

СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

КУРСОВ ПРОЕКТ

ПО ИЗКУСТВЕН ИНТЕЛЕКТ

Тема:

8.Сегментиране на клиентите е практиката на разделяне на една клиентска база в групи (клъстери) от лица, които имат сходни демографски или други характеристики.

Студент:

Кристиян Стоянов Стоянов, група 3, фак. №45559

София, юни 2022 г.

1. Формулировка на задачата

Напишете програма, която реализира избран тип сегментиране на дадено множество от клиенти.

1. Използвани алгоритми

За реализацията на сегментирането на данните в отделни клъстери се използва алгоритъма KMeans.

**Начин на работа на алгоритъма:**

Първоначално се избират K на брой точки, които са основата на всеки от клъстерите. Този избор може да се направим на произволен принцип.

След като са избрани първоначалните клъстери, се смята разстоянието на всяка точка до точките избрани за начало на клъстерите. Точката се добавя към клъстера, до който е най-близо.

След като всички точки са добавени към клъстер, се пресмятат средните точки за всеки от клъстерите.

Отново се итерира по всички точки, но този път се пресмятат разстоянията до средните точки. Ако за точка се установи, че е по-близо до центъра на друг клъстер, то клъстера, към който принадлежи се променя, към по близкия.

След това отново се смятат новите центрове на клъстерите и се повтаря местенето на точки между клъстерите, докато не се стигне стабилно разпределение, което не се променя.

Това състояние се взима като решение, но то може да не бъде оптимално. Поради тази причина се повтарят изчисленията, но с различен избор на начални точки за клъстерите.

След като се съберат още “решения”, се пресмята каква е вариацията на стойностите във всеки от клъстерите и се избира решението, което минимизира разстоянията вътре в клъстерите.

Описанието на алгоритъма е на база тази страница (Литература т.3): https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

1. Описание на програмната реализация

Програмата е изградена на набор от функции, които предоставят възможност на потребителя да анализира и сегментира данните в отделни клъстери. Тези функции се извикват от интерпретатор, който чете информация от стандартния вход.

Библиотеки използвани в реализацията на програмата:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from mpl\_toolkits.mplot3d import Axes3D

from sklearn.cluster import KMeans

### Функции:

def read\_data(filePath):

return pd.read\_csv(filePath)

Функцията read\_data приема път към csv файл, съдържащ информацията, която ще бъде анализирана.

def get\_distribution\_by\_column(col, data):

plt.figure(figsize=(10, 6))

sns.set(style = 'whitegrid')

sns.distplot(data[col])

plt.title('Distribution of ' + col, fontsize = 20)

plt.xlabel('Range of ' + col)

plt.ylabel('Count')

plt.ion()

plt.show()

Функцията get\_distribution\_by\_column приема име на колона и заредените данни под формата на data frame. След това, чертае хистограма на екрана, показвайки дистрибуцията на стойностите на подадената колона в подадения data set.

def limit\_columns(columns, data):

tmpDataFrame = data[data.columns]

return tmpDataFrame[columns]

Функцията limit\_columns приема списък от колони и заредените данни под формата на data frame. Връща нов data frame, в който са оставени само получените колони. Това е помощна фукнция, която се използва, за ограничаване до релевантните колони, преди да се използва КMeans алгоритъма върху данните.

def scatterplot\_2d(col1, col2, data, labeled):

plt.figure(figsize=(10,6))

if(labeled):

sns.scatterplot(x = col1,y = col2, hue="label", data = data,

legend='full',s = 60)

else:

sns.scatterplot(x = col1,y = col2, data = data[[col1, col2]] ,s = 60 )

plt.xlabel(col1)

plt.ylabel(col2)

plt.title(col2 + ' vs ' + col1)

plt.ion()

plt.show()

Функцията scatterplot\_2d получава 2 имена на колони, даните под формата на data frame и флаг с име labeled, който пояснява дали ще се визуализират данни, които вече са разделени на отделни клъстери.

def scattlerplot\_3d(col1, col2, col3, data, number\_of\_clusters):

fig = plt.figure(figsize=(20,10))

ax = fig.add\_subplot(111, projection='3d')

colors = ['purple', 'red', 'blue', 'green', 'yellow']

for i in range(0, number\_of\_clusters):

ax.scatter(data[col1][data.label == i], data[col2][data.label == i], data[col3][data.label == i], c=colors[i], s=60)

ax.view\_init(35, 185)

plt.xlabel(col1)

plt.ylabel(col2)

ax.set\_zlabel(col3)

plt.ion()

plt.show()

Функцията scatterplot\_3d е еквивалентна на scatterplot\_2d, но работи, върху вече сегментирани данни и се взимат в предвид, не 2, а 3 критерия при клъстеризацията.

def wcss(max\_clusters, data):

wcss=[]

for i in range(1,max\_clusters):

km=KMeans(n\_clusters=i)

km.fit(data)

wcss.append(km.inertia\_)

return wcss

Фукнцията wuss е помощтна функнция, която пресмята Within Cluster Sum of Squares. Получава като аргумент максималния брой клъстери, за които да се пресмята стойността, и връща масив от стойностите на wcss за 1, 2, …, <max\_clusters>. Използва се при визуализацията на elbow curve, което помага да се избере подходящ брой клъстери, на който да се разбие data set-a.

def elbowcurve(max\_clusters, data):

within\_cluster\_sum\_squares = wcss(max\_clusters, data)

plt.figure(figsize=(12,6))

plt.plot(range(1,max\_clusters),within\_cluster\_sum\_squares)

plt.plot(range(1,max\_clusters),within\_cluster\_sum\_squares, linewidth=2, color="red", marker ="8")

plt.xlabel("K Value")

plt.xticks(np.arange(1,max\_clusters,1))

plt.ylabel("WCSS")

plt.ion()

plt.show()

Функцията elbowcurve приема максималния брой клъстери за който да сметне WCSS и данните под формата на data frame. Използвайки функцията wcss визуализира elbowcurve с разстоянията от центъра на всеки клъстер, спрямо броя на клъстерите. Информацията от тази крива е полезна при избора на подходящ брой клъстери за разбиване на данните.

def label\_clusters(number\_of\_clusters, data, limited\_data):

km = KMeans(number\_of\_clusters)

km.fit(limited\_data)

labels = km.predict(limited\_data)

data["label"] = labels

return data

Функцията label\_clusters използва KMeans алгоритъма и слага етикет на всеки от редовете от data frame-a. Този етикет се използва като идентификатор към кой клъстер попада даден елемент от множеството от данни. Колоната label се използва при оцветяването на отделните точки, при тяхното визуализиране.

def get\_clustered\_ids(data, number\_of\_clusters, id\_row\_name):

for i in range(0, number\_of\_clusters):

row = data[data["label"] == i]

print(f'Number of rows in group\_{i} = {len(row)}')

print(row[id\_row\_name].values)

print('=======================================')

Функцията get\_clustered\_ids получава като аргумент данните под формата на data frame, броя на клъстерите, на който е разбито множеството и името на колоната с идентификатора (id). При изпълнението си, изписва на конзолата броя на елементите във всеки клъстер, както и техните идентификатори, за по-детайлна разбивка на клъстеризацията.

### Интерпретатор на командите:

print("Welcome to market-segmantation!")

print("If you are unsure how to use the program, write 'help'!")

data = None

line = input()

while line != "quit":

tokens = line.split(";")

if tokens[0] == "help":

//skipping the help prints, to save space

line = input()

continue

elif tokens[0] == "read\_data":

data = read\_data(tokens[1])

print("Successfully read data from csv!")

line = input()

continue

if data is None:

print("Please load data before doing further operations!")

line = input()

continue

if tokens[0] == "get\_distribution":

get\_distribution\_by\_column(tokens[1].strip(), data)

elif tokens[0] == "get\_elbowcurve":

limited\_data = limit\_columns(tokens[2:], data)

elbowcurve(int(tokens[1]), limited\_data)

elif tokens[0] == "get\_clusters\_2d":

limited\_data = limit\_columns(tokens[2:], data)

labeled\_data = label\_clusters(int(tokens[1]), data, limited\_data)

scatterplot\_2d(tokens[2], tokens[3], labeled\_data, True)

get\_clustered\_ids(labeled\_data, int(tokens[1]), tokens[4])

elif tokens[0] == "get\_clusters\_3d":

limited\_data = limit\_columns(tokens[2:], data)

labeled\_data = label\_clusters(int(tokens[1]), data, limited\_data)

scattlerplot\_3d(tokens[2], tokens[3], tokens[4], labeled\_data, int(tokens[1]))

get\_clustered\_ids(labeled\_data, int(tokens[1]), tokens[5])

else:

print("Invalid command. Please check the doc, or use 'help'!")

line = input()

Това е интерпретатора на командите прочетени от конзолата. Представлява цикъл while, който продължава да се изпълнява, докато не получи вход “quit”.

В самия цикъл, командата се разделя на име и аргументи, и спрямо името се извикват правилните функции и аргументите се поставят на правилните места в извиканите функции.

1. Примери, илюстриращи работата на програмната система

За примерни данни ще използваме market\_data.csv (източник: <https://github.com/prateekmaj21/Mall-Customer-Segementation>) Като съм променил имената на колоните, за да се изписват по-лесно на конзолата. Поставете файла до market-segmentation.py и изпълнете програмата използвайки

python3 market-segmentation.py

* за да се изпълни програмата е необходимо да има инсталиран python версия 3 и python3 да бъде в PATH променливата на машината (може да варира между отделните операционни системи).

Данните, с който рабитм изглеждат по следния начин:

CustomerID,Gender,Age,Annual Income,Spending Score

1,Male,19,15,39

2,Male,21,15,81

3,Female,20,16,6

4,Female,23,16,77

Имаме индетификатор на всеки клиент (CustomerID), пол(Gender), възраст (Age), годишни доходи (Annual Income), оценка на харченето от 0 до 100 (Spending Score).

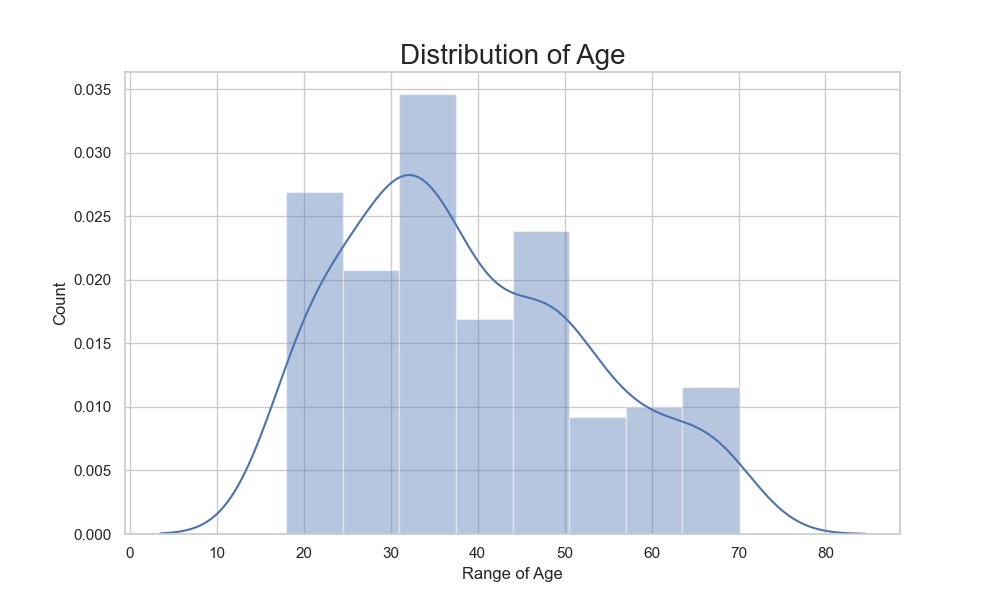
Стартираме market-segmentation.py и зареждаме данните използвайки read\_data командата (използвайки командата help, може да получим информация за формата на входа, който очаква програмата).

Вход: read\_data;market\_data.csv

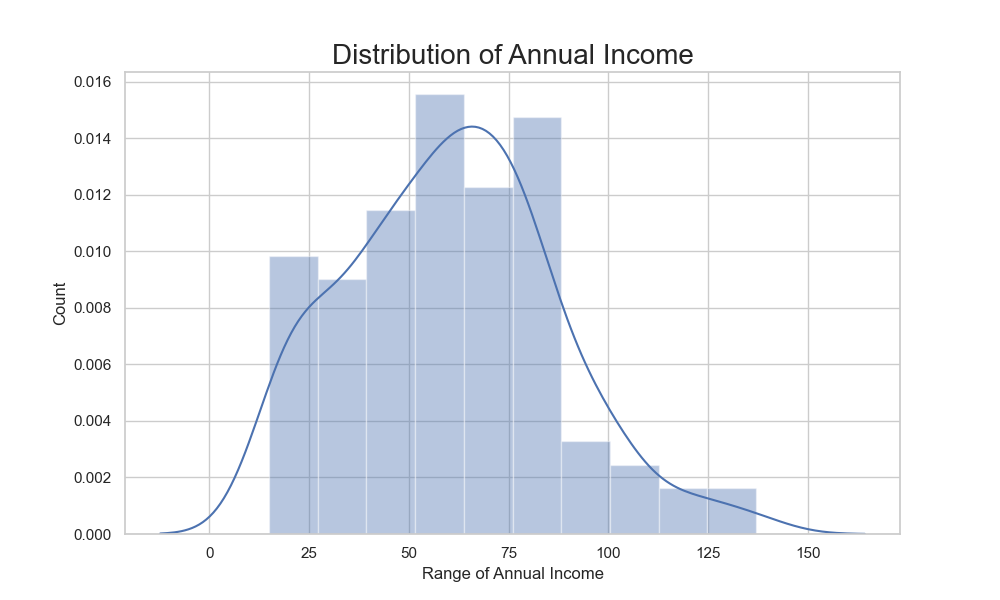
Изход: Successfully read data from csv!

След като успешно сме заредили програмата може да разгледаме дистрибуциите на елементите в данните по дадени параметри, например възраст и доходи:

Вход: get\_distribution;Age

Изход:

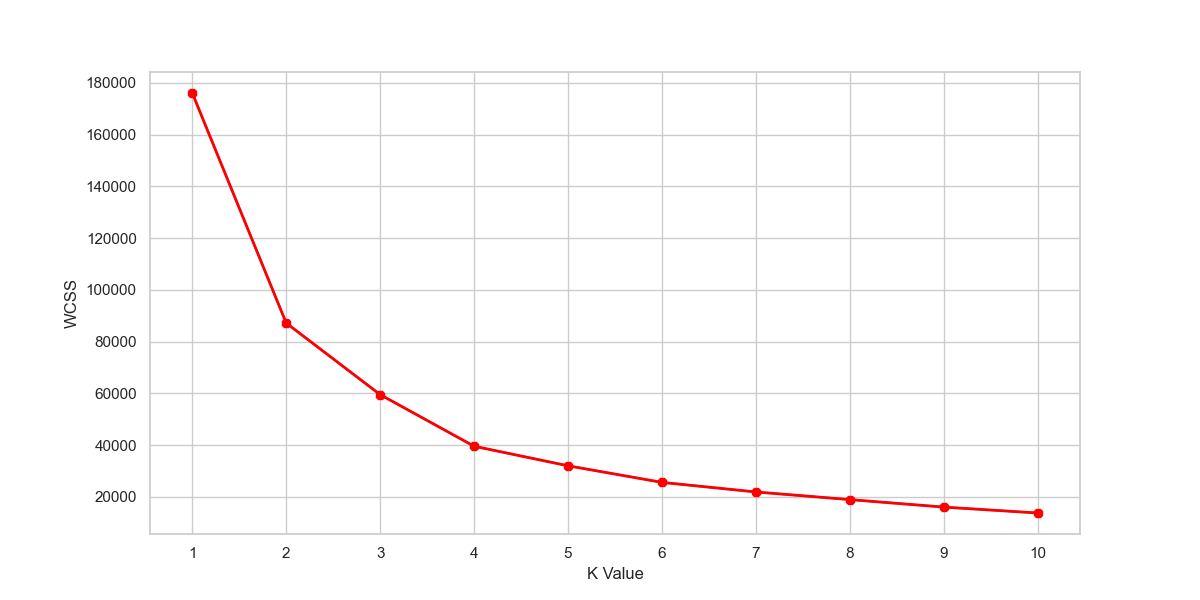
Вход: get\_distribution;Annual Income

Изход: 

Нека направим разбивка по тези два атрибута.

За целта, първо ще разгледаме elbow curve, спрямо броя клъстери, за да преценим, какъв брой, би дал най-полезен резултат.

Вход: get\_distribution;Annual Income

Изход:

Забелязва се, че с увеличаването на броя клъстери, разстоянието на елемнтите в даден клъстер от центъра му, намаля. Избираме точката, която се намира в elbow joint-a, която е съответно 5.

Нека направим сегментиране на 5 клъстера, по атрибутите възраст и годишни приходи:

Вход: get\_clusters\_2d;5;Age;Annual Income;CustomerID

Изход: Number of rows in group\_0 = 43

[124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141

142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159

160 161 162 163 164 165 166]

=======================================

Number of rows in group\_1 = 40

[41 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

66 67 68 69 70 71 72 73 74 75 76 77 78 80 81 83]

=======================================

Number of rows in group\_2 = 42

[ 79 82 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99

100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117

118 119 120 121 122 123]

=======================================

Number of rows in group\_3 = 41

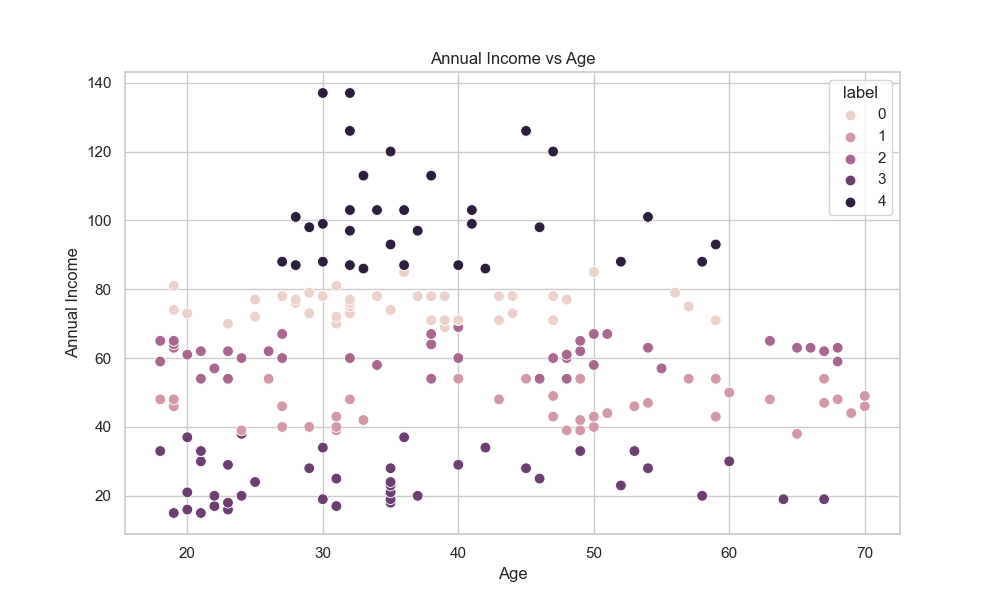
[ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 42]

=======================================

Number of rows in group\_4 = 34

[167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184

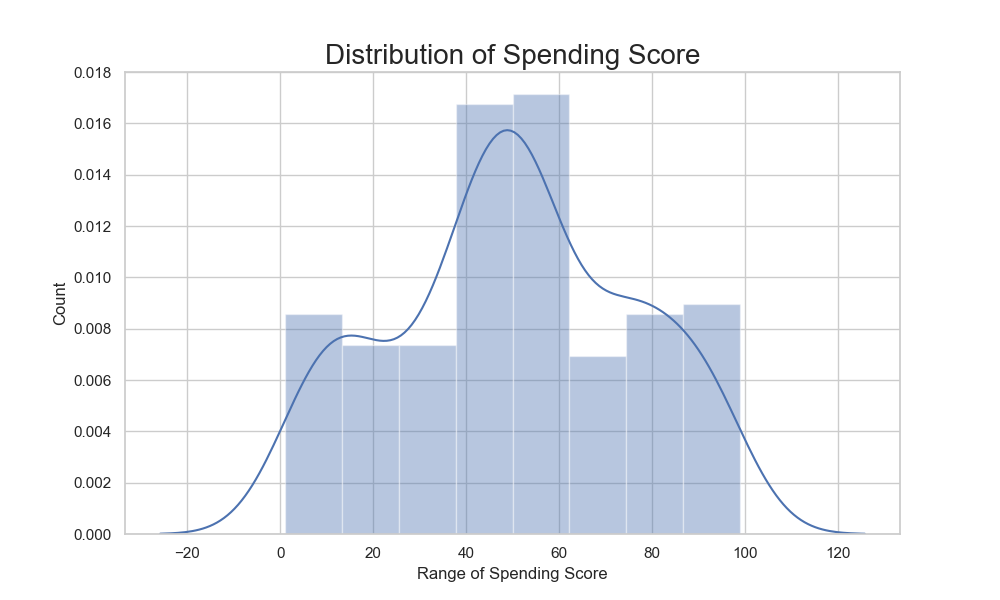
185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200]

Забелязва се, че заради равномерното разпределение на хора от различни възрасти, клъстеризацията зависи много повече върху годишните приходи (Аnnual Income) на клиентите. Поради тази причина, ще добавим още един атрибут, към критериите за клъстеризация.

Нека разгледаме дистрибуцията на оценката на харчене (spending score) на клиентите, тъй като полът е по-ограничен атрибут с по-малко възможни стойности.

Вход: get\_distribution;Spending Score

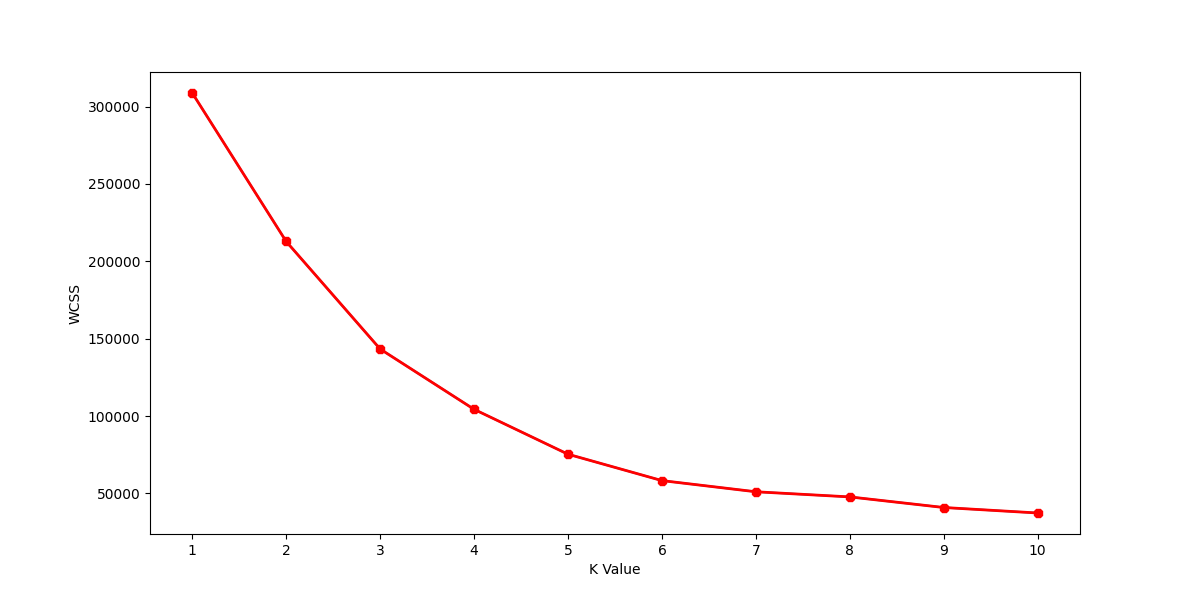
Изход:



Наблюдава се концентрация на в стойностите 40-60.

Нека добавим spending score към критериите за сегментиране и разгледаме elbow curve-a с 3 атрибута.

Вход: get\_elbowcurve;11;Age;Annual Income;Spending Score

Изход:

Забелязва се че elbow joint-a е отново при 5 клъстера. Така че, нека направим сегментиране на 5 клъстера по 3 атрибута - възраст, годишни доходи, оценка на харченето

Вход: get\_clusters\_3d;5;Age;Annual Income;Spending Score;CustomerID

Изход: Number of rows in group\_0 = 38

[125 127 129 131 133 135 137 139 141 143 145 147 149 151 153 155 157 159

161 163 165 167 169 171 173 175 177 179 181 183 185 187 189 191 193 195

197 199]

=======================================

Number of rows in group\_1 = 25

[ 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 52

53]

=======================================

Number of rows in group\_2 = 68

[ 54 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73

74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91

92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109

110 111 112 113 114 115 116 117 118 119 120 121 122 123]

=======================================

Number of rows in group\_3 = 39

[124 126 128 130 132 134 136 138 140 142 144 146 148 150 152 154 156 158

160 162 164 166 168 170 172 174 176 178 180 182 184 186 188 190 192 194

196 198 200]

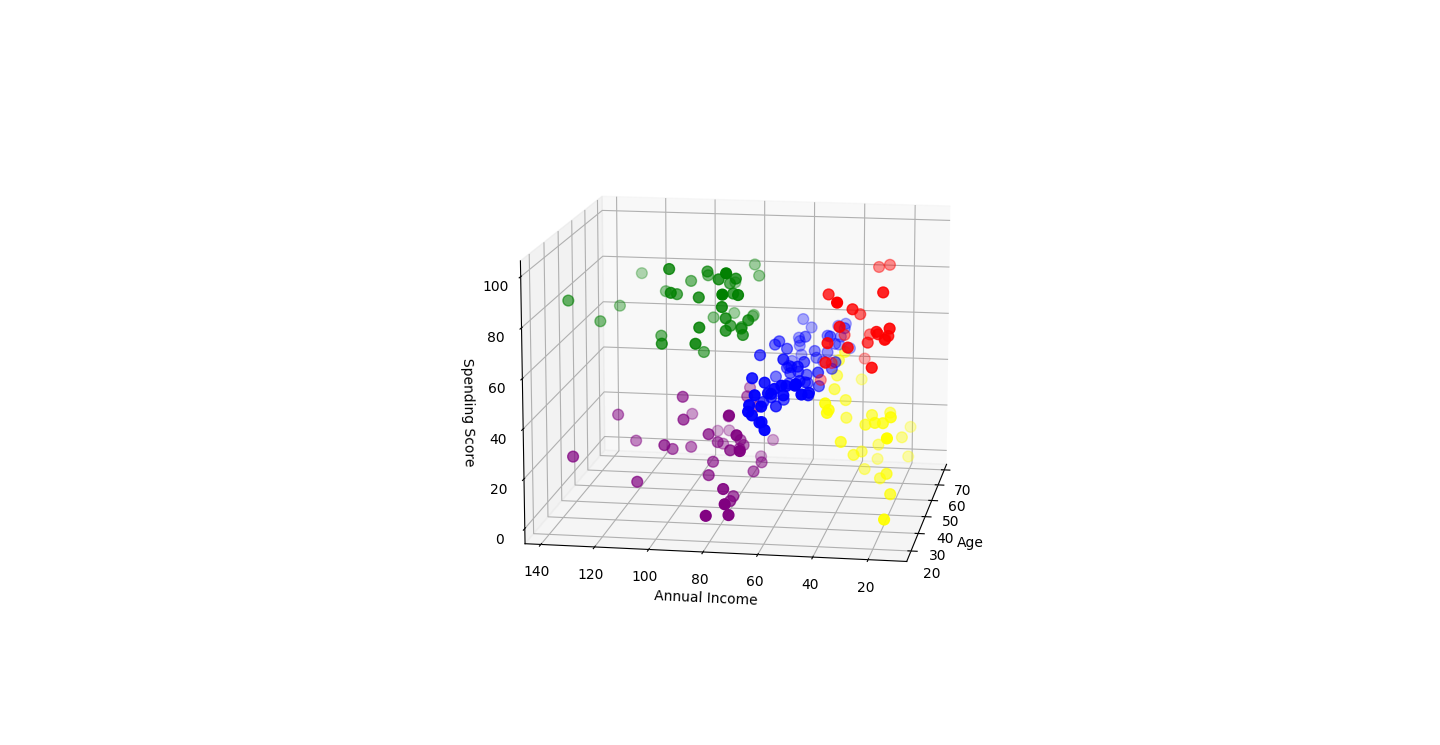
=======================================

Number of rows in group\_4 = 30

[ 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47

48 49 50 51 55 56]

=======================================



С това приключва демонстрацията на работа на програмата.

1. Литература

1. K-Means clustering with Mall Customer Segmentation Data | Full Detailed Code and Explanation by [Prateek Majumder](https://www.analyticsvidhya.com/blog/author/prateekmaj21/) (visited on Jun 21st 2022) - https://www.analyticsvidhya.com/blog/2021/05/k-means-clustering-with-mall-customer-segmentation-data-full-detailed-code-and-explanation/

2. [Mall-Customer-Segementation](https://github.com/prateekmaj21/Mall-Customer-Segementation) repository by [Prateek Majumder](https://www.analyticsvidhya.com/blog/author/prateekmaj21/) <https://github.com/prateekmaj21/Mall-Customer-Segementation>

3. K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks by [Imad Dabbura](https://medium.com/@ImadPhd?source=post_page-----aa03e644b48a--------------------------------) (visited on Jun 23rd 2022) - https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a