СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

КУРСОВ ПРОЕКТ
ПО СИСТЕМИ, ОСНОВАНИ НА ЗНАНИЯ

Тема:

Сегментиране на клиенти

Студенти:

Александър Александров Ангелков, II група, 71995
Георги Ивайлов Цеков, II група, 72039
Михаела Янкова Ангелова, I група, 72030

София, януари 2023 г.

1. Формулировка на задачата.

Сегментиране на клиентите е практиката на разделяне на една клиентска база в групи (клъстери) от лица, които имат сходни демографски или други характеристики. Напишете програма, която реализира метод за сегментиране на дадено множество от клиенти.

2. Използвани алгоритми.

- Elbow method – търси оптимално k, където k е броят на клъстерите. Оптималмният брой клъстери се определя, като се намери форма на “elbow” в графиката, от която WCSS (Within-Cluster Sum of Squares) започват да намаляват по-бавно;
- Silhouette method – търси оптимално k (брой на клъстерите), като търси максималния silhouette коефициент. Неговите стойности са в интервала [-1,1]. Графиката на метода показва близостта на точка от единия клъстер до точки в съседните клъстери. По този начин представя визуална оценка на параметри като брой клъстери;
- K-Means - стреми се да раздели обектите на k (k е дадено естествено число) клъстера по следния начин:
    - по случаен начин се избират центровете на всички клъстери;
    - повтарят се следните стъпки:
        - всеки обект се асоциира с (причислява към) клъстера с най-близък център;
        - замества се всеки център на клъстер със средното на всички обекти, асоциирани с него.

3. Описание на програмната реализация.

Структура:
1. Преобразуваме категорийните променливи към колони от булеви стойности;
2. Проверяваме дали данните са нормално разпределени по колони чрез тест на Шапиро (от библиотеката scipy);
2.1. Ако данните за всяка колона са нормално разпределени (p-value >= 0,05), прилагаме метода Z-score за откриване на потенциални outlier-и;
2.2. Ако данните за всяка колона не са нормано разпределени (p-value < 0.05), използваме метода IQR за откриване на потенциални outlier-и;
3. Записваме данните без outlier-и в нов файл;
4. Нормализираме данните по колони в интервала [0,1];

5. Прилагаме Elbow метода за намиране на оптимален брой k клъстери за данните със и без outlier-и за всички колони, за да покажем, че outlier-ите влияят на стойностите от графиката;
6. Прилагаме K-Means алгоритъма за оптимално k върху данните със и без outlier-и;
7. Прилагаме Silhouette метода за намиране на оптимално k за данните със и без outlier-и за две колони и показваме близките, но различни резултати от метода, с което доказваме, че outlier-ите влияят отрицателно на данните;
8. Прилагаме Elbow метода за две колони от данните със и без outlier-и;
9. Сравняваме резултатите от двата метода за оптимален k брой клъстери;
10. Прилагаме K-Means алгоритъма за данните със и без outlier-и за две колони;
11. Анализираме получените резултати по клъстери.

Компоненти:
1. Функция elbow, която прилага Elbow метода;
2. Функция graphics_for_df_and_new_df, която визуализира графиките на данните със и без outlier-и;
3. Функция silhouette, която прилага Silhouette метода;
4. Функция visualization, която визуализира графиката на образувалите се клъстери от данните.

4. Примери, илюстриращи работата на програмната система.

```
def elbow(scaled_df):
    wcss=[]
    for i in range(1,20):
        km = KMeans(i)
        km.fit(scaled_df)
        wcss.append(km.inertia_)
    np.array(wcss)
    print('_____')
    print('Array of predicted clusters in which every element belongs:')
    return wcss
```

```
def graphics_for_df_and_new_df(title, wcss, elbow_point, elbow_point_no):
    fig, ax = plt.subplots(figsize=(10,6))
    ax = plt.plot(range(1,20), wcss, linewidth=2, color="red", marker ="8")
    ax_no = plt.plot(range(1,20), wcss_no, linewidth=2, color="blue", marker ="8")
    plt.axvline(x=elbow_point, ls='-', color="red")
    plt.axvline(x=elbow_point_no, ls=':', color="blue")
    plt.ylabel('Within-Cluster Sum of Squares')
    plt.xlabel('Number of Clusters')
    plt.xticks(np.arange(0, 20, 1)) #generates an array of numbers from 0 to 9 with a step size of 1
    plt.title(title, fontsize = 18)
    plt.show()
```

```
def silhouette(X, title):
    silhouette_average = []
    for k in range(2,20):
        kmeans = KMeans(n_clusters = k).fit(X)
        labels = kmeans.labels_
        silhouette_average.append(silhouette_score(X, labels, metric = 'euclidean'))
    optimal_n_clusters = range(2,20)[np.argmax(silhouette_average)]
    print('Optimal K depending on the silhouette score: ' + str(optimal_n_clusters))
    plt.plot(range(2,20),silhouette_average,'bx-')
    plt.xlabel('Number of clusters')
    plt.ylabel('Silhouette score')
    plt.title(title)
    plt.show()
```

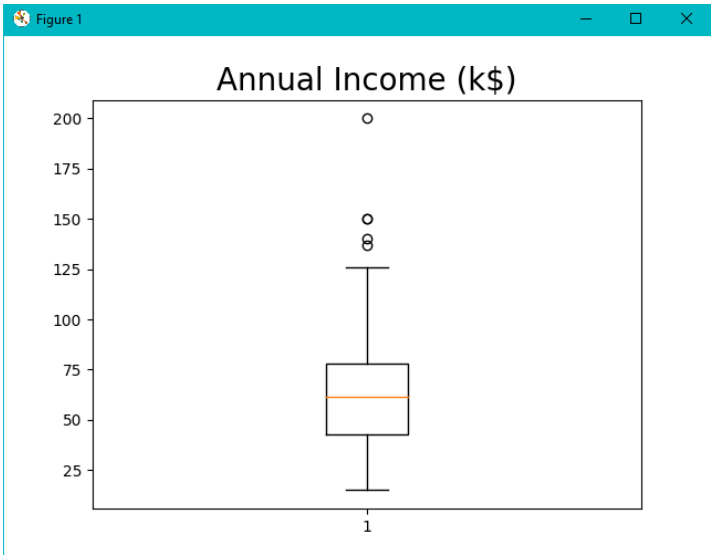
```
def visualization(new_df, title, km_no):
    plt.figure(figsize=(10,7))
    plt.scatter(new_df['Annual Income (k$)'], new_df['Spending Score (1-100)'],  c=km_no.labels_ , cmap='rainbow')
    plt.xlabel('Annual Income (k$)')
    plt.ylabel('Spending Score (1-100)')
    plt.title(title)
    plt.show()
```

```
if shapiro_test.pvalue < 0.05: #find outliers from a data that is normally distributed
    print('The data from ' + col + ' is not normally distributed,' +
    ' that\'s why we should use the IQR (Interquartile Range) method for identifying the outliers.')
    Q1 = new_df[col].quantile(0.25)
    Q3 = new_df[col].quantile(0.75)
    IQR = Q3 - Q1
    print('IQR of ' + col + ': ' + str(IQR))
    for i in range(new_df[col].size):
        value = new_df[col][i]
        if value < (Q1 - 1.5 * IQR) or value > (Q3 + 1.5 * IQR): #pattern for checking for outliers
            print('Outlier value: ' + str(value) + ' in ' + col)
            new_df.loc[i, col] = 'Outlier'
            new_df.to_csv(new_data, index=False)
```

```
else: #find outliers from a data that is not normally distributed
    print('The data from ' + col + ' is normally distributed,' +
    ' that\'s why we should use the Z-score method for identifying the outliers.')
    mean = np.mean(col)
    std = np.std(col)
    print('The mean of the dataset is ', mean)
    print('The standart deviation is ', std)
    for i in range(new_df[col].size):
        value = new_df[col][i]
        z = (value - mean) / std
        if z > 3: #pattern for checking for outliers
            print('Outlier value: ' + str(value) + ' in ' + col)
            new_df.loc[i, col] = 'Outlier'
            new_df.to_csv(new_data, index=False)
```

```python
#ELBOW METHOD WITH OUTLIERS
wcss = elbow(scaled_df)
kn = KneeLocator(range(1,20), wcss, curve='convex', direction='decreasing')
elbow_point = kn.knee
km = KMeans(n_clusters = elbow_point)
y_predicted = km.fit_predict(df[['Age','Annual Income (k$)', 'Spending Score (1-100)', 'isMale', 'isFemale']])
print(y_predicted)

#ELBOW METHOD WITHOUT OUTLIERS
wcss_no = elbow(scaled_new_df)
kn_no = KneeLocator(range(1,20), wcss_no, curve='convex', direction='decreasing')
elbow_point_no = kn_no.knee
km_no = KMeans(n_clusters = elbow_point_no)
y_predicted_no = km_no.fit_predict(new_df[['Age','Annual Income (k$)', 'Spending Score (1-100)', 'isMale', 'isFemale']
print(y_predicted_no)
```

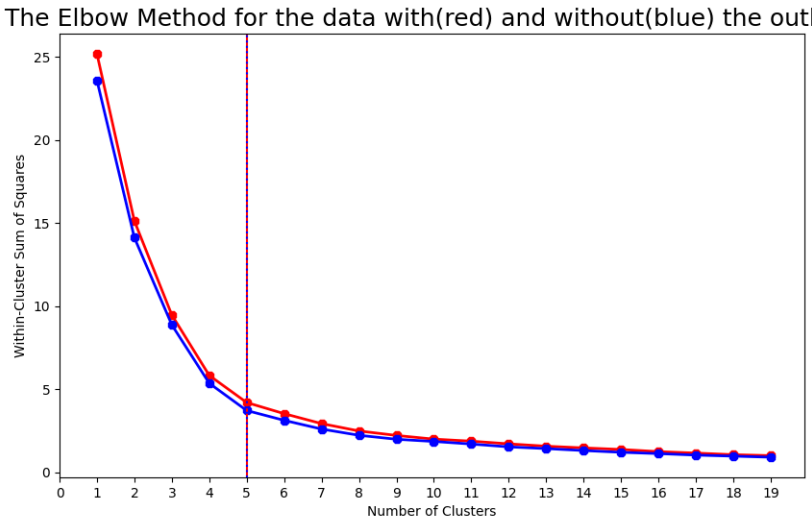
```python
#SILHOUETTE METHOD WITH OUTLIERS
print('_____')
print('Silhouette method WITH OUTLIERS:')
silhouette(df[['Annual Income (k$)', 'Spending Score (1-100)']], 'Silhouette method WITH OUTLIERS')

#SILHOUETTE METHOD WITHOUT OUTLIERS
print('Silhouette method WITHOUT OUTLIERS:')
silhouette(new_df[['Annual Income (k$)', 'Spending Score (1-100)']], 'Silhouette method WITHOUT OUTLIERS')
```



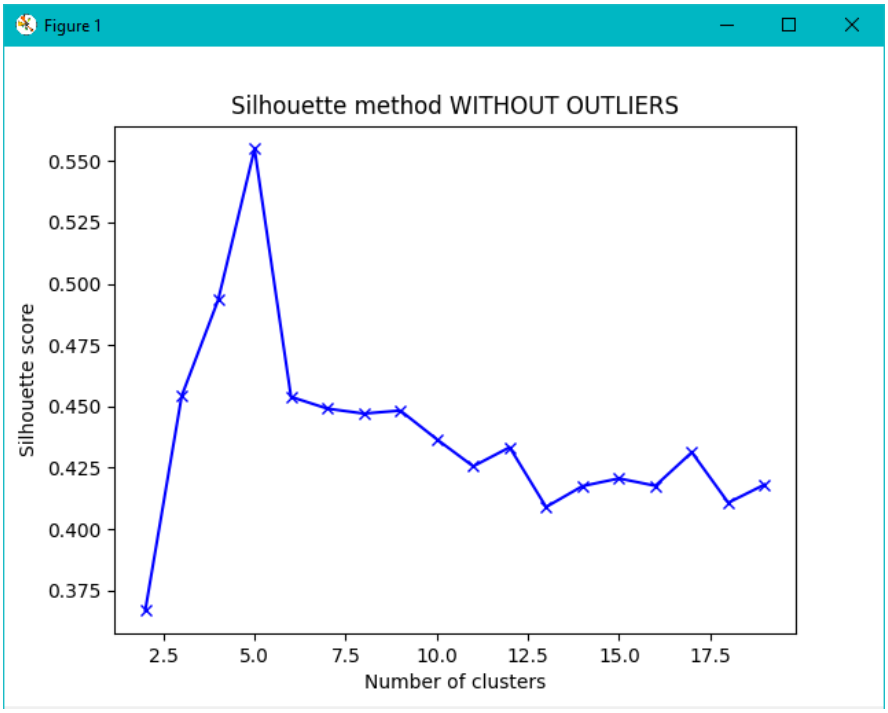
примерен box-plot, който показва outlier-ите на данните от Annual Income, минималната и максималната стойност, Q1 и Q3 (съответно първи и трети квартил) и средната стойност

```
_____
Array of predicted clusters in which every element belongs:
[4 3 4 3 0 4 4 3 4 3 4 3 3 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4
 3 4 3 0 3 0 3 4 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 2 1 2 1 0 1 2 1 2 1 2 1 2 1 0 1 2 1 0 1
 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1]
```

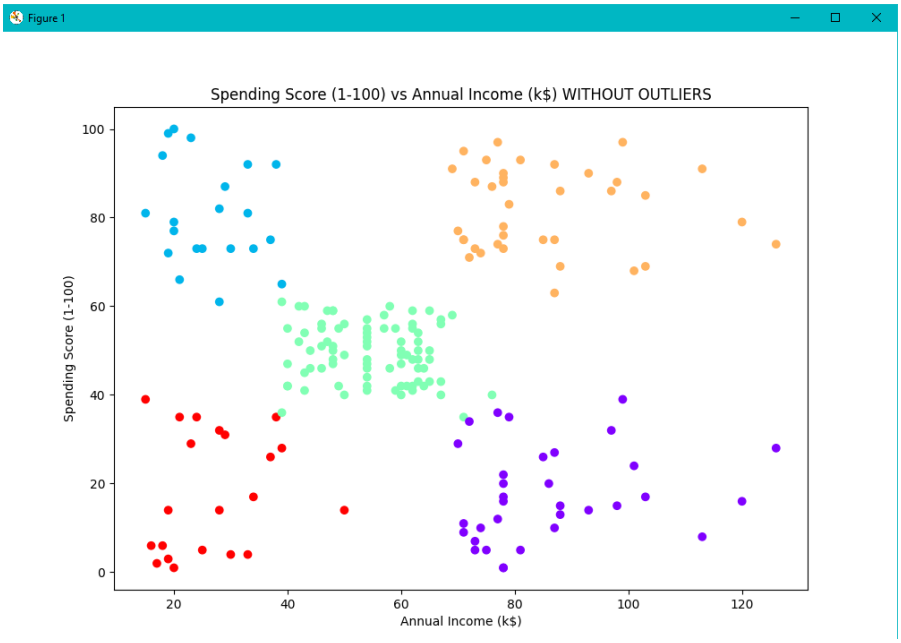
Резултат след прилагане на K-Means



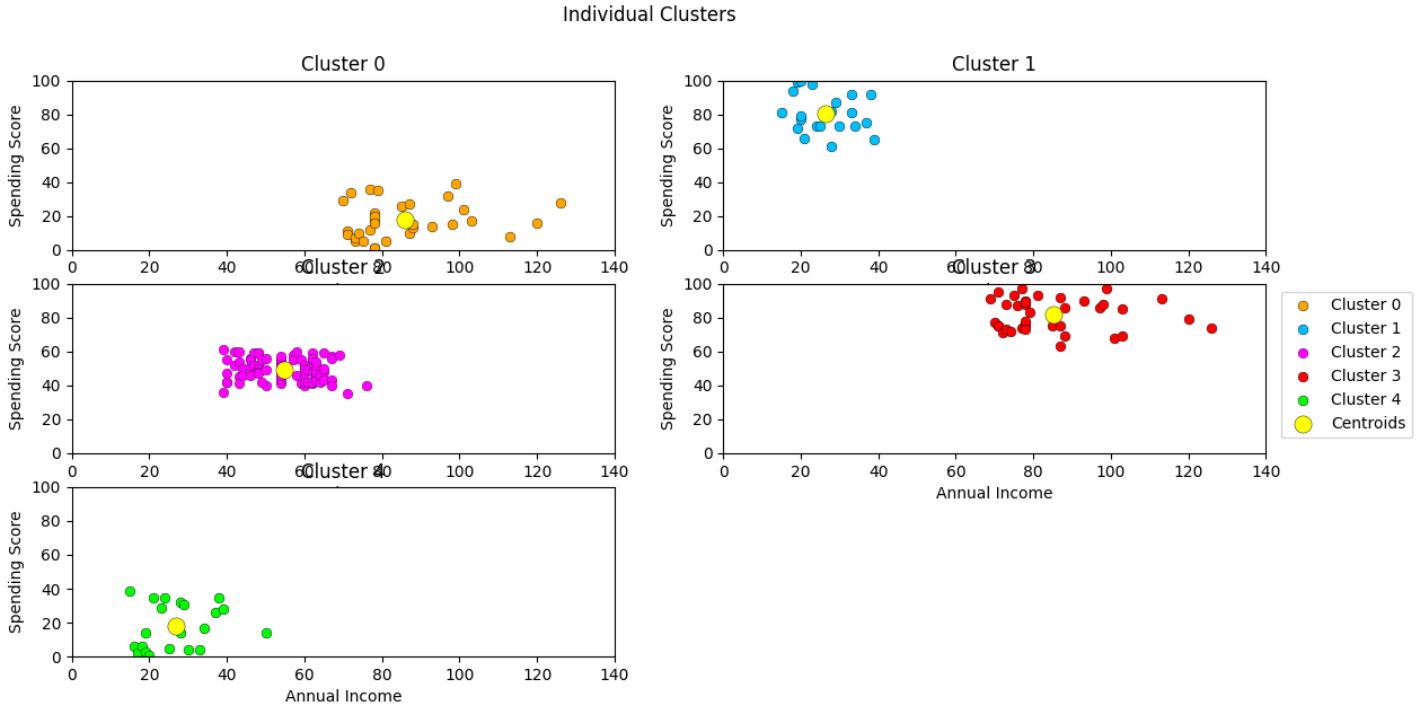
Elbow метод със (в червено) и без (в синьо) outlier-и



Silhouette метод без outlier-и, показващ оптимален брой k клъстери



Резултат от клъстериране по колоните Spending Score и Annual Income


Individual Clusters
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Spending Score
Annual Income
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Centroids

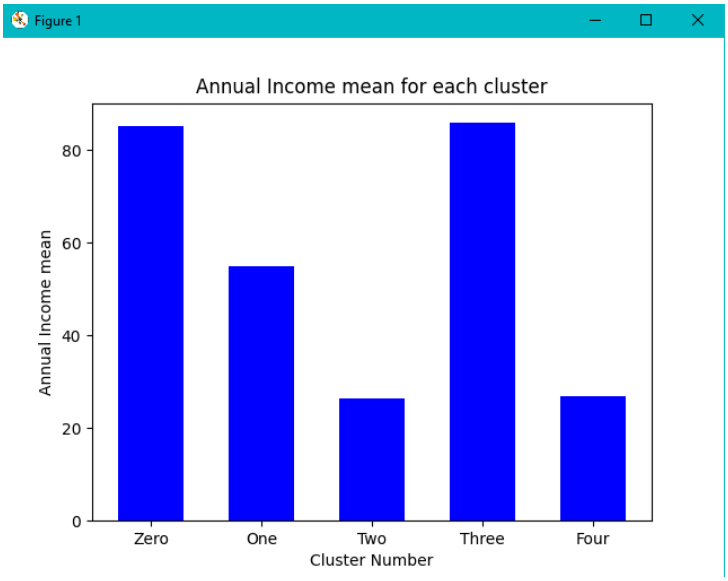


Figure 1
Annual Income mean for each cluster
Annual Income mean
Zero
One
Two
Three
Four
Cluster Number

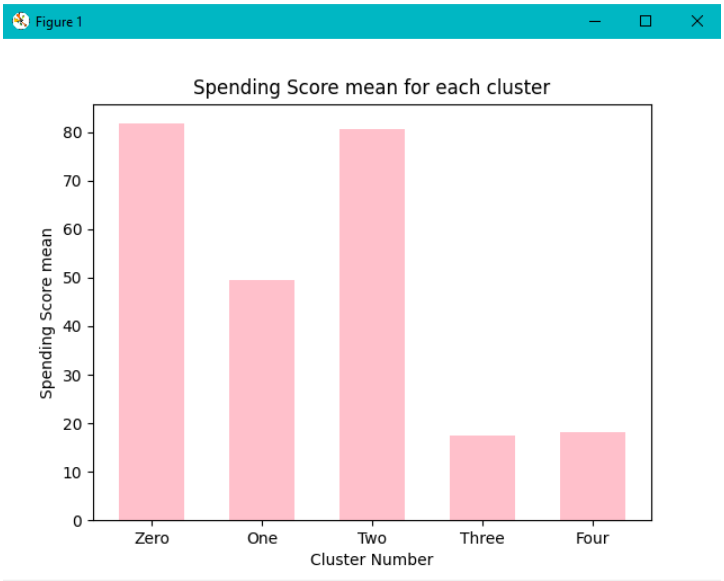


Figure 1
Spending Score mean for each cluster
Spending Score mean
Zero
One
Two
Three
Four
Cluster Number

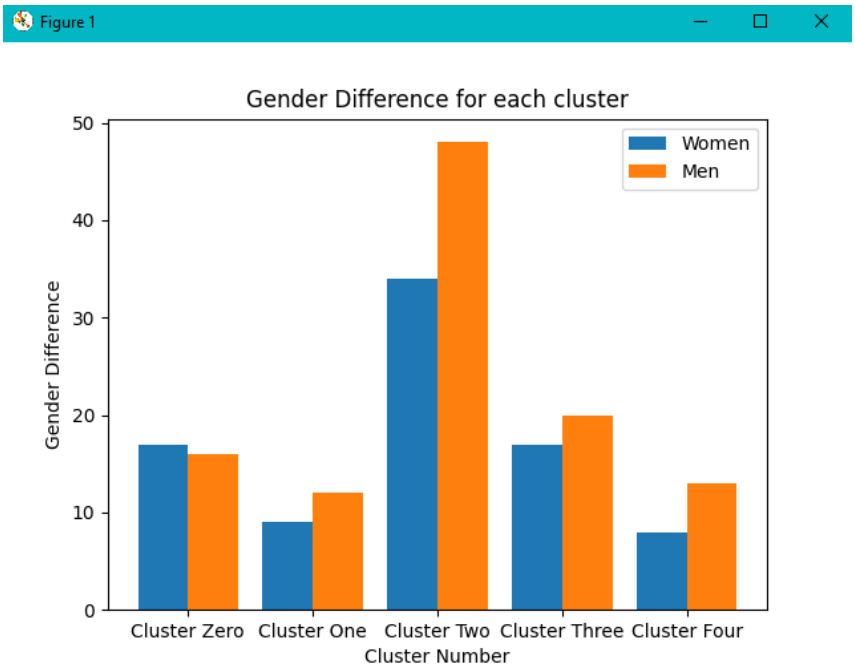


Figure 1
Gender Difference for each cluster
Women
Men
Gender Difference
Cluster Zero
Cluster One
Cluster Two
Cluster Three
Cluster Four
Cluster Number

5. Литература.

- ✓ https://en.wikipedia.org/wiki/Market_segmentation
- ✓ https://www.analyticsvidhya.com/blog/2021/05/k-means-clustering-with-mall-customer-segmentation-data-full-detailed-code-and-explanation/
- ✓ https://www.geeksforgeeks.org/