | 0.552819012 | 561.6277566 | 0.661971547 | 78.85144143 | 24.18035247
WINE DATASET | 0.571138194 | 561.8156579 | 0.534243178 | 2370689.687 | -2370571.805
K-medoids | IRIS PLANT DATASET | 0.520198401 | 521.5609065 | 0.668624441 | 98.86857318 | 5.11476015
WINE DATASET | 0.566648041 | 539.3792354 | 0.529239413 | 16376.96932 | -16243.64278
Dendrogram | IRIS PLANT DATASET | - | - | - | - | -
WINE DATASET | - | - | - | - | -
AGNES | IRIS PLANT DATASET | 0.554323661 | 558.0580408 | 558.0580408 | - | -
WINE DATASET | 0.56447964 | 552.8517115 | 0.535734307 | - | -
BIRCH | IRIS PLANT DATASET | 0.501952485 | 458.4725106 | 0.625830592 | - | -
WINE DATASET | 0.56447964 | 552.8517115 | 0.535734307 | - | -
DBSCAN | IRIS PLANT DATASET | 0.486034197 | 220.297515 | 7.222448016 | - | -
WINE DATASET | 0.441329594 | 208.9449396 | 7.812129203 | - | -
OPTICS | IRIS PLANT DATASET | 0.486034197 | 220.297515 | 7.222448016 | - | -
WINE DATASET | 0.441329594 | 208.9449396 | 7.812129203 | - | -
K-means++ | IRIS PLANT DATASET | 0.552819012 | 561.6277566 | 0.661971547 | 78.85144143 | 24.18035247
WINE DATASET | 0.571138194 | 561.8156579 | 0.534243178 | 2370689.687 | -2370571.805
Bisecting K-means | IRIS PLANT DATASET | 0.30930662 | 61.17725176 | 1.099971025 | 152.3479518 | -44.7653206
WINE DATASET | 0.003840257 | 1.06619667 | 9.045634695 | 4543749.615 | -4543626.929
Sheet1
Ready
100%
↑: 0.50 KB/s
↓: 0.94 KB/s
ENG
IN
13:44
30/09/2022