

## A priedas. Apleistų teritorijų automatizuotos klasterizacijos pagal apibrėžtus rodiklius algoritmo *Python* programinis kodas

```
1 # Hierarchical Clusterin
2 # Python 3.5.2
3
4 # Importing the libraries
5 import numpy as np
6 import matplotlib.pyplot as plt
7 import pandas as pd
8
9 # Importing the mall dataset with pandas
10 dataset = pd.read_csv('BF_data.csv')
11 X = dataset.iloc[:, [2, 3]].values
12
13 # Using the dendrogram to find the optimal number of clusters
14 import scipy.cluster.hierarchy as sch
15 dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
16 plt.title('Dendrogram')
17 plt.xlabel('Residents around BF')
18 plt.ylabel('Distance from city center')
19 plt.show()
20
21 # Fitting hierarchical clustering to the mall dataset
22 from sklearn.cluster import AgglomerativeClustering
23 hc = AgglomerativeClustering(n_clusters = 3, affinity = 'euclidean', linkage = 'ward')
24 y_hc = hc.fit_predict(X)
25
26 # Visualize the clusters
27 # Visualising the clusters
28 plt.scatter(X[y_hc == 0, 0], X[y_hc == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
29 plt.scatter(X[y_hc == 1, 0], X[y_hc == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
30 plt.scatter(X[y_hc == 2, 0], X[y_hc == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
31 plt.title('Clusters of Urban Brownfields')
32 plt.xlabel('Number of Residents interacting with BF')
33 plt.ylabel('Distance from city center (m)')
34 plt.legend()
35 plt.show()
36
37 # Prepared by Vytautas Bielinskas
```

### A.1. pav. Hierarchinio klasterizavimo programinis kodas *Python* programavimo kalba

Fig. A.1. Hierarchical clustering source code in *Python* programming language