Лабораторная работа 5

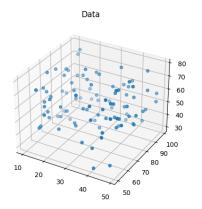
В работе использовалось окружение: Python 3.9, matplotlib 3.8.0, pandas 2.2.2, scikit-learn 1.4.2.

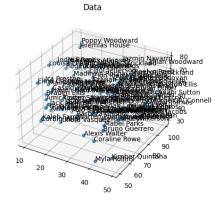
Цель работы: попытаться кластеризовать тестовые данные (все значения случайны).

1. Визуализация тестовых данных

```
import pandas
import matplotlib.pyplot as plt
ax3d = plt.figure().add_subplot(projection = '3d')
points = pandas.read_csv('data.csv', encoding="ISO-8859-1")
ages = points['Age'].values
heights = points['Height'].values
weights = points['Weight'].values
names = points['Name'].values
ages = ages.flatten()
heights = heights.flatten()
weights = weights.flatten()
ax3d.scatter(ages, heights, weights)
plt.title("Data")
plt.savefig('data_no_names.png')
for x, y, z, label in zip(ages, heights, weights, names):
    ax3d.text(x, y, z, label)
plt.savefig('data.png')
```

Лабораторная работа 5





2. Кластеризация методами к средних и иерархическим методом

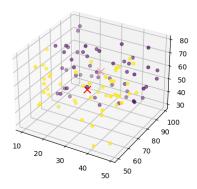
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans, AgglomerativeClustering
def plot_clusters_3d(ax, data, labels, centers=None):
    ax.scatter(data[:, 0], data[:, 1], data[:, 2], c=labels,
    if centers is not None:
        ax.scatter(centers[:, 0], centers[:, 1], centers[:, 2
def kmeans_clustering(data, k):
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(data)
    labels = kmeans.labels_
    centers = kmeans.cluster centers
    return labels, centers
def hierarchical_clustering(data, k):
    hierarchical = AgglomerativeClustering(n_clusters=k)
    labels = hierarchical.fit_predict(data)
    return labels
def run_kmeans_clustering(data, k_values):
```

Лабораторная работа 5

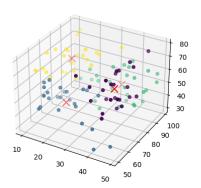
```
fig, axs = plt.subplots(2, 2, figsize=(12, 10), subplot_k
    axs = axs.flatten()
    for i, k in enumerate(k_values):
        labels, centers = kmeans_clustering(data, k)
        axs[i].set_title(f'KMeans Clustering (k={k})')
        plot_clusters_3d(axs[i], data, labels, centers)
    plt.savefig('k-means.png')
def run_hierarchical_clustering(data, k_values):
    fig, axs = plt.subplots(2, 2, figsize=(12, 10), subplot_k
    axs = axs.flatten()
    for i, k in enumerate(k values):
        labels = hierarchical_clustering(data, k)
        axs[i].set_title(f'Hierarchical Clustering (k={k})')
        plot clusters 3d(axs[i], data, labels)
    plt.savefig('hierarchical-method.png')
if __name__ == "__main__":
    data = pd.read csv('data.csv')
    selected_data = data[['Age', 'Height', 'Weight']].values
    k_{values} = [2, 3, 4, 5]
    run_kmeans_clustering(selected_data, k_values)
    run_hierarchical_clustering(selected_data, k_values)
```

Лабораторная работа 5 3

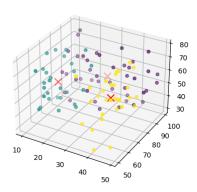
KMeans Clustering (k=2)



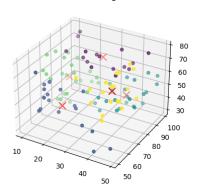
KMeans Clustering (k=4)



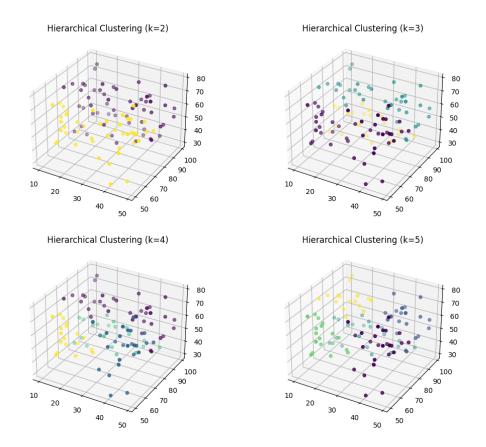
KMeans Clustering (k=3)



KMeans Clustering (k=5)



Лабораторная работа 5



Как мы видим, методы показали одинаковые результаты при количестве кластеров равном 2, однако на больших размерностях методы проявили себя по-разному.

Лабораторная работа 5 5