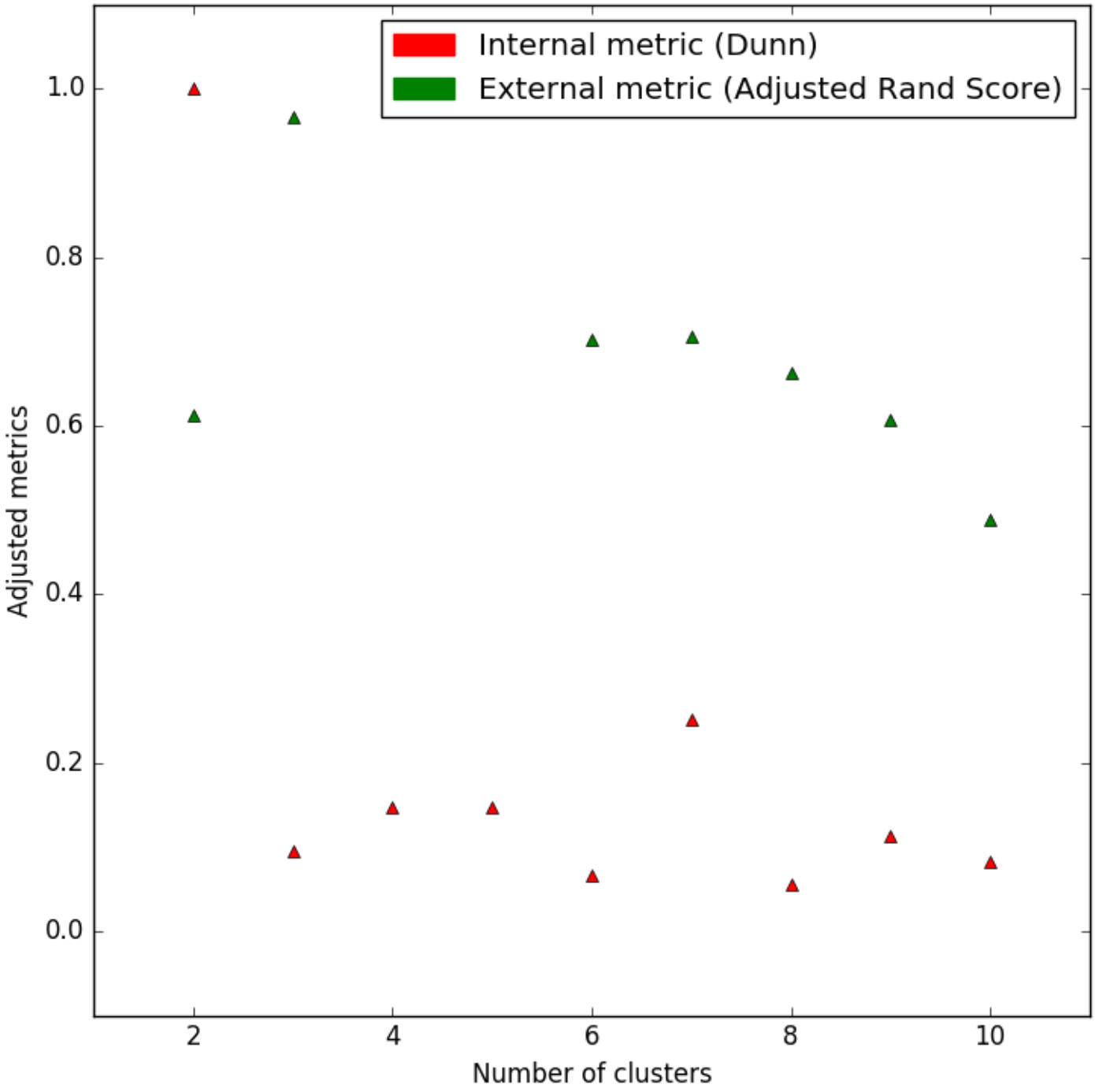
```
# Gabriel Hidasy Rezende, RA116928
# MC886 - Machine Learning, Exercicio 4

import numpy as np
from sklearn.cluster import KMeans
from sklearn import metrics
import matplotlib.pyplot as plt
from sklearn.metrics import silhouette_score
# Leitura dos dados
def readData():
    data = []
    with open("cluster-data.csv") as df:
        data = [x.split(",") for x in df.read().split("\n")[1:-1]]
    with open("cluster-data-class.csv") as f:
        labels = [x for x in f.read().split("\n")[1:-1]]
    return np.array(data, dtype="float"), np.array(labels, dtype="float")

data, labels = readData()

# Métrica interna - Implementei Dunn (não encontrei no scikit)
internal = []
for k in range(2, 11):
    # Faz a clusterização
    kmeans = KMeans(n_init=5, n_clusters=k, init="random", n_jobs=4)
    kmeans.fit(data)
    result = kmeans.predict(data)
    # Dunn é baseada no tamanho dos clusters, e na distancia entre
    # eles, para isso precisamos dos centroides e do diametro dos
    # clusters
    centroids = kmeans.cluster_centers_
    intra_dist = np.zeros((k,1),dtype='float')
    # Achar o elemento mais distante do centro do cluster para cada
    # cluster
    for y in range(0,len(result)):
        tmp = (((data[y][0]-centroids[result[y]][0])**2 +
                (data[y][1]-centroids[result[y]][1])**2)**(1/2))
        if(tmp>intra_dist[result[y]]):
            intra_dist[result[y]] = tmp
    max_intra_dist = max(intra_dist)
    min_centroids_dist = 1000000
    for y in centroids:
        for z in centroids:
            if(z[0]==y[0] and z[1]==y[1]): # Mesmo centroide
                continue
            # Distancia euclidiana entre eles, raiz pode ser removida
            tmp = (((y[0]-z[0])**2)+((y[1]-z[1])**2))**(1/2)
            if(tmp<min_centroids_dist):
                min_centroids_dist = tmp
    dunn = min_centroids_dist/max_intra_dist
    internal.append((k, dunn[0]))

#No caso do DUNN quanto maior melhor
internal.sort(key=lambda x: x[1], reverse=True)
print("Best K according to dunn is", internal[0][0])

# Normalmente 2

# Outra métrica interna
internal2 = []
for k in range(2, 11):
    kmeans = KMeans(n_init=5, n_clusters=k, n_jobs=4)
    kmeans.fit(data)
```

```
        predicted = kmeans.predict(data)
        quality = silhouette_score(data, predicted)
        internal2.append((k, quality))

    internal2.sort(key=lambda x: x[1], reverse=True)
    print("Best k according to the silhouette score is", internal2[0][0])

    # Normalmente 2

    # Para a métrica externa usei o adjusted_rand_score, que é parte do
    # scikit.
    external = []
    for k in range(2, 11):
        kmeans = KMeans(n_init=5, n_clusters=k, n_jobs=4)
        kmeans.fit(data)
        predicted = kmeans.predict(data)
        quality = metrics.adjusted_rand_score(labels, predicted)
        external.append((k, quality))

    external.sort(key=lambda x: x[1], reverse=True)
    print("Best k according to the adjusted rand score is", external[0][0])

    # Normalmente 4-5

    # Para o plot, os resultados foram normalizados entre 0 e 1
    max_external = max(external, key=lambda x: x[1])[1]
    max_internal = max(internal, key=lambda x: x[1])[1]
    max_internal2 = max(internal2, key=lambda x: x[1])[1]

    for el in external:
        plt.plot(el[0], el[1]/max_external, 'g^')

    for el in internal:
        plt.plot(el[0], el[1]/max_internal, 'r^')

    for el in internal2:
        plt.plot(el[0], el[1]/max_internal2, 'b^')


    plt.ylabel('Adjusted metrics')
    plt.xlabel('Number of clusters')

    plt.ylim((-0.1, 1.1))
    plt.xlim((1, 11))

    import matplotlib.patches as mpatches

    red_patch = mpatches.Patch(color='red', label='Internal metric (Dunn)')
    blue_patch = mpatches.Patch(color='blue', label='Internal metric (Silhouette)')
    green_patch = mpatches.Patch(color='green',
                                 label='External metric (Adjusted Rand Score)')
    plt.legend(handles=[red_patch, green_patch, blue_patch])
    plt.show()

    # O melhor k segundo as métricas internas (Dunn e silhouette) é 2, segundo a
    # métrica externa (Adjusted Rand Score) é 4-5
```


Internal metric (Dunn)
External metric (Adjusted Rand Score)
Adjusted metrics
1.0
0.8
0.6
0.4
0.2
0.0
2
4
6
8
10
Number of clusters