Aula 11 - Exercício prático aprendizado supervisionado – parte2 Vitor

Galioti Martini

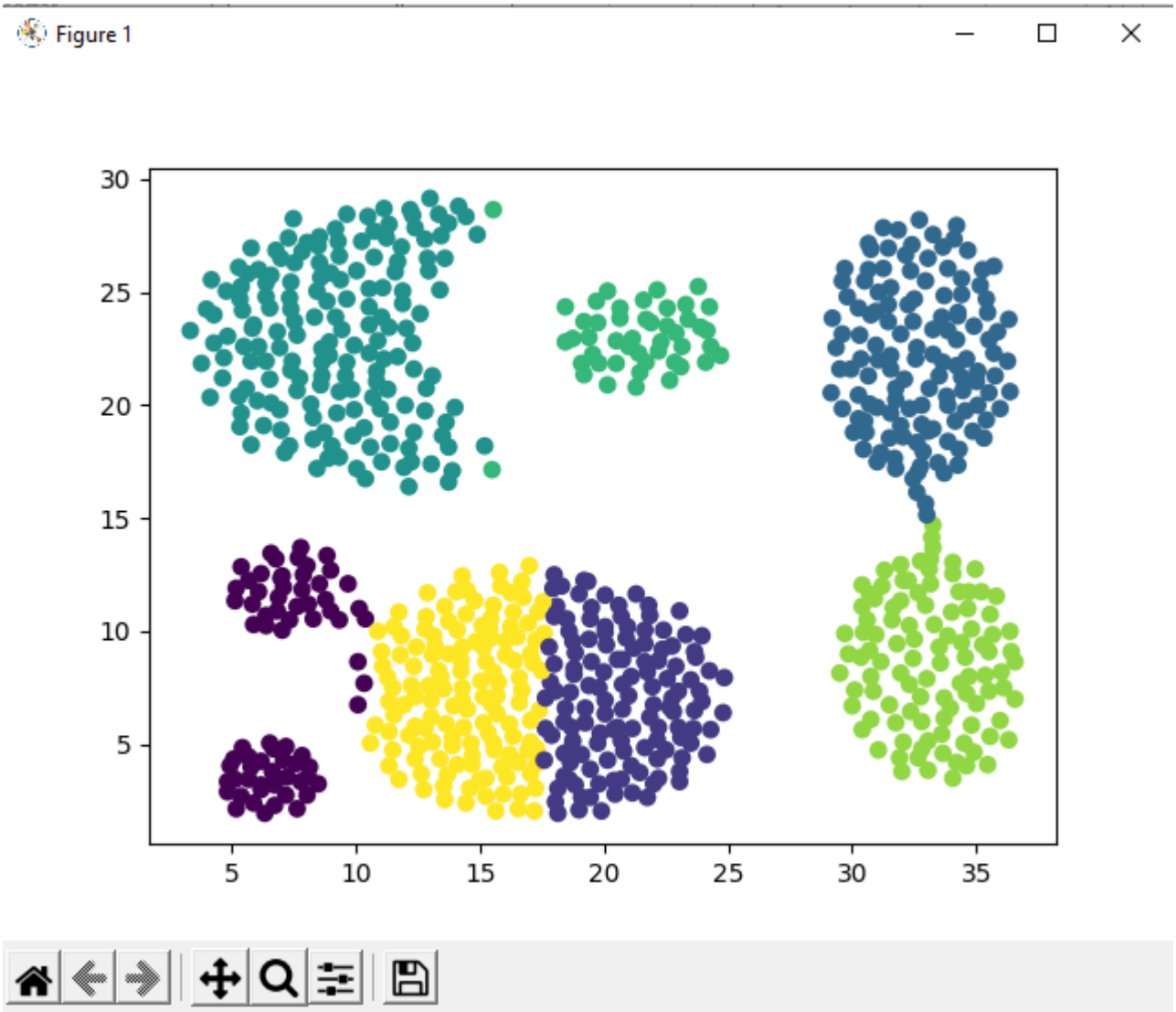
135543

Todas as implementações a seguir foram feitas utilizando a biblioteca sklearn:

Dataset: Aggregation

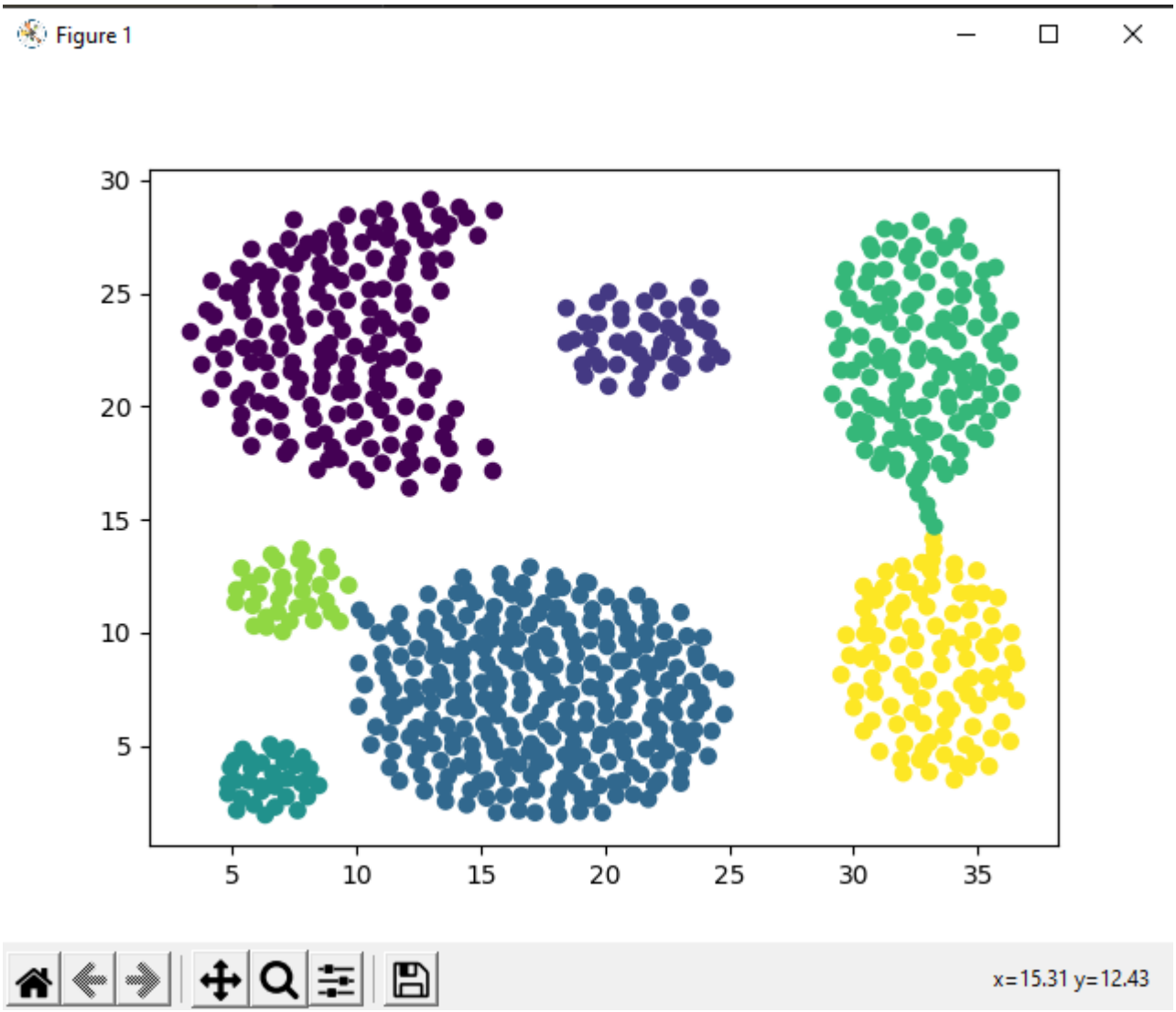
KMeans:

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

data = pd.read_csv('Aggregation.csv')
X = np.array(data)
kmeans = KMeans(n_clusters=int(max(data['C'])), random_state = 0).fit(X)
plt.scatter(X[:,0], X[:,1],c=kmeans.labels_)
plt.show()
```



SingleLinkage:

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

data = pd.read_csv('Aggregation.csv')
X = np.array(data)
clustering = AgglomerativeClustering(n_clusters=int(max(data['C'])),linkage =
'single').fit(X)
plt.scatter(X[:,0], X[:,1],c=clustering.labels_)
plt.show()
```

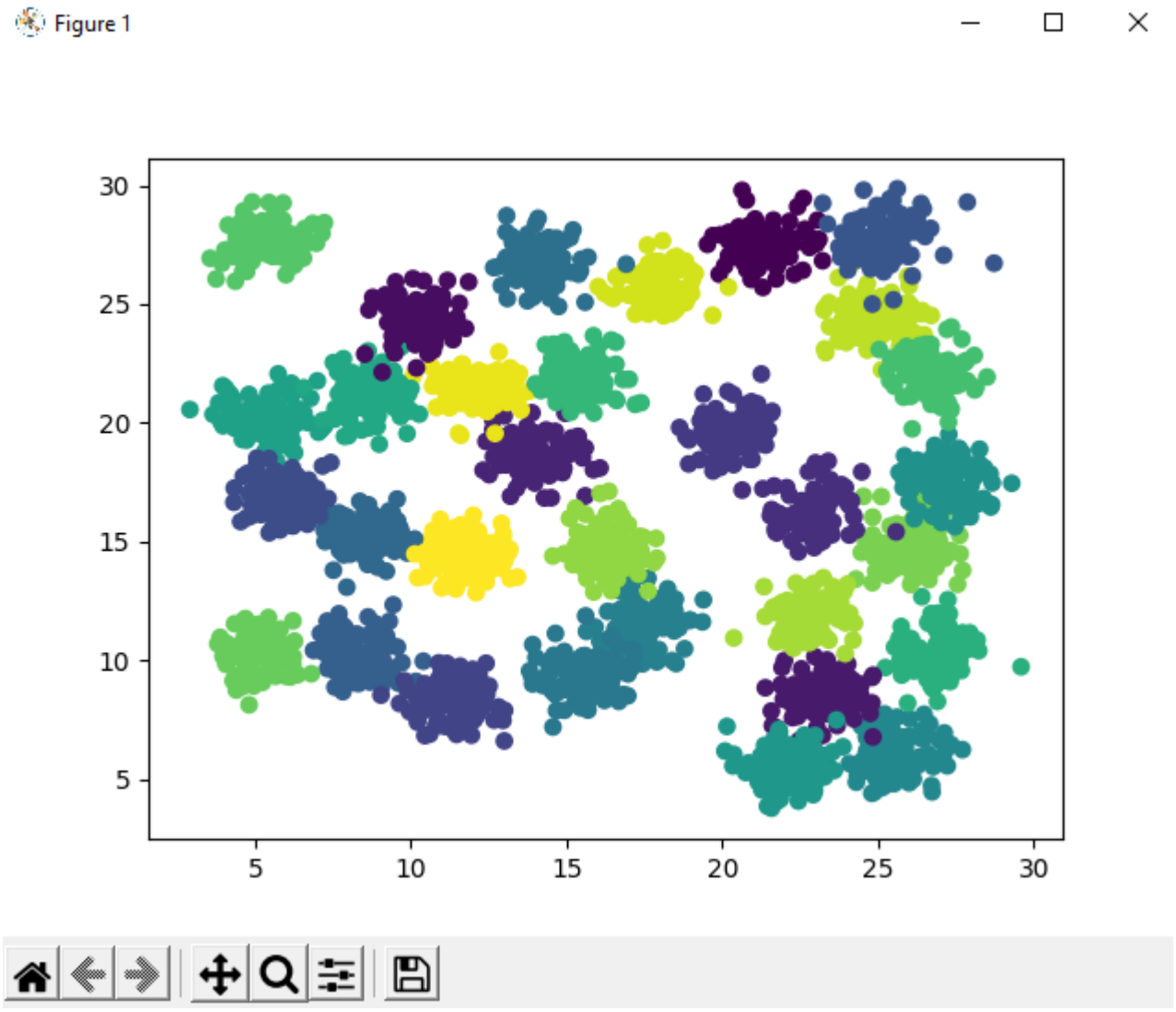


Os dois algoritmos geraram os mesmos grupos, porém partições diferentes. No entanto, os elementos ficaram bem distribuídos nos dois gráficos.

Dataset: D31

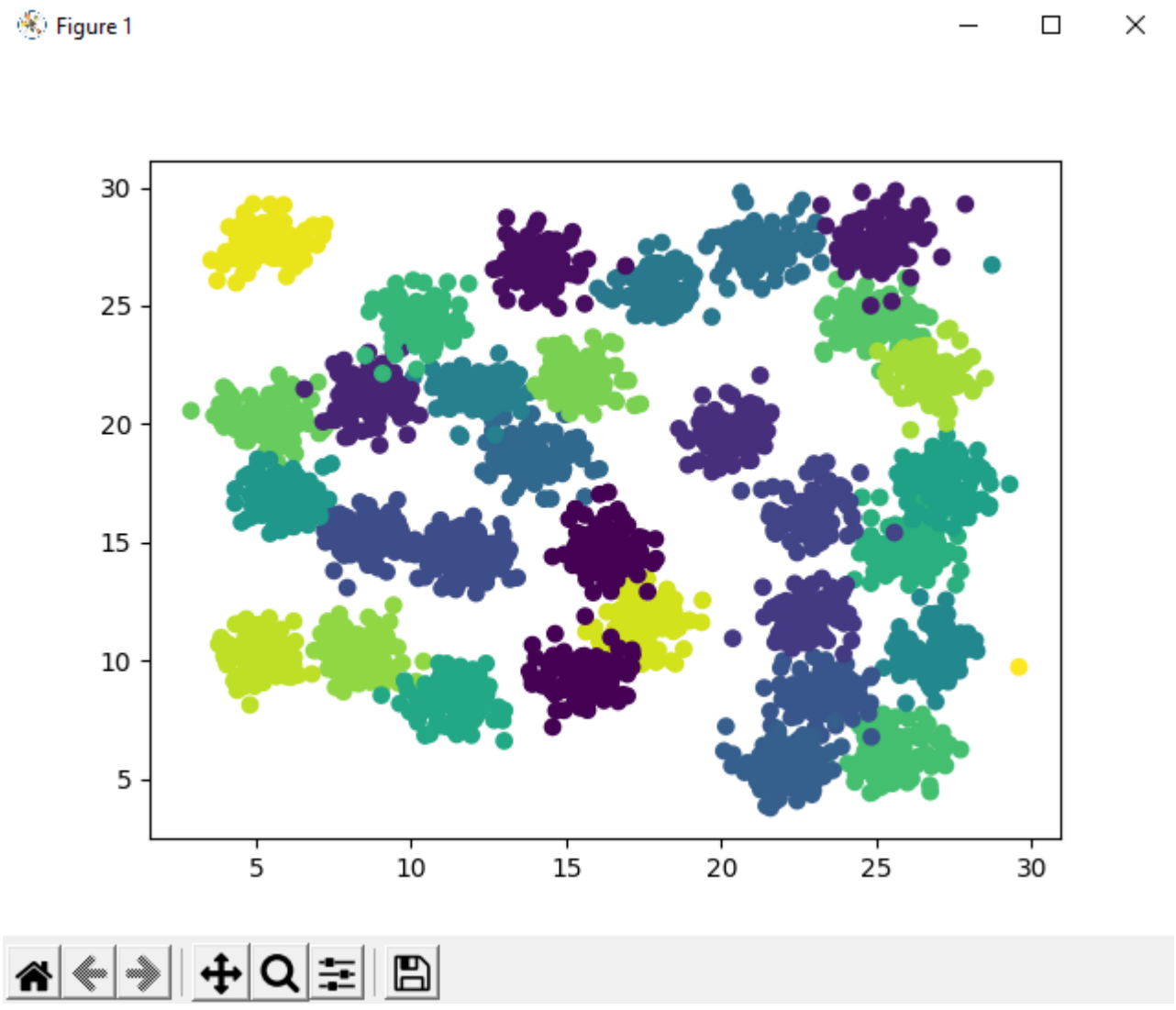
KMeans:

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

data = pd.read_csv('D31.csv')

X = np.array(data)
kmeans = KMeans(n_clusters=int(max(data['C'])), random_state = 0).fit(X)
plt.scatter(X[:,0], X[:,1],c=kmeans.labels_)
plt.show()
```



SingleLinkage:

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

data = pd.read_csv('D31.csv')

X = np.array(data)
clustering = AgglomerativeClustering(n_clusters=int(max(data['C'])),linkage =
'single').fit(X)
plt.scatter(X[:,0], X[:,1],c=clustering.labels_)
plt.show()
```

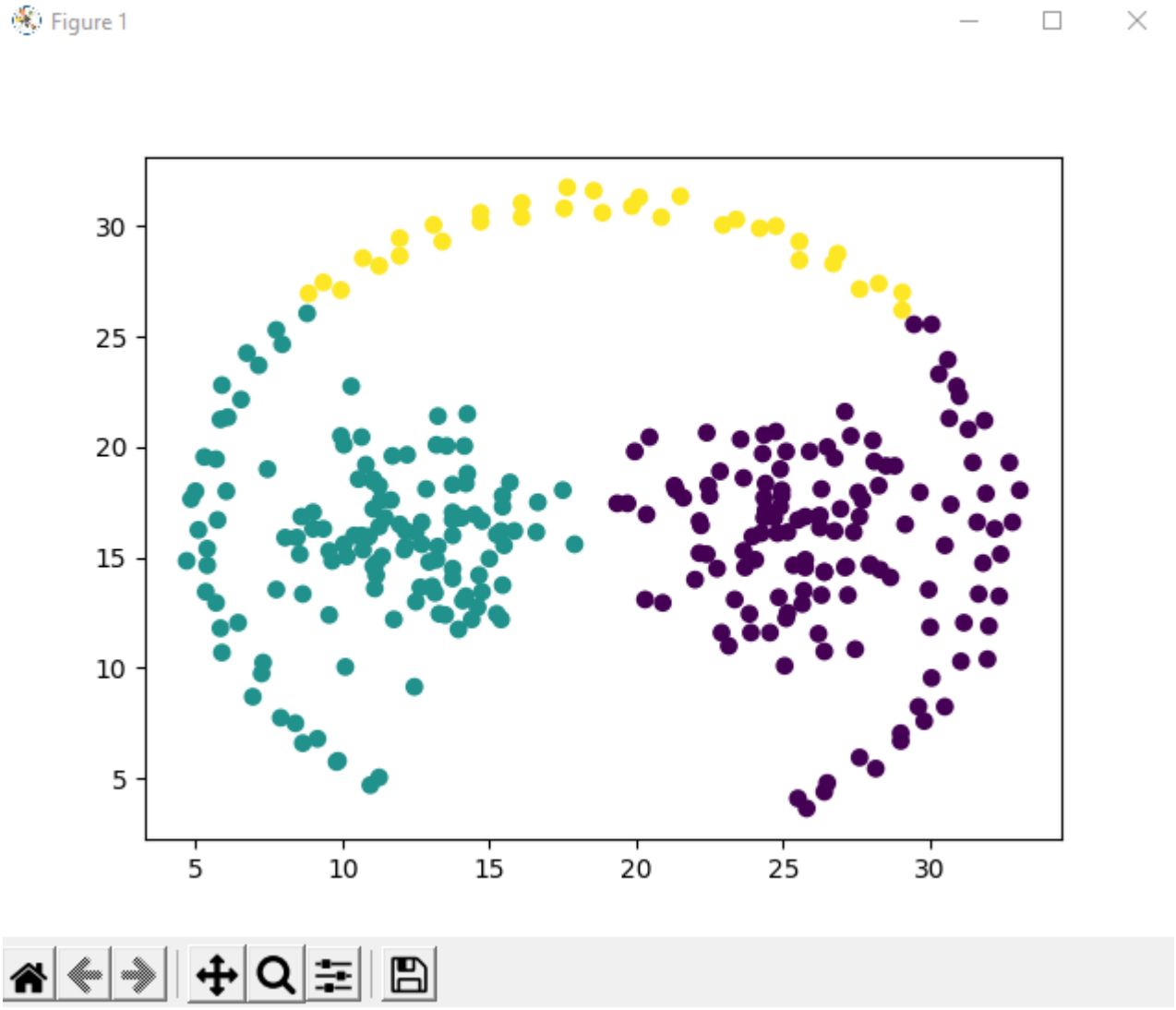


Nesse dataset se repetiu a situação do anterior, os grupos ficaram iguais, porém com partições diferentes, essas bem distribuídas.

Dataset: Pathbased

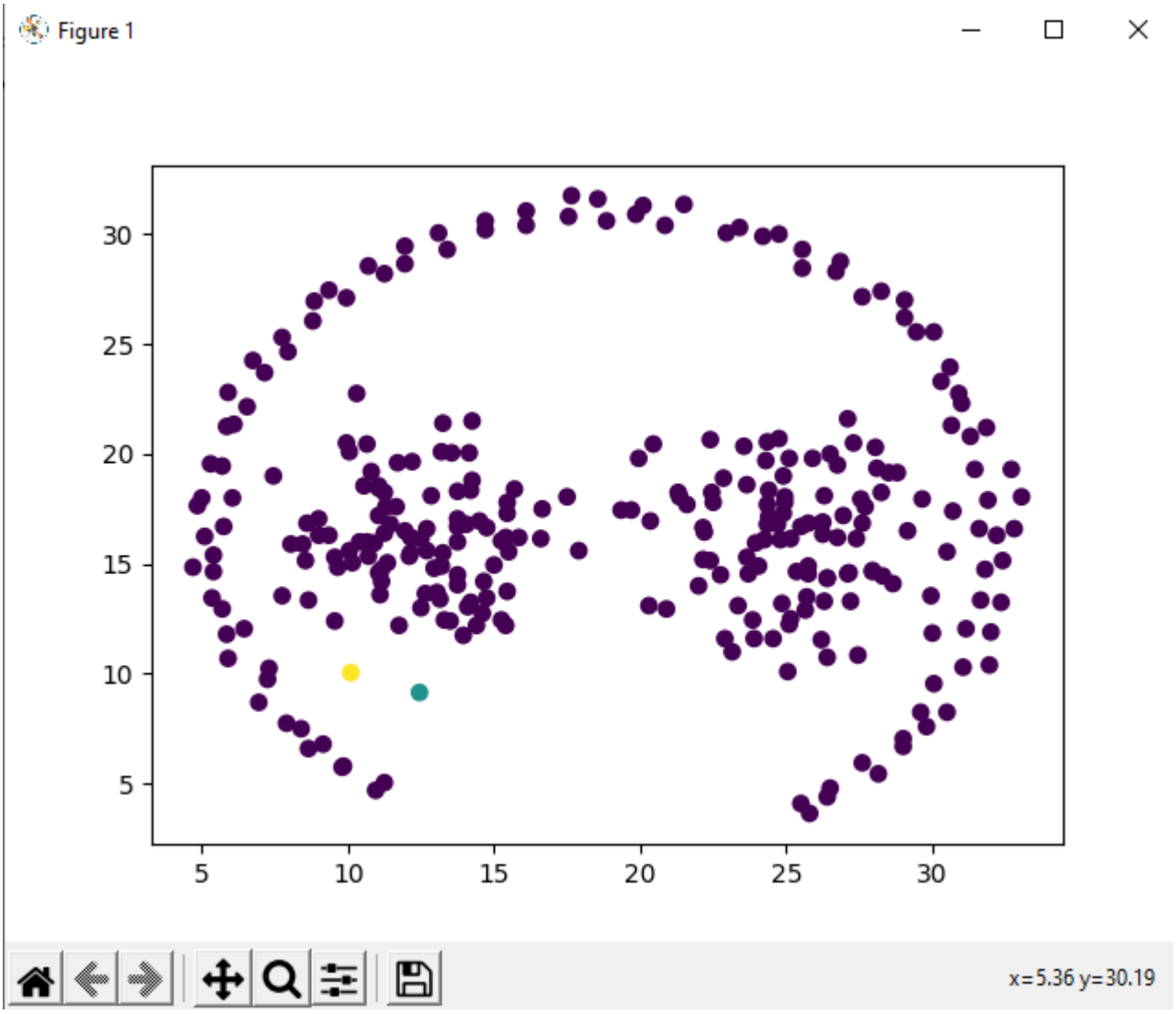
KMeans:

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

data = pd.read_csv('pathbased.csv')

X = np.array(data)
kmeans = KMeans(n_clusters=int(max(data['C'])), random_state = 0).fit(X)
plt.scatter(X[:,0], X[:,1],c=kmeans.labels_)
plt.show()
```



SingleLinkage:

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

data = pd.read_csv('pathbased.csv')

X = np.array(data)
clustering = AgglomerativeClustering(n_clusters=int(max(data['C'])),linkage =
'single').fit(X)
plt.scatter(X[:,0], X[:,1],c=clustering.labels_)
plt.show()
```

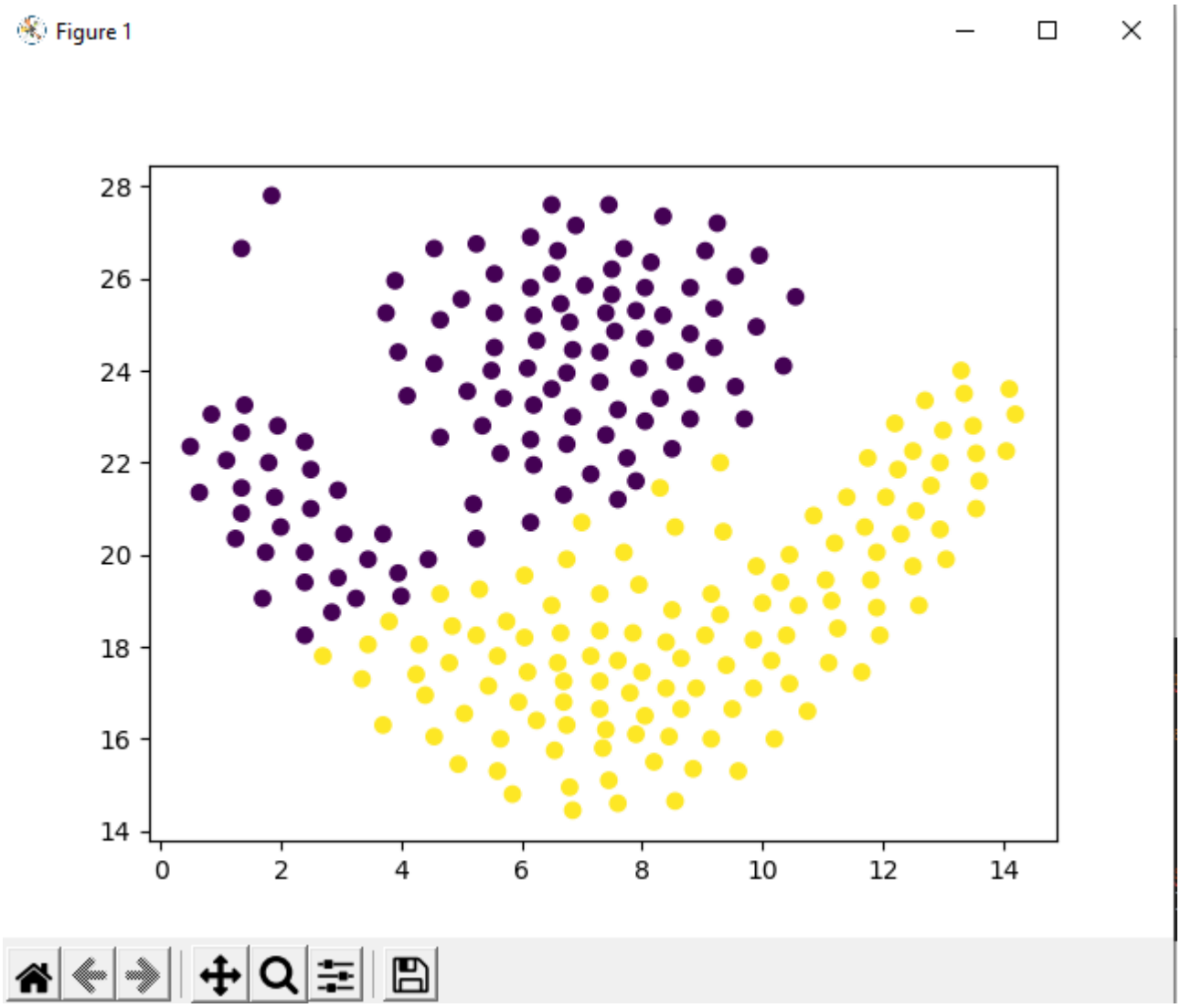


Nesse dataset, os grupos ficaram iguais e as partições diferentes, porém as partições do algoritmo SingleLinkage ficaram mal distribuídas com apenas 2 elementos únicos em suas partições e o restante em um outro. Portanto para esse dataset o algoritmo KMeans atende melhor.

Dataset: Flame

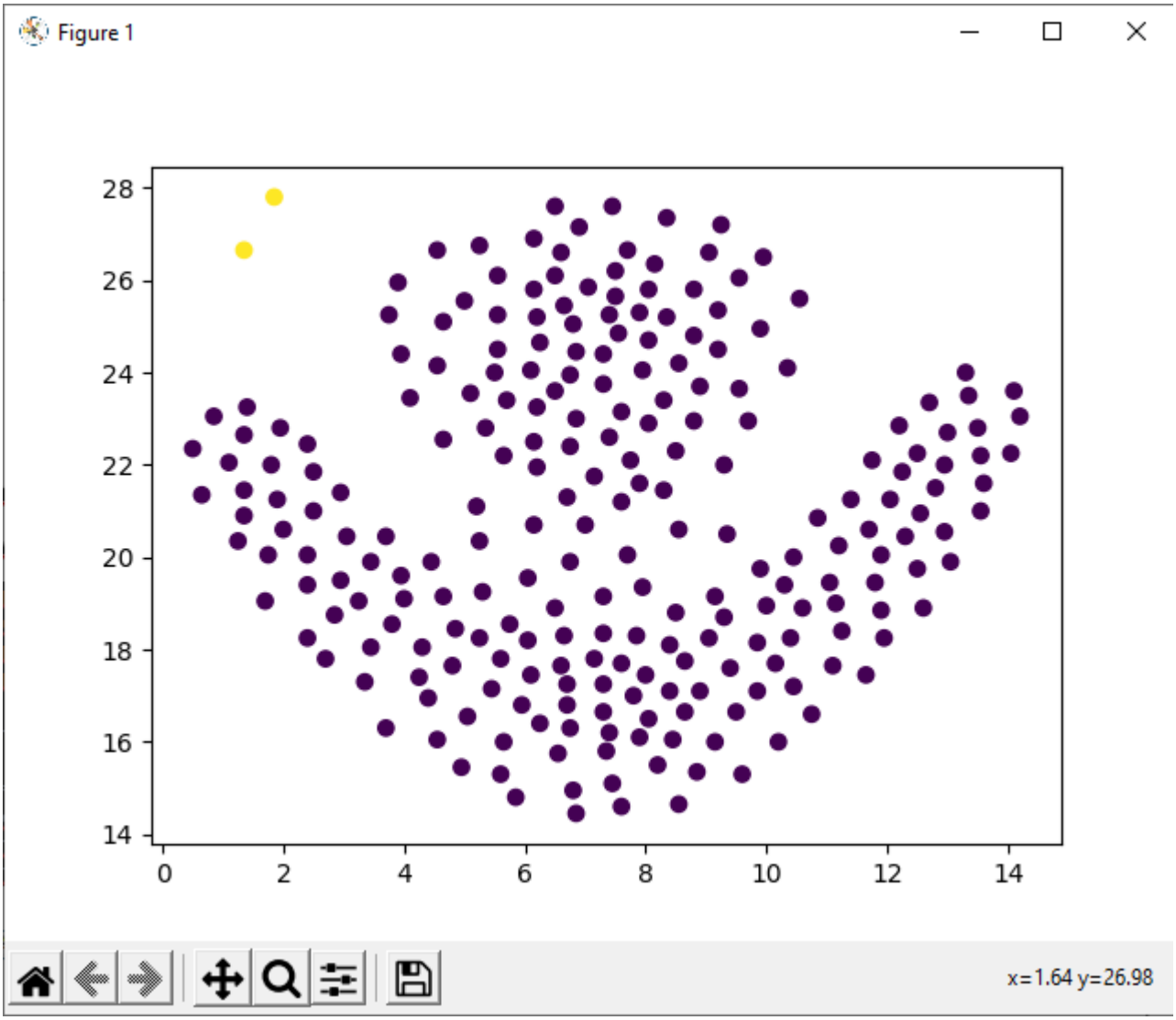
KMeans:

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

data = pd.read_csv('flame.csv')

X = np.array(data)
kmeans = KMeans(n_clusters=int(max(data['C'])), random_state = 0).fit(X)
plt.scatter(X[:,0], X[:,1],c=kmeans.labels_)
plt.show()
```



SingleLinkage:

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

data = pd.read_csv('flame.csv')

X = np.array(data)
clustering = AgglomerativeClustering(n_clusters=int(max(data['C'])),linkage =
'single').fit(X)
plt.scatter(X[:,0], X[:,1],c=clustering.labels_)
plt.show()
```



Nesse dataset a situação foi parecida com a do anterior, no algoritmo SingleLinkage as partições ficaram mal distribuídas com apenas dois elementos em uma partição e o restante em outra, então a melhor escolha na análise desse dataset é o algoritmo KMeans.

Fontes consultadas:
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html#sklearn.cluster.AgglomerativeClustering
https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html
https://stackoverflow.com/questions/28227340/kmeans-scatter-plot-plot-different-colors-per-cluster