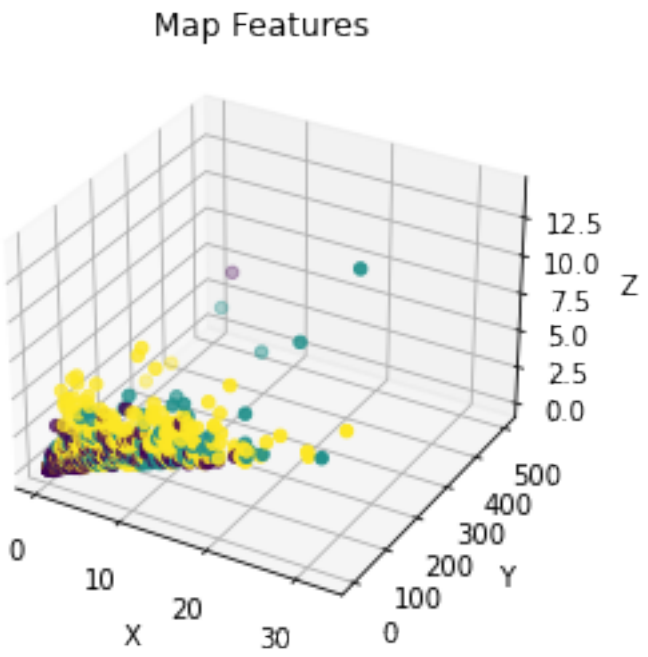
terrain_agrupamiento

November 14, 2021

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import SpectralClustering
from sklearn.cluster import OPTICS
from sklearn.cluster import DBSCAN
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.metrics import pairwise_distances

def plot_d(points, labels, title):
    fig = plt.figure()
    if points.shape[1] > 2:
        ax = fig.add_subplot(projection='3d')
        ax.scatter(points[:,0], points[:,1], points[:,2], c=labels,␣
 ↪cmap='viridis')
        ax.set_xlabel('X')
        ax.set_ylabel('Y')
        ax.set_zlabel('Z')
        ax.set_title(title)
    else:
        plt.scatter(points[:,0], points[:,1], c=labels, cmap='viridis')
        plt.xlabel('X')
        plt.ylabel('Y')
        plt.title(title)
    plt.show()
```

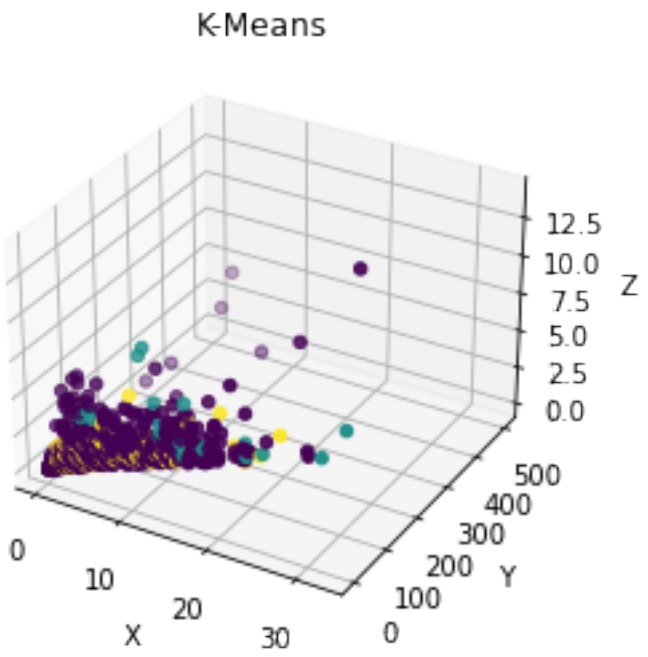
[ ]:
```
# Data
terrains = ('Plano', 'DM', 'Elevaciones')

x = np.load('processed_terrain_data.npy')
y = np.load('terrain_data_labels.npy')
```

[ ]:
```
plot_d(x, y, 'Map Features')
```



1 K-Means

[ ]:
```
kmeans = KMeans(n_clusters=len(terrains)).fit(x)
clustering_labels = kmeans.labels_
centers = kmeans.cluster_centers_

plot_d(x, clustering_labels, 'K-Means')
```

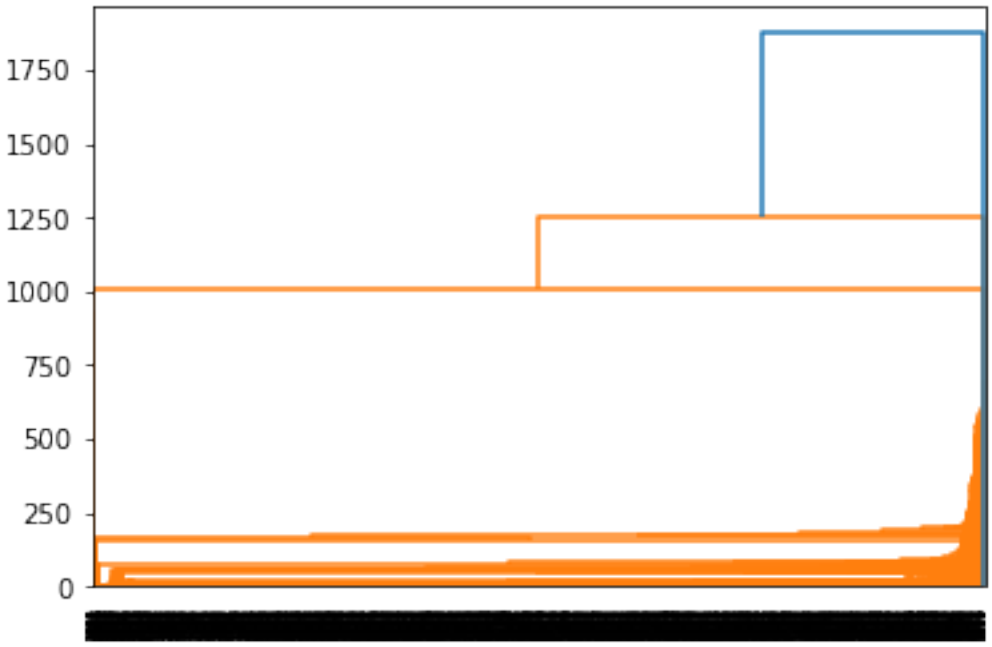


2 Dendogram

Sabe que rollo

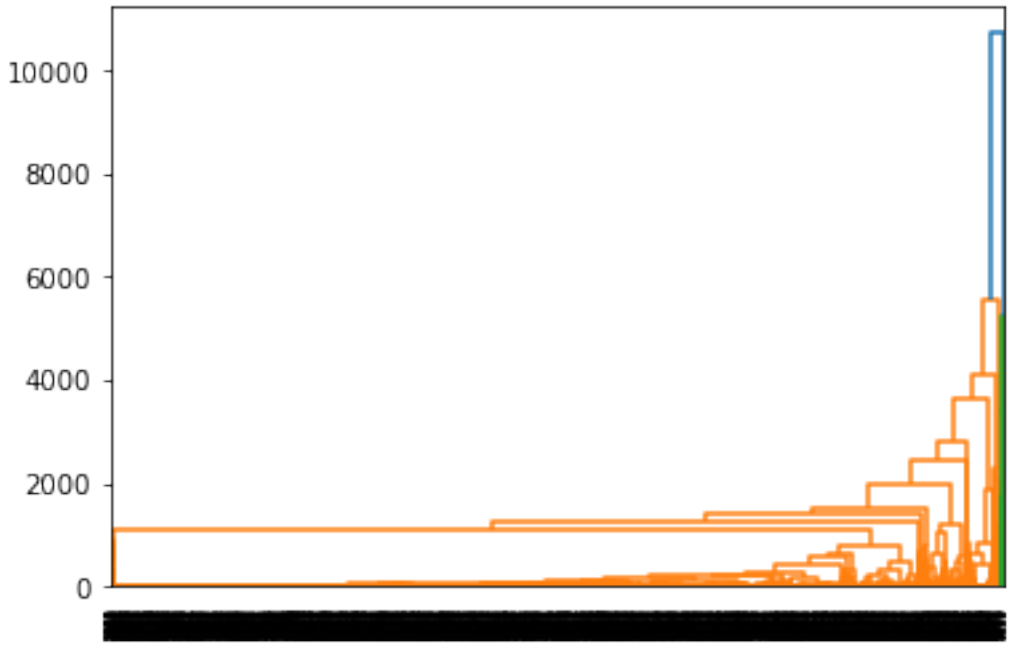
[ ]:
```
####### Dendrogram plot #######
print('----- Dendrogram plot -----')

linked = linkage(x, 'single')

plt.figure()
dendrogram(linked, orientation='top',distance_sort='descending',␣
 ↪show_leaf_counts=True)
plt.show()
```

```
----- Dendrogram plot -----
```



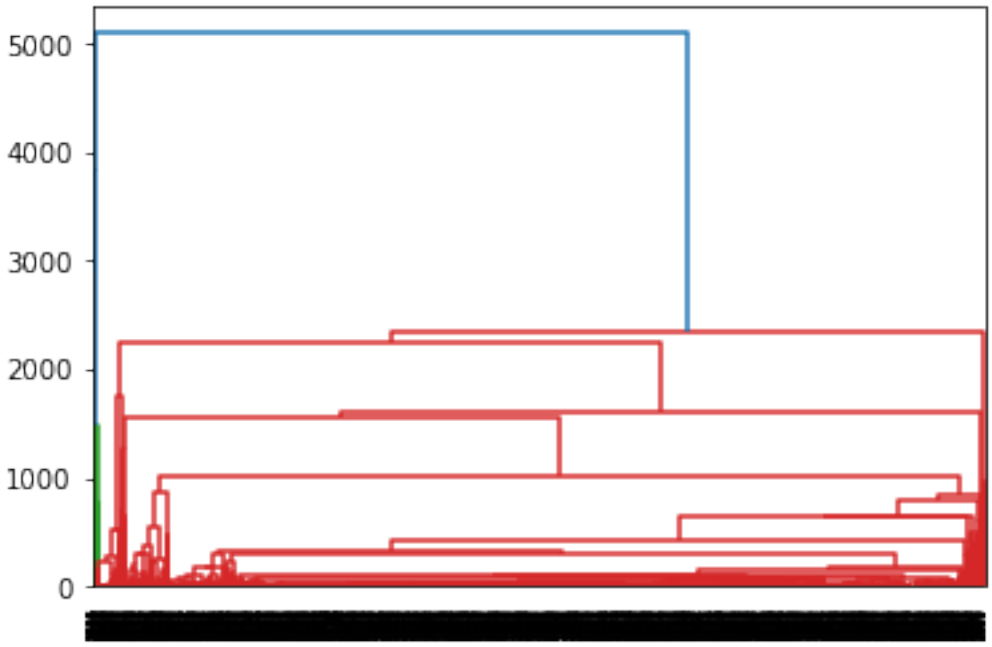
[ ]:
```
######## Dendrogram plot ########
print('----- Dendrogram plot -----')

linked = linkage(x, 'complete')

plt.figure()
dendrogram(linked, orientation='top',distance_sort='descending',␣
 ↪show_leaf_counts=True)
plt.show()
```

```
----- Dendrogram plot -----
```



[ ]:
```
####### Dendrogram plot #######
print('----- Dendrogram plot -----')

linked = linkage(x, 'average')

plt.figure()
dendrogram(linked, orientation='top',distance_sort='descending',␣
 ↪show_leaf_counts=True)
plt.show()
```

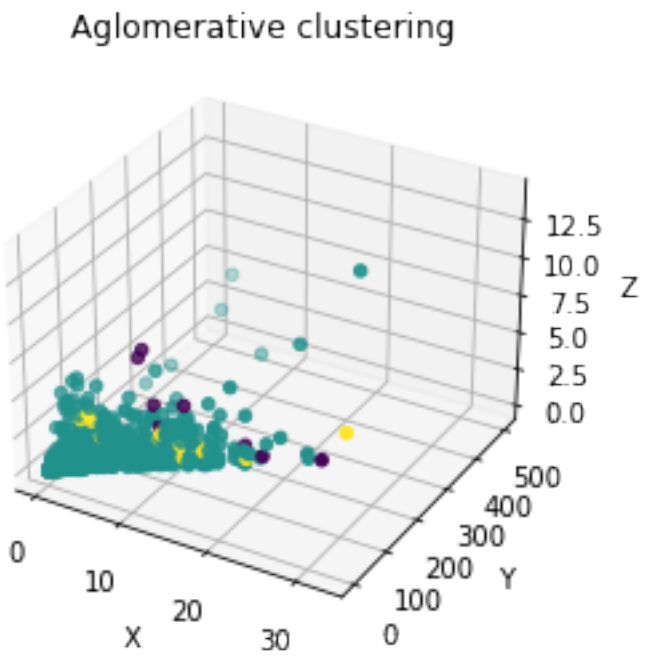
```
----- Dendrogram plot -----
```



3 Aglomerative

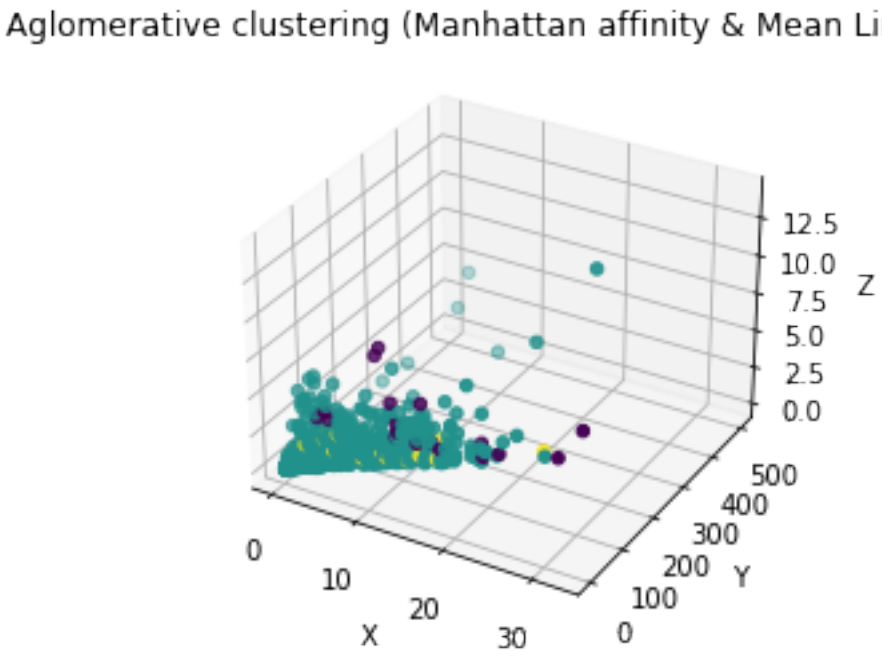
[ ]:
```
####### Aglomerative clustering #######
print('----- Aglomerative clustering -----')
agl = AgglomerativeClustering(n_clusters=len(terrains), linkage = 'complete').
 ↪fit(x)
clustering_labels = agl.labels_
print('Labels: ', clustering_labels)
plot_d(x, clustering_labels, 'Aglomerative clustering')
```

```
----- Aglomerative clustering -----
Labels:  [1 1 1 … 1 1 1]
```



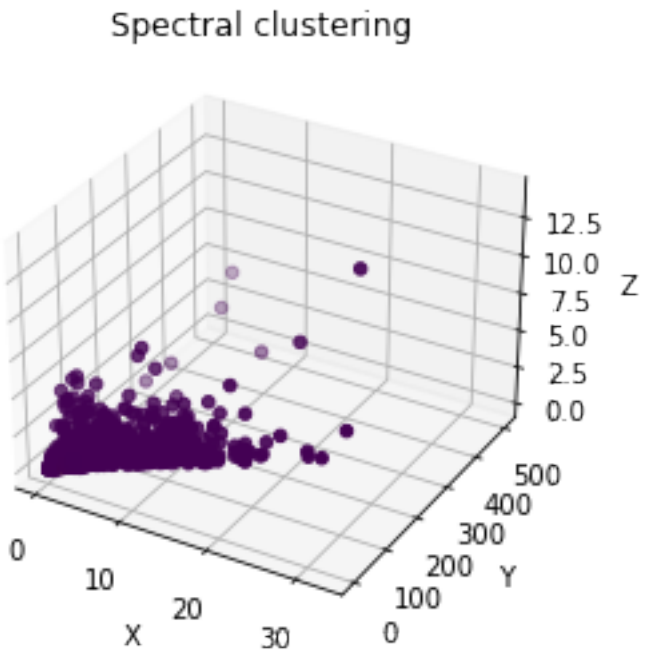
[ ]:
```
# Aglomerative clustering with manhattan distance and mean linkage
agl = AgglomerativeClustering(n_clusters=len(terrains), linkage = 'complete',␣
 ↪affinity = 'manhattan').fit(x)
clustering_labels = agl.labels_
print('Labels: ', clustering_labels)
plot_d(x, clustering_labels, 'Aglomerative clustering (Manhattan affinity &␣
 ↪Mean Linkage)')
```

```
Labels:  [1 1 1 … 1 1 1]
```



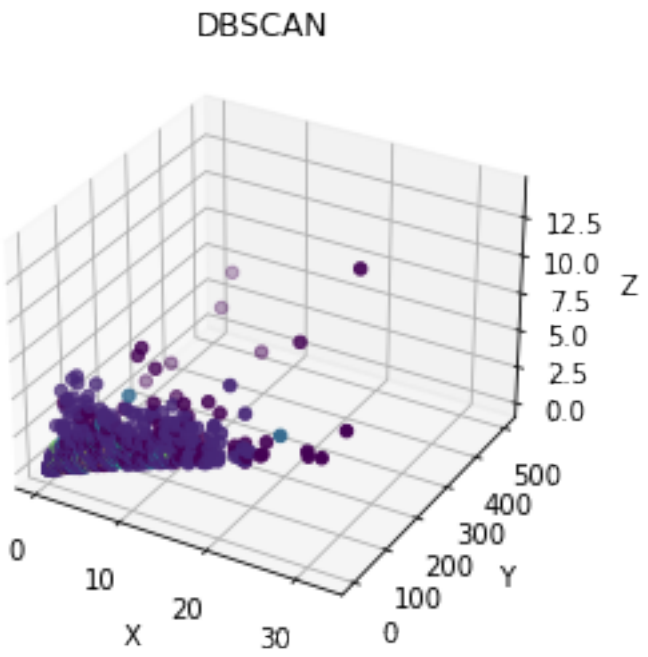
[ ]:
```
####### Spectral clustering #######
print('----- Spectral clustering -----')
spectral = SpectralClustering(n_clusters=len(terrains)).fit(x)
clustering_labels = spectral.labels_
print('Labels: ', clustering_labels)
plot_d(x, clustering_labels, 'Spectral clustering')
```

```
----- Spectral clustering -----

/home/hivini/anaconda3/envs/tf-gpu/lib/python3.9/site-
packages/sklearn/manifold/_spectral_embedding.py:236: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
  warnings.warn("Graph is not fully connected, spectral embedding"

Labels:  [0 0 0 … 0 0 0]
```



[ ]:
```
print('----- DBSCAN -----')
dbscan = DBSCAN(eps=20).fit(x)
clustering_labels = dbscan.labels_
print('Labels: ', clustering_labels)
plot_d(x, clustering_labels, 'DBSCAN')
```

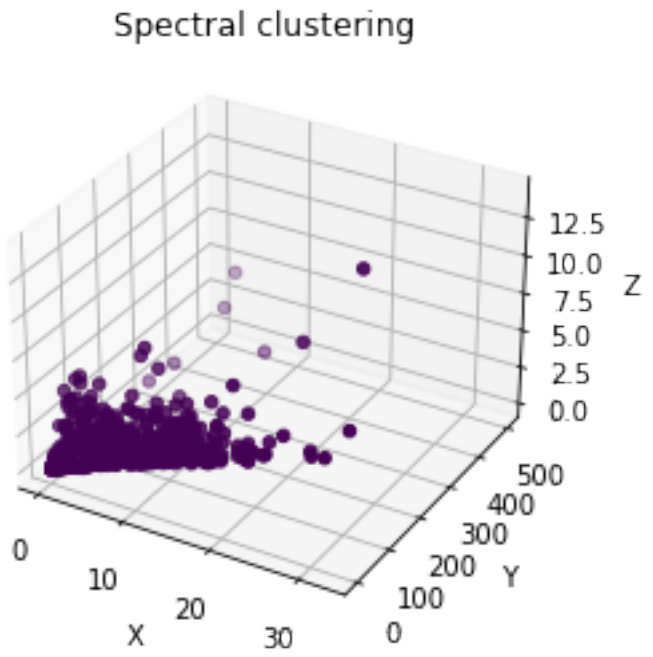
```
----- DBSCAN -----
Labels:  [0 0 0 … 0 8 0]
```



[ ]:
```
# Using custom affinity matrix
dist = pairwise_distances(x, metric = 'manhattan')
similarity = np.exp(- dist ** 2)

spectral = SpectralClustering(n_clusters=len(terrains), affinity =␣
 ↪'precomputed').fit(similarity)
clustering_labels = spectral.labels_
print('Labels: ', clustering_labels)
plot_d(x, clustering_labels, 'Spectral clustering')
```

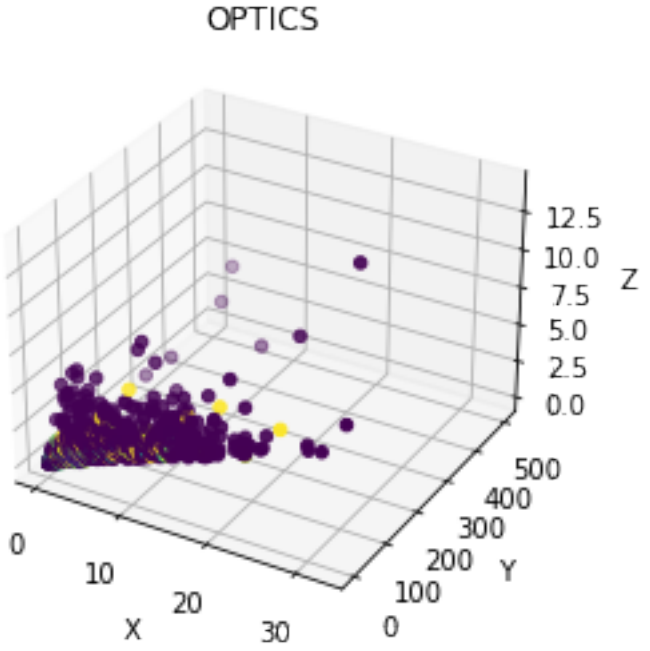
```
/home/hivini/anaconda3/envs/tf-gpu/lib/python3.9/site-
packages/sklearn/manifold/_spectral_embedding.py:236: UserWarning: Graph is not
fully connected, spectral embedding may not work as expected.
  warnings.warn("Graph is not fully connected, spectral embedding"
```

```
Labels:  [0 0 0 … 0 0 0]
```



[ ]:
```
####### OPTICS #######
print('----- OPTICS -----')
optics = OPTICS(min_samples = 20).fit(x)
clustering_labels = optics.labels_
print('Labels: ', clustering_labels)
plot_d(x, clustering_labels, 'OPTICS')
```

```
----- OPTICS -----
Labels:  [-1 -1  0 … -1 -1  4]
```


OPTICS
12.5
10.0
7.5
5.0
2.5
0.0
Z
500
400
300
200
100
0
Y
0
10
20
30
X