

Name: Allan Rodrigues
Class: TE IT A
Roll no: 59
Pid:191104

St. Francis Institute of Technology, Mumbai-400 103
Department of Information Technology

A.Y. 2021-2022
Class: TE-ITA/B, Semester: VI

Subject: **Data Science Lab**

Experiment – 7: To implement Clustering.

1. **Aim:** To implement unsupervised learning with clustering concepts.
2. **Objectives:** After study of this experiment, the student will be able to
 - Understand clustering.
3. **Outcomes:** After study of this experiment, the student will be able to
 - Understand concepts of clustering in data science.
4. **Prerequisite:** Fundamentals of Python Programming and Database Management System.
5. **Requirements:** Python Installation, Personal Computer, Windows operating system, Internet Connection, Microsoft Word.
6. **Pre-Experiment Exercise:**

Brief Theory:

- Concept of clustering machine learning. (Naive Byes, ID3, KNN, Random Forest)

Laboratory Exercise

- A. **Procedure:** (Mall_Customers Dataset)

Refer a separate DSL_EXPT7_Clustering.ipynb file for commands.

- B. Paste Screenshots of above commands

DSLExt7.ipynb

```
[1] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline

[2] data=pd.read_csv("Mall_Customers.csv")

[3] print("Number of customers we have data for-", len(data))

Number of customers we have data for- 200

data.head()
```

	CustomerID	Genre	Age	Annual Income (k\$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

corr() is used to find the pairwise correlation of all columns in the dataframe. Any na values are automatically excluded.

0s completed at 22:18

DSLExt7.ipynb

```
[5] data.corr()
```

	CustomerID	Age	Annual Income (k\$)	Spending Score (1-100)
CustomerID	1.000000	-0.026763	0.977548	0.013835
Age	-0.026763	1.000000	-0.012398	-0.327227
Annual Income (k\$)	0.977548	-0.012398	1.000000	0.009903
Spending Score (1-100)	0.013835	-0.327227	0.009903	1.000000

Distribution of Data


```
#Distribution of Annual Income
plt.figure(figsize=(10, 6))
sns.set(style = 'whitegrid')
sns.distplot(data['Annual Income (k$)'])
plt.title('Distribution of Annual Income (k$)', fontsize = 20)
plt.xlabel('Range of Annual Income (k$)')
plt.ylabel('Count')
```

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code

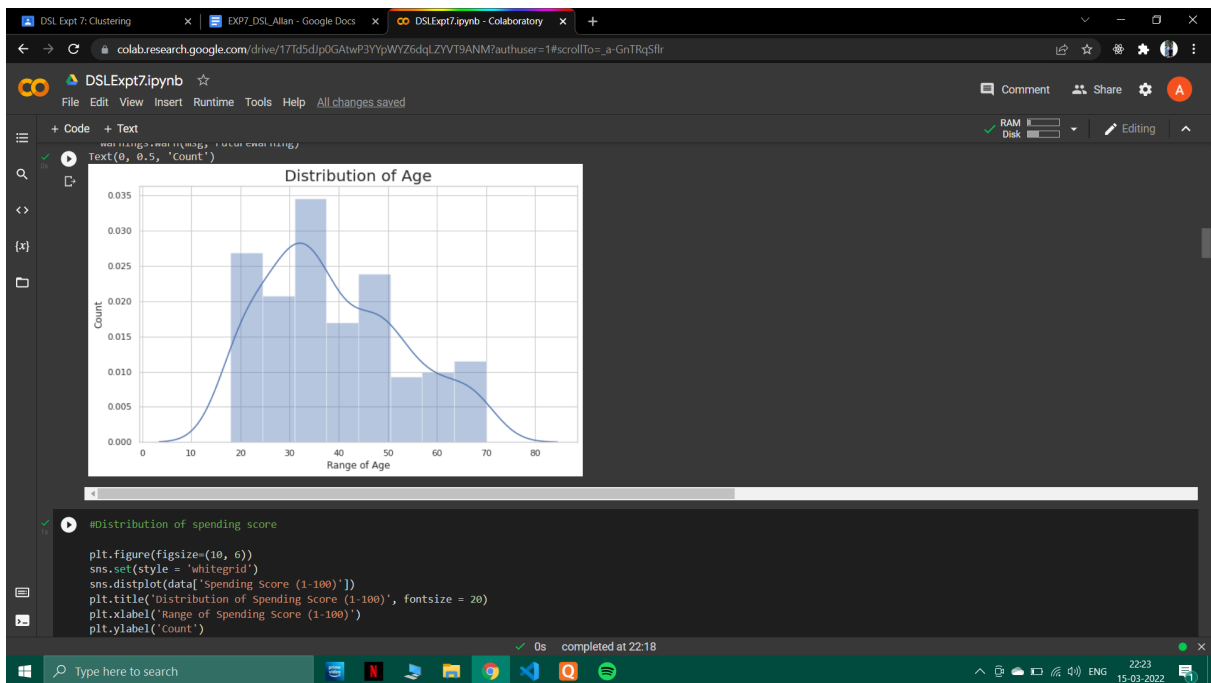
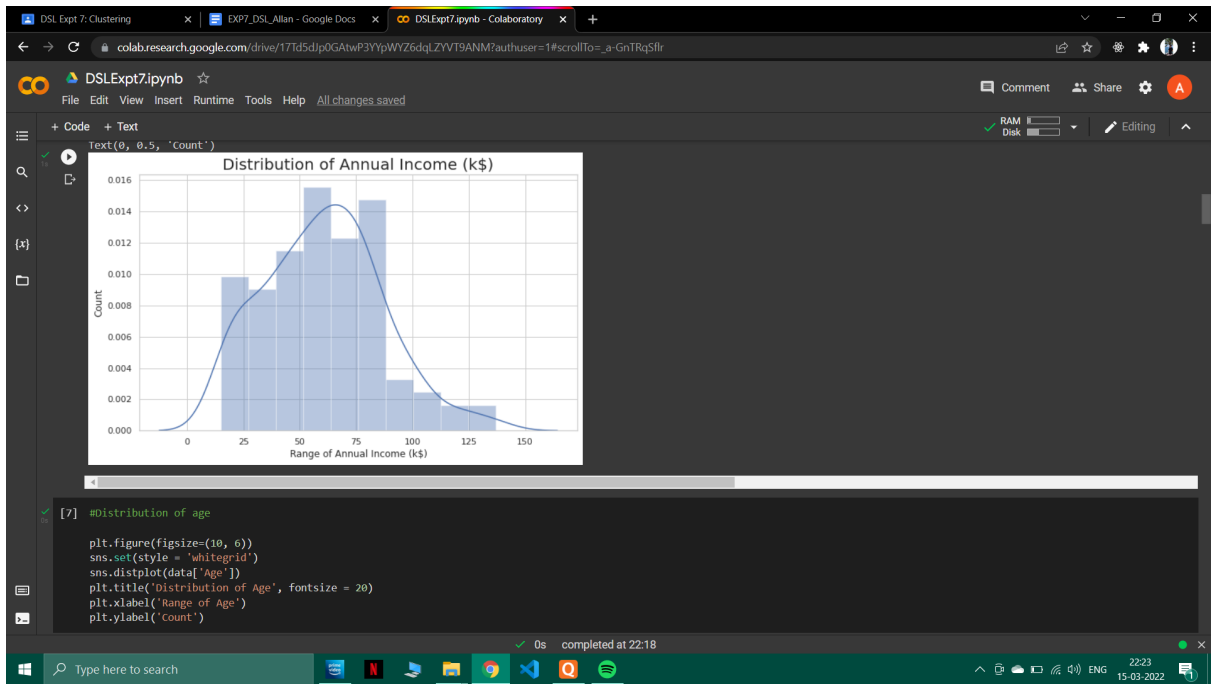
warnings.warn(msg, FutureWarning)

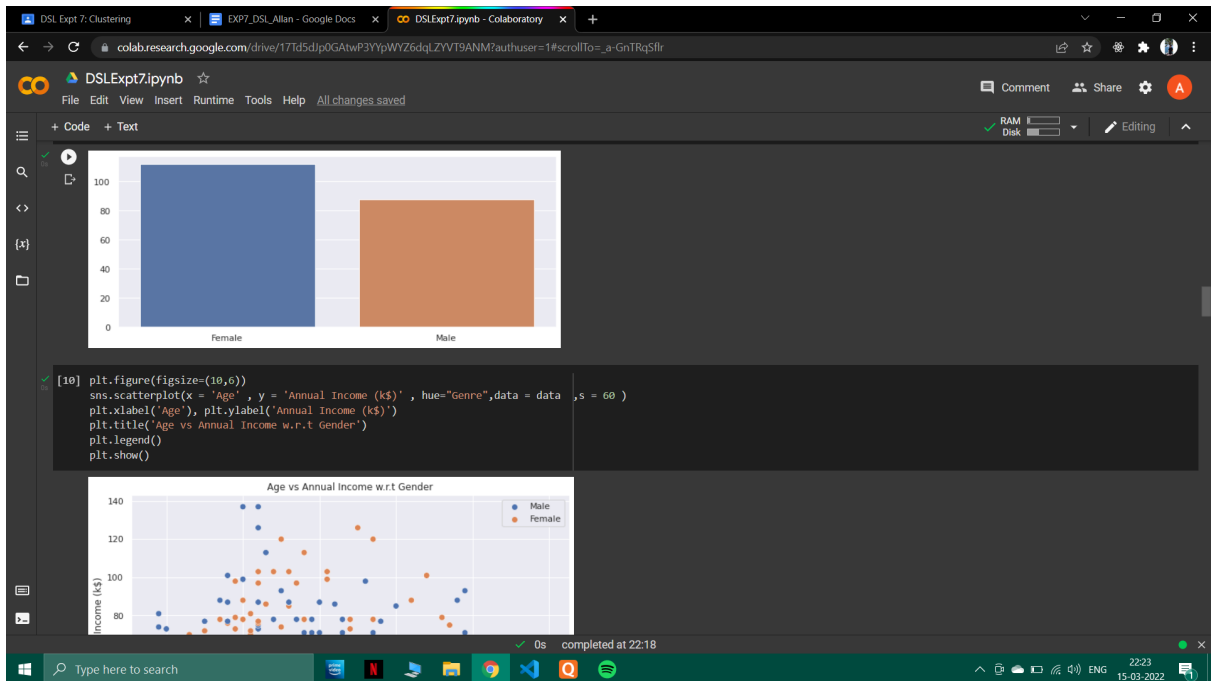
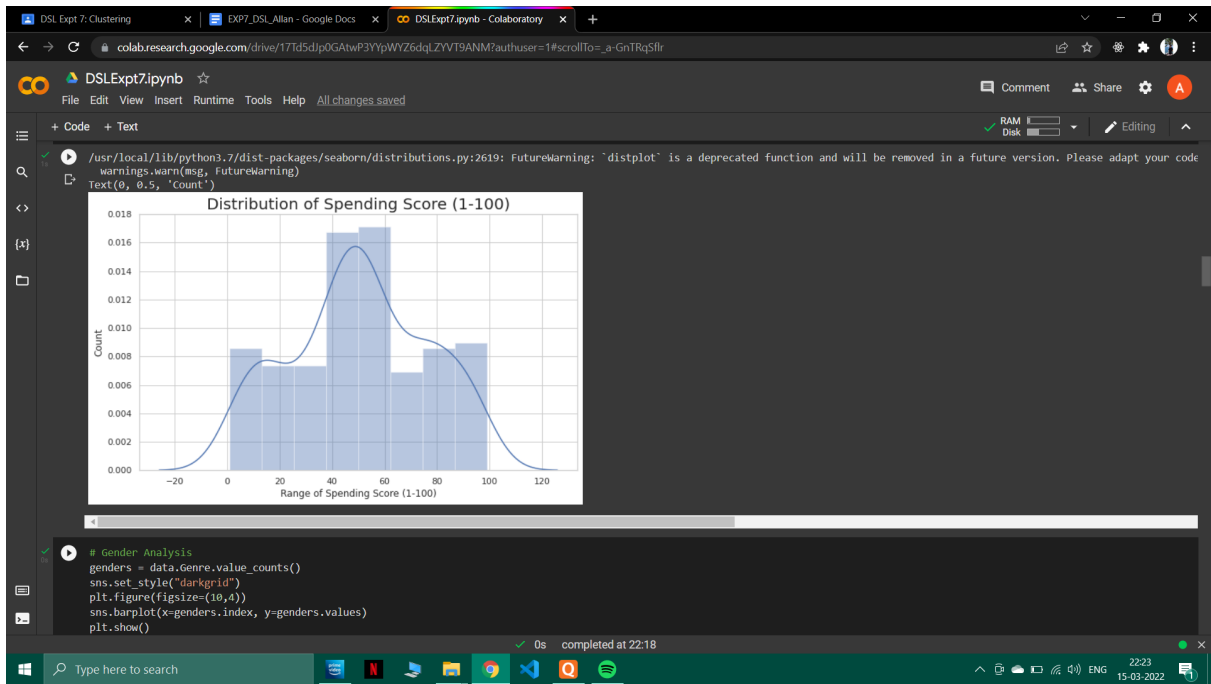
Text(0, 0.5, 'Count')

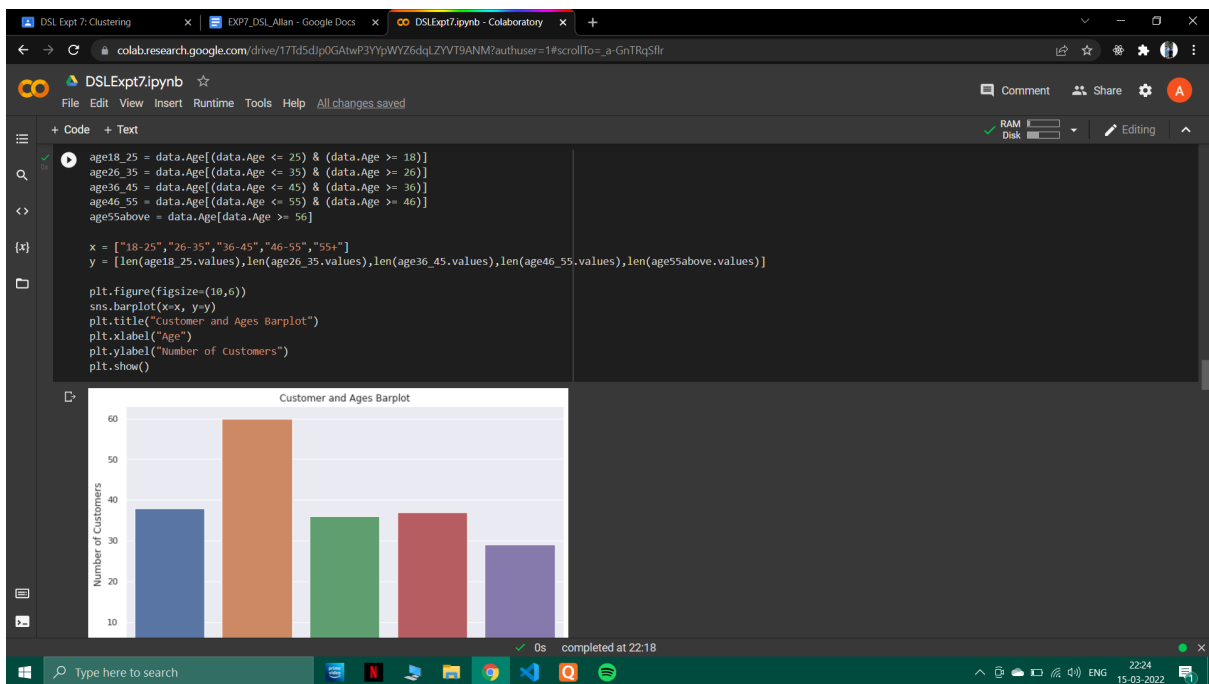
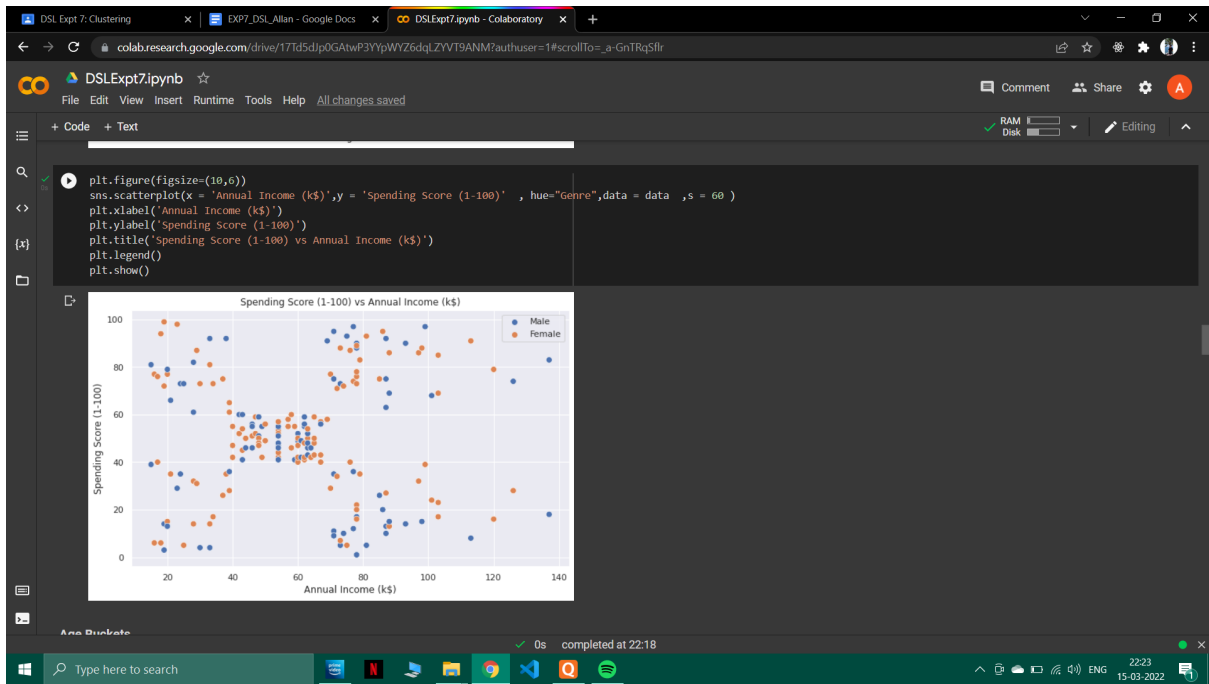
Distribution of Annual Income (k\$)

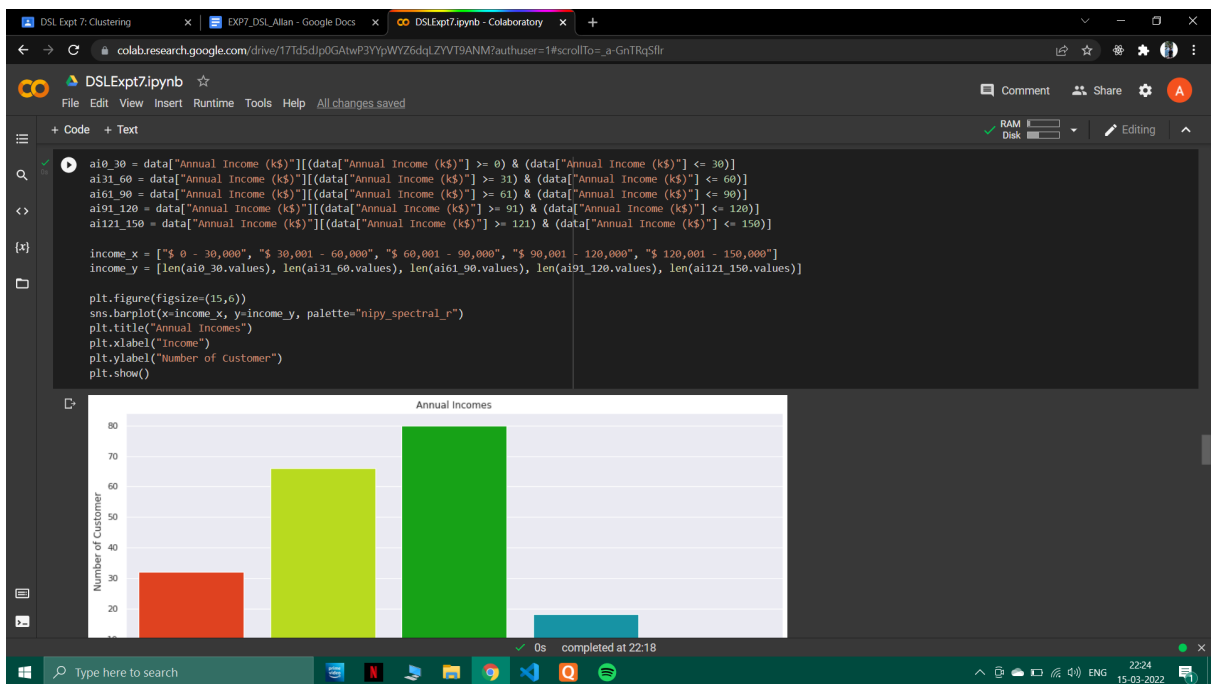
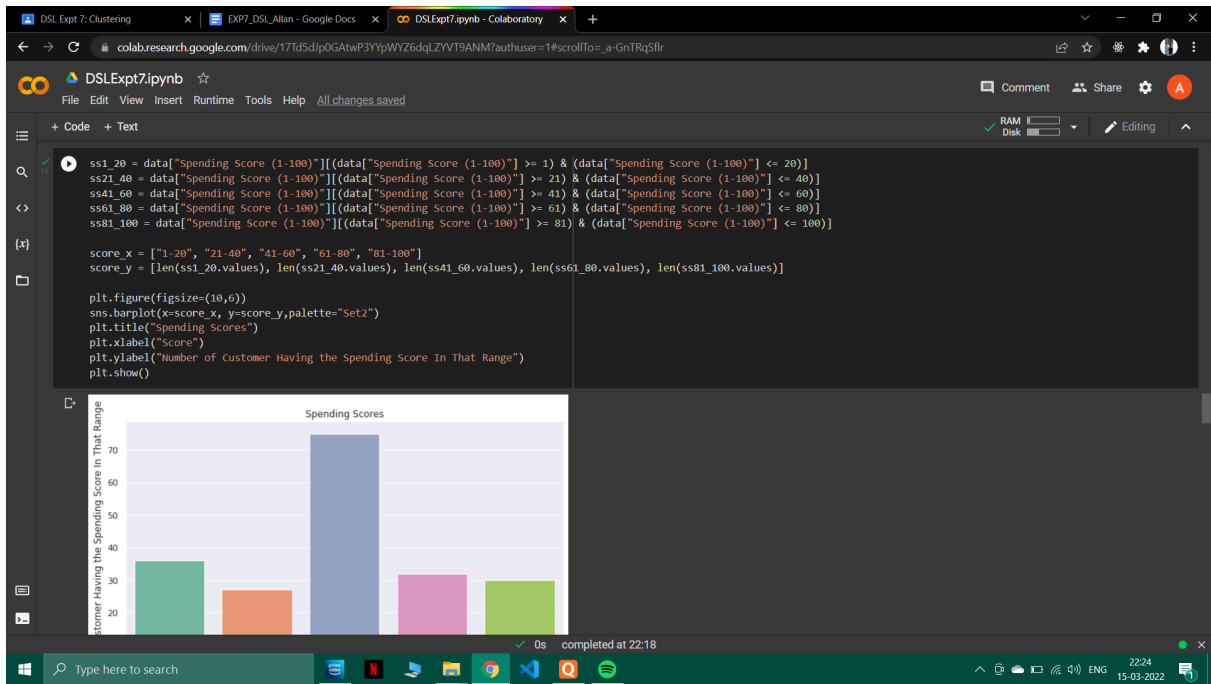


0s completed at 22:18









DSLExt7.ipynb - Colaboratory

colabresearch.google.com/drive/17Td5djp0GAtwP3YypWYZ6dQLZYVT9ANM?authuser=1#scrollTo=nk50EeRYavhx

DSLExt7.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[15] data

CustomerID	Genre	Age	Annual Income (k\$)	Spending Score (1-100)
0	1	Male	19	39
1	2	Male	21	81
2	3	Female	20	6
3	4	Female	23	16
4	5	Female	31	17
...
195	196	Female	35	120
196	197	Female	45	126
197	198	Male	32	126
198	199	Male	32	137
199	200	Male	30	137

200 rows x 5 columns

Clustering Based on 2 Features

```
#we take just the Annual Income and Spending score
df1=data[["customerID","genre","Age","Annual Income (k$)","Spending Score (1-100)"]]
X=df1[["Annual Income (k$)","Spending Score (1-100)"]]
```

0s completed at 22:18

DSLExt7.ipynb - Colaboratory

colabresearch.google.com/drive/17Td5djp0GAtwP3YypWYZ6dQLZYVT9ANM?authuser=1#scrollTo=Hbmwe9oxbLl0

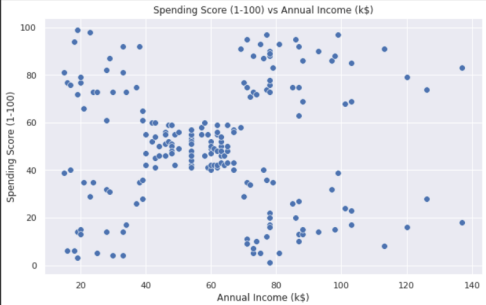
DSLExt7.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
#Scatterplot of the input data

plt.figure(figsize=(10,6))
sns.scatterplot(x = 'Annual Income (k$)',y = 'Spending Score (1-100)', data = X ,s = 60 )
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Spending Score (1-100) vs Annual Income (k$)')
plt.show()
```



0s completed at 22:18

DSLExt7.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[18]

Annual Income (k\$)
20
40
60
80
100
120
140

[19]

```
#Using the elbow method to find out the optimal number of #clusters.  
#KMeans class from the sklearn library.  
from sklearn.cluster import KMeans
```

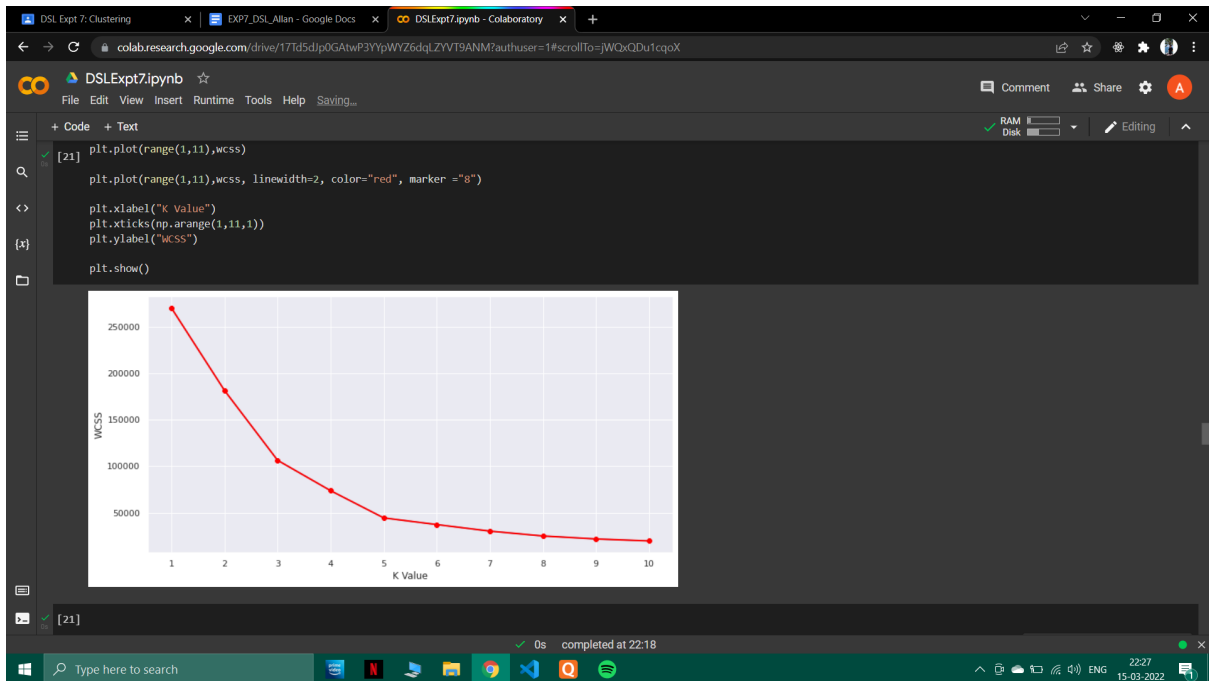
```
wcss=[]  
  
for i in range(1,11):  
    km=KMeans(n_clusters=i)  
    km.fit(X)  
    wcss.append(km.inertia_)
```

#The elbow curve

```
plt.figure(figsize=(12,6))  
  
plt.plot(range(1,11),wcss)  
  
plt.plot(range(1,11),wcss, linewidth=2, color="red", marker="8")  
  
plt.xlabel("K Value")  
plt.xticks(np.arange(1,11,1))  
plt.ylabel("WCSS")  
  
plt.show()
```

0s completed at 22:18

22:25 15-03-2022




```
DSLExt7.ipynb
File Edit View Insert Runtime Tools Help

+ Code + Text

[ ] #Taking 5 clusters
km1=KMeans(n_clusters=5)

[ ] #Fitting the input data
km1.fit(X)

KMeans(n_clusters=5)

[ ] KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
n_clusters=5, n_init=10, random_state=None, tol=0.0001, verbose=0)

KMeans(n_clusters=5)

▶ #predicting the labels of the input data
y=km1.predict(X)

[ ] #adding the labels to a column named label
df1["label"] = y

[ ] #The new dataframe with the clustering done
df1.head()

CustomerID  Genre  Age  Annual Income (k$)  Spending Score (1-100)  label
0  1818  Male  40  15000  46  0
1  1815  Male  41  16000  82  0
2  1814  Male  42  17000  87  0
3  1813  Male  43  18000  93  0
4  1812  Male  44  19000  97  0
5  1811  Male  45  20000  100  0
6  1810  Male  46  21000  103  0
7  1809  Male  47  22000  106  0
8  1808  Male  48  23000  109  0
9  1807  Male  49  24000  112  0
10 1806  Male  50  25000  115  0
11 1805  Male  51  26000  118  0
12 1804  Male  52  27000  121  0
13 1803  Male  53  28000  124  0
14 1802  Male  54  29000  127  0
15 1801  Male  55  30000  130  0
16 1800  Male  56  31000  133  0
17 1799  Male  57  32000  136  0
18 1798  Male  58  33000  139  0
19 1797  Male  59  34000  142  0
20 1796  Male  60  35000  145  0
21 1795  Male  61  36000  148  0
22 1794  Male  62  37000  151  0
23 1793  Male  63  38000  154  0
24 1792  Male  64  39000  157  0
25 1791  Male  65  40000  160  0
26 1790  Male  66  41000  163  0
27 1789  Male  67  42000  166  0
28 1788  Male  68  43000  169  0
29 1787  Male  69  44000  172  0
30 1786  Male  70  45000  175  0
31 1785  Male  71  46000  178  0
32 1784  Male  72  47000  181  0
33 1783  Male  73  48000  184  0
34 1782  Male  74  49000  187  0
35 1781  Male  75  50000  190  0
36 1780  Male  76  51000  193  0
37 1779  Male  77  52000  196  0
38 1778  Male  78  53000  199  0
39 1777  Male  79  54000  202  0
40 1776  Male  80  55000  205  0
41 1775  Male  81  56000  208  0
42 1774  Male  82  57000  211  0
43 1773  Male  83  58000  214  0
44 1772  Male  84  59000  217  0
45 1771  Male  85  60000  220  0
46 1770  Male  86  61000  223  0
47 1769  Male  87  62000  226  0
48 1768  Male  88  63000  229  0
49 1767  Male  89  64000  232  0
50 1766  Male  90  65000  235  0
51 1765  Male  91  66000  238  0
52 1764  Male  92  67000  241  0
53 1763  Male  93  68000  244  0
54 1762  Male  94  69000  247  0
55 1761  Male  95  70000  250  0
56 1760  Male  96  71000  253  0
57 1759  Male  97  72000  256  0
58 1758  Male  98  73000  259  0
59 1757  Male  99  74000  262  0
60 1756  Male  100 75000 265  0
61 1755  Male  101 76000 268  0
62 1754  Male  102 77000 271  0
63 1753  Male  103 78000 274  0
64 1752  Male  104 79000 277  0
65 1751  Male  105 80000 280  0
66 1750  Male  106 81000 283  0
67 1749  Male  107 82000 286  0
68 1748  Male  108 83000 289  0
69 1747  Male  109 84000 292  0
70 1746  Male  110 85000 295  0
71 1745  Male  111 86000 298  0
72 1744  Male  112 87000 301  0
73 1743  Male  113 88000 304  0
74 1742  Male  114 89000 307  0
75 1741  Male  115 90000 310  0
76 1740  Male  116 91000 313  0
77 1739  Male  117 92000 316  0
78 1738  Male  118 93000 319  0
79 1737  Male  119 94000 322  0
80 1736  Male  120 95000 325  0
81 1735  Male  121 96000 328  0
82 1734  Male  122 97000 331  0
83 1733  Male  123 98000 334  0
84 1732  Male  124 99000 337  0
85 1731  Male  125 100000 340  0
86 1730  Male  126 101000 343  0
87 1729  Male  127 102000 346  0
88 1728  Male  128 103000 349  0
89 1727  Male  129 104000 352  0
90 1726  Male  130 105000 355  0
91 1725  Male  131 106000 358  0
92 1724  Male  132 107000 361  0
93 1723  Male  133 108000 364  0
94 1722  Male  134 109000 367  0
95 1721  Male  135 110000 370  0
96 1720  Male  136 111000 373  0
97 1719  Male  137 112000 376  0
98 1718  Male  138 113000 379  0
99 1717  Male  139 114000 382  0
100 1716  Male  140 115000 385  0
101 1715  Male  141 116000 388  0
102 1714  Male  142 117000 391  0
103 1713  Male  143 118000 394  0
104 1712  Male  144 119000 397  0
105 1711  Male  145 120000 400  0
106 1710  Male  146 121000 403  0
107 1709  Male  147 122000 406  0
108 1708  Male  148 123000 409  0
109 1707  Male  149 124000 412  0
110 1706  Male  150 125000 415  0
111 1705  Male  151 126000 418  0
112 1704  Male  152 127000 421  0
113 1703  Male  153 128000 424  0
114 1702  Male  154 129000 427  0
115 1701  Male  155 130000 430  0
116 1700  Male  156 131000 433  0
117 1699  Male  157 132000 436  0
118 1698  Male  158 133000 439  0
119 1697  Male  159 134000 442  0
120 1696  Male  160 135000 445  0
121 1695  Male  161 136000 448  0
122 1694  Male  162 137000 451  0
123 1693  Male  163 138000 454  0
124 1692  Male  164 139000 457  0
125 1691  Male  165 140000 460  0
126 1690  Male  166 141000 463  0
127 1689  Male  167 142000 466  0
128 1688  Male  168 143000 469  0
129 1687  Male  169 144000 472  0
130 1686  Male  170 145000 475  0
131 1685  Male  171 146000 478  0
132 1684  Male  172 147000 481  0
133 1683  Male  173 148000 484  0
134 1682  Male  174 149000 487  0
135 1681  Male  175 150000 490  0
136 1680  Male  176 151000 493  0
137 1679  Male  177 152000 496  0
138 1678  Male  178 153000 499  0
139 1677  Male  179 154000 502  0
140 1676  Male  180 155000 505  0
141 1675  Male  181 156000 508  0
142 1674  Male  182 157000 511  0
143 1673  Male  183 158000 514  0
144 1672  Male  184 159000 517  0
145 1671  Male  185 160000 520  0
146 1670  Male  186 161000 523  0
147 1669  Male  187 162000 526  0
148 1668  Male  188 163000 529  0
149 1667  Male  189 164000 532  0
150 1666  Male  190 165000 535  0
151 1665  Male  191 166000 538  0
152 1664  Male  192 167000 541  0
153 1663  Male  193 168000 544  0
154 1662  Male  194 169000 547  0
155 1661  Male  195 170000 550  0
156 1660  Male  196 171000 553  0
157 1659  Male  197 172000 556  0
158 1658  Male  198 173000 559  0
159 1657  Male  199 174000 562  0
160 1656  Male  200 175000 565  0
161 1655  Male  201 176000 568  0
162 1654  Male  202 177000 571  0
163 1653  Male  203 178000 574  0
164 1652  Male  204 179000 577  0
165 1651  Male  205 180000 580  0
166 1650  Male  206 181000 583  0
167 1649  Male  207 182000 586  0
168 1648  Male  208 183000 589  0
169 1647  Male  209 184000 592  0
170 1646  Male  210 185000 595  0
171 1645  Male  211 186000 598  0
172 1644  Male  212 187000 601  0
173 1643  Male  213 188000 604  0
174 1642  Male  214 189000 607  0
175 1641  Male  215 190000 610  0
176 1640  Male  216 191000 613  0
177 1639  Male  217 192000 616  0
178 1638  Male  218 193000 619  0
179 1637  Male  219 194000 622  0
180 1636  Male  220 195000 625  0
181 1635  Male  221 196000 628  0
182 1634  Male  222 197000 631  0
183 1633  Male  223 198000 634  0
184 1632  Male  224 199000 637  0
185 1631  Male  225 200000 640  0
186 1630  Male  226 201000 643  0
187 1629  Male  227 202000 646  0
188 1628  Male  228 203000 649  0
189 1627  Male  229 204000 652  0
190 1626  Male  230 205000 655  0
191 1625  Male  231 206000 658  0
192 1624  Male  232 207000 661  0
193 1623  Male  233 208000 664  0
194 1622  Male  234 209000 667  0
195 1621  Male  235 210000 670  0
196 1620  Male  236 211000 673  0
197 1619  Male  237 212000 676  0
198 1618  Male  238 213000 679  0
199 1617  Male  239 214000 682  0
200 1616  Male  240 215000 685  0
201 1615  Male  241 216000 688  0
202 1614  Male  242 217000 691  0
203 1613  Male  243 218000 694  0
204 1612  Male  244 219000 697  0
205 1611  Male  245 220000 700  0
206 1610  Male  246 221000 703  0
207 1609  Male  247 222000 706  0
208 1608  Male  248 223000 709  0
209 1607  Male  249 224000 712  0
210 1606  Male  250 225000 715  0
211 1605  Male  251 226000 718  0
212 1604  Male  252 227000 721  0
213 1603  Male  253 228000 724  0
214 1602  Male  254 229000 727  0
215 1601  Male  255 230000 730  0
216 1600  Male  256 231000 733  0
217 1599  Male  257 232000 736  0
218 1598  Male  258 233000 739  0
219 1597  Male  259 234000 742  0
220 1596  Male  260 235000 745  0
221 1595  Male  261 236000 748  0
222 1594  Male  262 237000 751  0
223 1593  Male  263 238000 754  0
224 1592  Male  264 239000 757  0
225 1591  Male  265 240000 760  0
226 1590  Male  266 241000 763  0
227 1589  Male  267 242000 766  0
228 1588  Male  268 243000 769  0
229 1587  Male  269 244000 772  0
230 1586  Male  270 245000 775  0
231 1585  Male  271 246000 778  0
232 1584  Male  272 247000 781  0
233 1583  Male  273 248000 784  0
234 1582  Male  274 249000 787  0
235 1581  Male  275 250000 790  0
236 1580  Male  276 251000 793  0
237 1579  Male  277 252000 796  0
238 1578  Male  278 253000 799  0
239 1577  Male  279 254000 802  0
240 1576  Male  280 255000 805  0
241 1575  Male  281 256000 808  0
242 1574  Male  282 257000 811  0
243 1573  Male  283 258000 814  0
244 1572  Male  284 259000 817  0
245 1571  Male  285 260000 820  0
246 1570  Male  286 261000 823  0
247 1569  Male  287 262000 826  0
248 1568  Male  288 263000 829  0
249 1567  Male  289 264000 832  0
250 1566  Male  290 265000 835  0
251 1565  Male  291 266000 838  0
252 1564  Male  292 267000 841  0
253 1563  Male  293 268000 844  0
254 1562  Male  294 269000 847  0
255 1561  Male  295 270000 850  0
256 1560  Male  296 271000 853  0
257 1559  Male  297 272000 856  0
258 1558  Male  298 273000 859  0
259 1557  Male  299 274000 862  0
260 1556  Male  300 275000 865  0
261 1555  Male  301 276000 868  0
262 1554  Male  302 277000 871  0
263 1553  Male  303 278000 874  0
264 1552  Male  304 279000 877  0
265 1551  Male  305 280000 880  0
266 1550  Male  306 281000 883  0
267 1549  Male  307 282000 886  0
268 1548  Male  308 283000 889  0
269 1547  Male  309 284000 892  0
270 1546  Male  310 285000 895  0
271 1545  Male  311 286000 898  0
272 1544  Male  312 287000 901  0
273 1543  Male  313 288000 904  0
274 1542  Male  314 289000 907  0
275 1541  Male  315 290000 910  0
276 1540  Male  316 291000 913  0
277 1539  Male  317 292000 916  0
278 1538  Male  318 293000 919  0
279 1537  Male  319 294000 922  0
280 1536  Male  320 295000 925  0
281 1535  Male  321 296000 928  0
282 1534  Male  322 297000 931  0
283 1533  Male  323 298000 934  0
284 1532  Male  324 299000 937  0
285 1531  Male  325 300000 940  0
286 1530  Male  326 301000 943  0
287 1529  Male  327 302000 946  0
288 1528  Male  328 303000 949  0
289 1527  Male  329 304000 952  0
290 1526  Male  330 305000 955  0
291 1525  Male  331 306000 958  0
292 1524  Male  332 307000 961  0
293 1523  Male  333 308000 964  0
294 1522  Male  334 309000 967  0
295 1521  Male  335 310000 970  0
296 1520  Male  336 311000 973  0
297 1519  Male  337 312000 976  0
298 1518  Male  338 313000 979  0
299 1517  Male  339 314000 982  0
300 1516  Male  340 315000 985  0
301 1515  Male  341 316000 988  0
302 1514  Male  342 317000 991  0
303 1513  Male  343 318000 994  0
304 1512  Male  344 319000 997  0
305 1511  Male  345 320000 1000  0
306 1510  Male  346 321000 1003  0
307 1509  Male  347 322000 1006  0
308 1508  Male  348 323000 1009  0
309 1507  Male  349 324000 1012  0
310 1506  Male  350 325000 1015  0
311 1505  Male  351 326000 1018  0
312 1504  Male  352 327000 1021  0
313 1503  Male  353 328000 1024  0
314 1502  Male  354 329000 1027  0
315 1501  Male  355 330000 1030  0
316 1500  Male  356 331000 1033  0
317 1499  Male  357 332000 1036  0
318 1498  Male  358 333000 1039  0
319 1497  Male  359 334000 1042  0
320 1496  Male  360 335000 1045  0
321 1495  Male  361 336000 1048  0
322 1494  Male  362 337000 1051  0
323 1493  Male  363 338000 1054  0
324 1492  Male  364 339000 1057  0
325 1491  Male  365 340000 1060  0
326 1490  Male  366 341000 1063  0
327 1489  Male  367 342000 1066  0
328 1488  Male  368 343000 1069  0
329 1487  Male  369 344000 1072  0
330 1486  Male  370 345000 1075  0
331 1485  Male  371 346000 1078  0
332 1484  Male  372 347000 1081  0
333 1483  Male  373 348000 1084  0
334 1482  Male  374 349000 1087  0
335 1481  Male  375 350000 1090  0
336 1480  Male  376 351000 1093  0
337 1479  Male  377 352000 1096  0
338 1478  Male  378 353000 1099  0
339 1477  Male  379 354000 1102  0
340 1476  Male  380 355000 1105  0
341 1475  Male  381 356000 1108  0
342 1474  Male  382 357000 1111  0
343 1473  Male  383 358000 1114  0
344 1472  Male  384 359000 1117  0
345 1471  Male  385 360000 1120  0
346 1470  Male  386 361000 1123  0
347 1469  Male  387 362000 1126  0
348 1468  Male  388 363000 1129  0
349 1467  Male  389 364000 1132  0
350 1466  Male  390 365000 1135  0
351 1465  Male  391 366000 1138  0
352 1464  Male  392 367000 1141  0
353 1463  Male  393 368000 1144  0
354 1462  Male  394 369000 1147  0
355 1461  Male  395 370000 1150  0
356 1460  Male  396 371000 1153  0
357 1459  Male  397 372000 1156  0
358 1458  Male  398 373000 1159  0
359 1457  Male  399 374000 1162  0
360 1456  Male  400 375000 1165  0
361 1455  Male  401 376000 1168  0
362 1454  Male  402 377000 1171  0
363 1453  Male  403 378000 1174  0
364 1452  Male  404 379000 1177  0
365 1451  Male  405 380000 1180  0
366 1450  Male  406 381000 1183  0
367 1449  Male  407 382000 1186  0
368 1448  Male  408 383000 1189  0
369 1447  Male  409 384000 1192  0
370 1446  Male  410 385000 1195  0
371 1445  Male  411 386000 1198  0
372 1444  Male  412 387000 1201  0
373 1443  Male  413 388000 1204  0
374 1442  Male  414 389000 1207  0
375 1441  Male  415 390000 1210  0
376 1440  Male  416 391000 1213  0
377 1439  Male  417 392000 1216  0
378 1438  Male  418 393000 1219  0
379 1437  Male  419 394000 1222  0
380 1436  Male  420 395000 1225  0
381 1435  Male  421 396000 1228  0
382 1434  Male  422 397000 1231  0
383 1433  Male  423 398000 1234  0
384 1432  Male  424 399000 1237  0
385 1431  Male  425 400000 1240  0
386 1430  Male  426 401000 1243  0
387 1429  Male  427 402000 1246  0
388 1428  Male  428 403000 1249  0
389 1427  Male  429 404000 1252  0
390 1426  Male  430 405000 1255  0
391 1425  Male  431 406000 1258  0
392 1424  Male  432 407000 1261  0
393 1423  Male  433 408000 1264  0
394 1422  Male  434 409000 1267  0
395 1421  Male  435 410000 1270  0
396 1420  Male  436 411000 1273  0
397 1419  Male  437 412000 1276  0
398 1418  Male  438 413000 1279  0
399 1417  Male  439 414000 1282  0
400 1416  Male  440 415000 1285  0
401 1415  Male  441 416000 1288  0
402 1414  Male  442 417000 1291  0
403 1413  Male  443 418000 1294  0
404 1412  Male  444 419000 1297  0
405 1411  Male  445 420000 130
```

```
DSLExt7: Clustering x EXP7_DSLExt7 - Google Docs x DSLExt7.ipynb - Colaboratory x +
colabresearch.google.com/drive/17Td5djp0GAtwP3YypWYz6dqlZYVT9ANM?authuser=1#scrollTo=jWQxQDu1cqoX

DSLExt7.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
RAM 100% Disk 100% Editing
▶ cust1=df1[df1["label"]==1]
print('Number of customer in 1st group=', len(cust1))
print('They are -', cust1["CustomerID"].values)
print("-----")
cust2=df1[df1["label"]==2]
print('Number of customer in 2nd group=', len(cust2))
print('They are -', cust2["CustomerID"].values)
print("-----")
cust3=df1[df1["label"]==3]
print('Number of customer in 3rd group=', len(cust3))
print('They are -', cust3["CustomerID"].values)
print("-----")
cust4=df1[df1["label"]==4]
print('Number of customer in 4th group=', len(cust4))
print('They are -', cust4["CustomerID"].values)
print("-----")
cust5=df1[df1["label"]==5]
print('Number of customer in 5th group=', len(cust5))
print('They are -', cust5["CustomerID"].values)
print("-----")

Number of customer in 1st group= 81
They are - [ 44 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 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
118 119 120 121 122 123 127 133 143]

-----
Number of customer in 2nd group= 23
They are - [ 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45]

-----
Number of customer in 3rd group= 35
They are - [125 129 131 135 137 139 141 145 147 149 151 153 155 157 159 161 163 165
167 169 171 173 175 177 179 181 183 185 187 189 191 193 195 197 199]

0s completed at 22:18
```

```
DSLExt7: Clustering x EXP7_DSLExt7 - Google Docs x DSLExt7.ipynb - Colaboratory x +
colabresearch.google.com/drive/17Td5djp0GAtwP3YypWYz6dqlZYVT9ANM?authuser=1#scrollTo=jWQxQDu1cqoX

DSLExt7.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
RAM 100% Disk 100% Editing
▶ #Now we shall take 3 input Features

df2=df1[["CustomerID","Genre","Age","Annual Income (k$)","Spending Score (1-100)"]]
df2.head()

CustomerID  Genre  Age  Annual Income (k$)  Spending Score (1-100)
0           1   Male   19                15                39
1           2   Male   21                15                81
2           3  Female   20                16                 6
3           4  Female   23                16                77
4           5  Female   31                17                40

[ ] #Taking the features

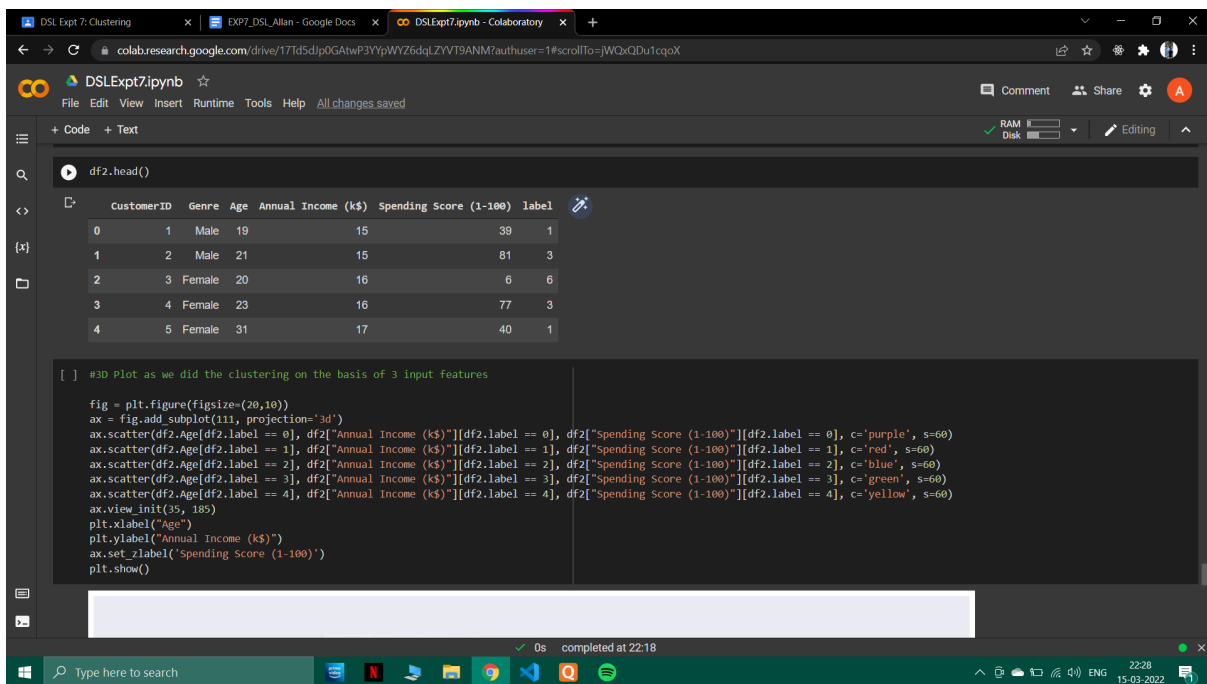
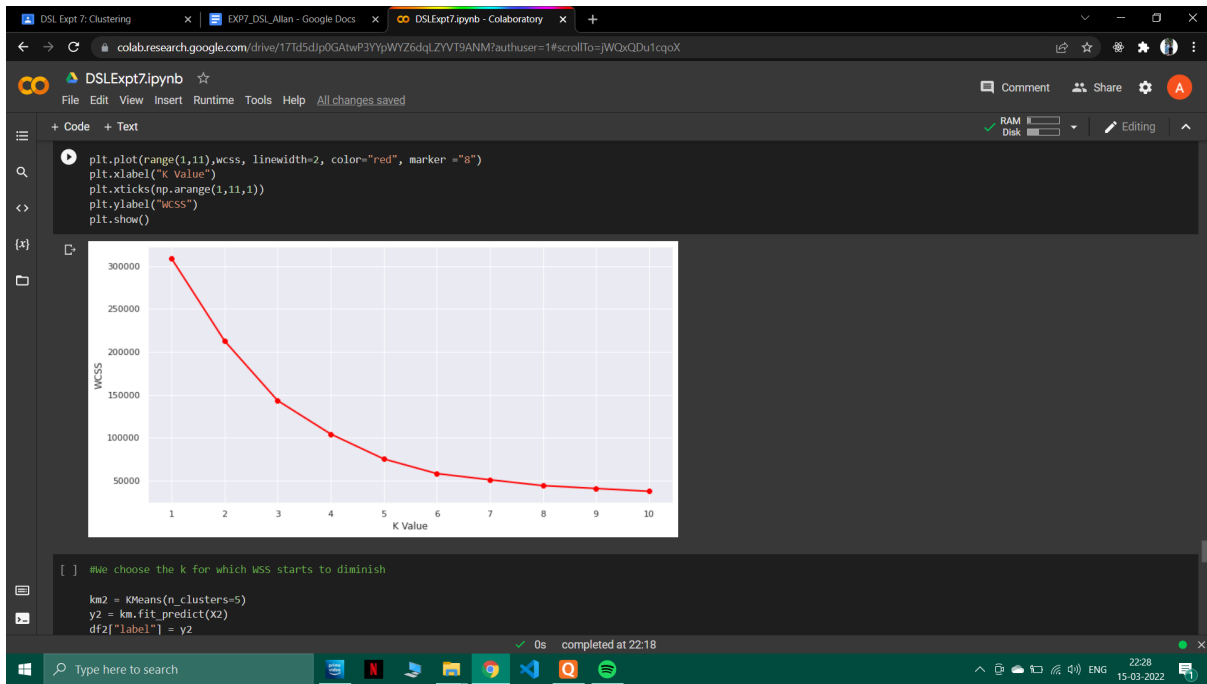
X2=df2[["Age","Annual Income (k$)","Spending Score (1-100)"]]

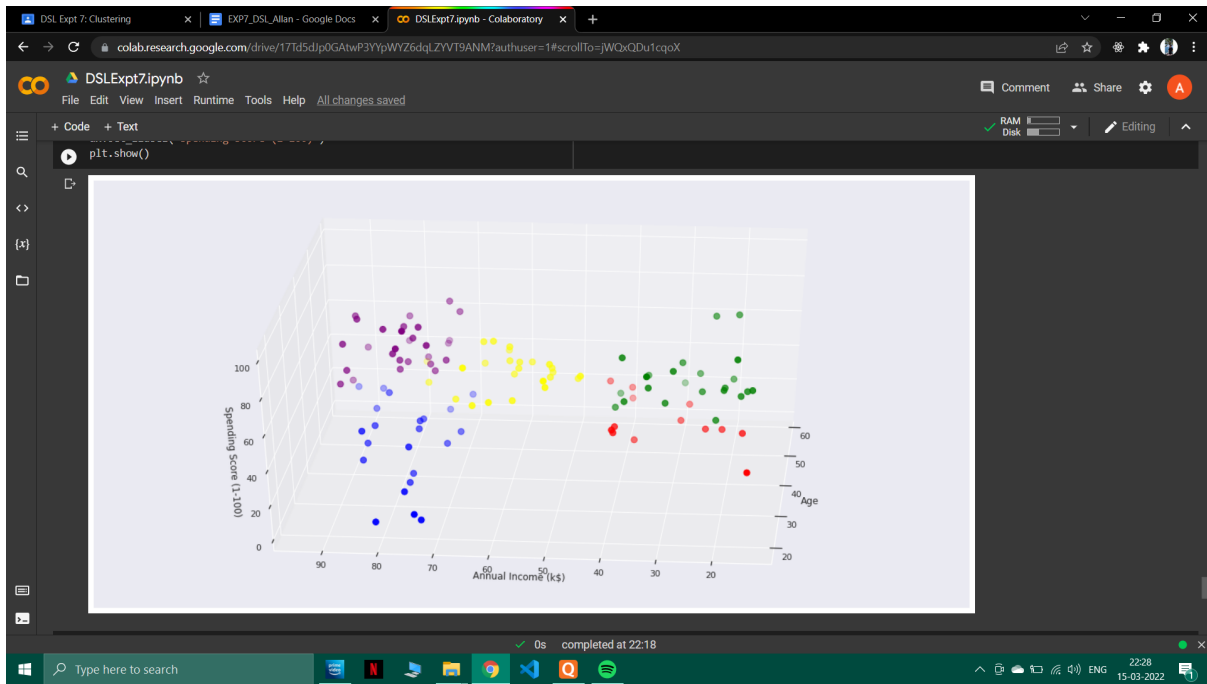
[ ] #Now we calculate the Within Cluster Sum of Squared Errors (WCSS) for different values of k.

wcss = []
for k in range(1,11):
    kmeans = KMeans(n_clusters=k, init="k-means++")
    kmeans.fit(X2)
    wcss.append(kmeans.inertia_)

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

0s completed at 22:18
```





```

cust1=df2[df2["label"]==1]
print('Number of customer in 1st group=', len(cust1))
print('They are -', cust1["CustomerID"].values)
print("-----")
cust2=df2[df2["label"]==2]
print('Number of customer in 2nd group=', len(cust2))
print('They are -', cust2["CustomerID"].values)
print("-----")
cust3=df2[df2["label"]==3]
print('Number of customer in 3rd group=', len(cust3))
print('They are -', cust3["CustomerID"].values)
print("-----")
cust4=df2[df2["label"]==4]
print('Number of customer in 4th group=', len(cust4))
print('They are -', cust4["CustomerID"].values)
print("-----")
cust5=df2[df2["label"]==5]
print('Number of customer in 5th group=', len(cust5))
print('They are -', cust5["CustomerID"].values)
print("-----")

```

```

Number of customer in 1st group= 13
They are - [ 1  5 17 21 27 29 39 43 45 48 49 50 56]
-----
Number of customer in 2nd group= 22
They are - [129 131 135 137 139 141 145 149 151 153 155 157 159 163 165 167 169 171
 173 175 177 179]
-----
Number of customer in 3rd group= 28
They are - [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]
-----
Number of customer in 4th group= 23
They are - [ 2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46]
-----

```

8. Post-Experiments Exercise

A. Extended Theory: (Soft Copy)

- Types of clustering.

1. Centroid-based Clustering

Centroid-based clustering organizes the data into non-hierarchical clusters, in contrast to hierarchical clustering defined below. k-means is the most widely-used centroid-based

clustering algorithm. Centroid-based algorithms are efficient but sensitive to initial conditions and outliers. This course focuses on k-means because it is an efficient, effective, and simple clustering algorithm.

2. Density-based Clustering

Density-based clustering connects areas of high example density into clusters. This allows for arbitrary-shaped distributions as long as dense areas can be connected. These algorithms have difficulty with data of varying densities and high dimensions. Further, by design, these algorithms do not assign outliers to clusters.

3. Distribution-based Clustering

This clustering approach assumes data is composed of distributions, such as Gaussian distributions. The distribution-based algorithm clusters data into three Gaussian distributions. As distance from the distribution's center increases, the probability that a point belongs to the distribution decreases. The bands show that decrease in probability. When you do not know the type of distribution in your data, you should use a different algorithm.

4. Hierarchical Clustering

Hierarchical clustering creates a tree of clusters. Hierarchical clustering, not surprisingly, is well suited to hierarchical data, such as taxonomies. In addition, another advantage is that any number of clusters can be chosen by cutting the tree at the right level.

B. Questions:

■ Application of Clustering

Clustering techniques can be used in various areas or fields of real-life examples such as data mining, web cluster engines, academics, bioinformatics, image processing & transformation etc.

Recommendation engines

In this method, the clustering method provided an idea of like-minded users. The computation/estimation as data provided by several users is leveraged for improving the performance of collaborative filtering methods. And this can be implemented for rendering recommendations in diverse applications.

Market and Customer segmentation

A process of splitting the target market into smaller and more defined categories is known as market segmentation. This segments customers/audiences into groups of similar

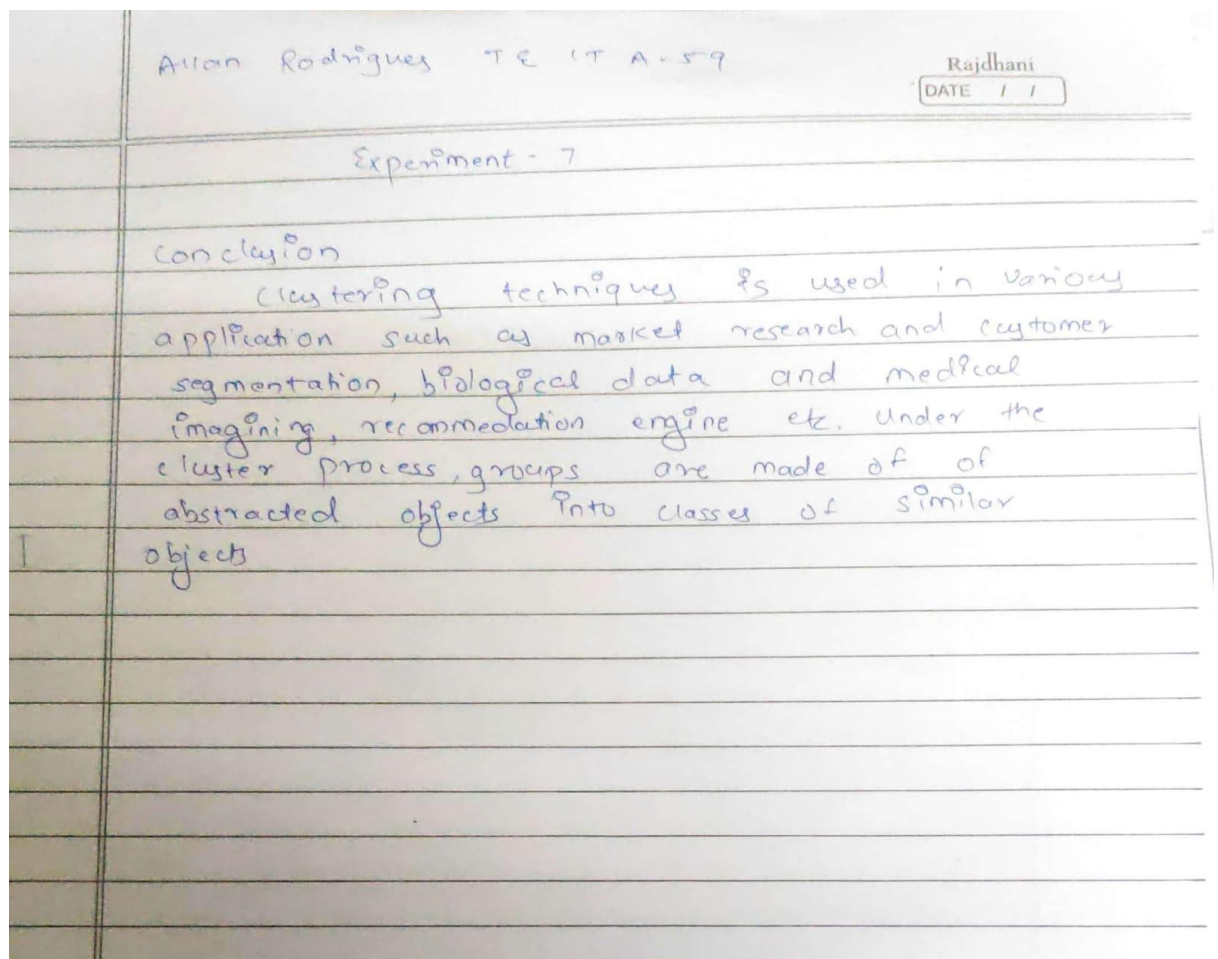
characteristics (needs, location, interests or demographics) where target and personalization, under it, is an immense business.

Social Network Analysis (SNA)

Clustering methods are required in such analysis in order to map and measure the relationship and conflicts amid people, groups, companies, computer networks, and other similar connected information/knowledge entities.

C. Conclusion:

Write the significance of the topic studied in the experiment.



CS Scanned with CamScanner

D. References:

1. <https://machinelearningmastery.com/clustering-algorithms-with-python/>
2. <https://developers.google.com/machine-learning/clustering/clustering-algorithms>