7hxs4pifi

October 8, 2023

0.1 NAME - AKARSHIT MISRA

0.2 REG NO. 21BAI1597

0.2.1 Data Preprocessing

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly as py
import plotly.graph_objs as go
from sklearn.cluster import KMeans
import warnings
import os
```

[2]:
```
df = pd.read_csv('/content/Mall_Customers.csv')
df.head()
```

[2]:

	CustomerID	Gender	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

[3]:
```
df.shape
```

[3]: (200, 5)

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
```

```
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

[5]:
```
df.isnull().sum()
```

[5]:
```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

[6]:
```
df.describe()
```

[6]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

[10]:
```
df.Spending_Score.nunique()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-10-266e4f136b3f> in <cell line: 1>()
----> 1 df.Spending_Score.nunique()

/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py in␣
 ↪__getattr__(self, name)
   5900         ):
   5901             return self[name]
-> 5902         return object.__getattribute__(self, name)
   5903
   5904     def __setattr__(self, name: str, value) -> None:

AttributeError: 'DataFrame' object has no attribute 'Spending_Score'
```

[8]:
```
df.Age.unique()
```

[8]:
```
array([19, 21, 20, 23, 31, 22, 35, 64, 30, 67, 58, 24, 37, 52, 25, 46, 54,
       29, 45, 40, 60, 53, 18, 49, 42, 36, 65, 48, 50, 27, 33, 59, 47, 51,
       69, 70, 63, 43, 68, 32, 26, 57, 38, 55, 34, 66, 39, 44, 28, 56, 41])
```

[11]:
```
df.Age.value_counts()
```

[11]:
```
32    11
35     9
19     8
31     8
30     7
49     7
40     6
38     6
47     6
27     6
36     6
23     6
34     5
20     5
29     5
50     5
48     5
21     5
24     4
18     4
28     4
67     4
59     4
54     4
43     3
60     3
45     3
39     3
33     3
37     3
22     3
25     3
46     3
68     3
52     2
44     2
66     2
57     2
26     2
53     2
42     2
```

```
63     2
70     2
51     2
58     2
65     2
41     2
55     1
69     1
64     1
56     1
Name: Age, dtype: int64
```

[12]:
```
df.corr()
```

```
<ipython-input-12-2f6f6606aa2c>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  df.corr()
```

[12]:

	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

[13]:
```
df.corr().Age.sort_values(ascending =False)
```

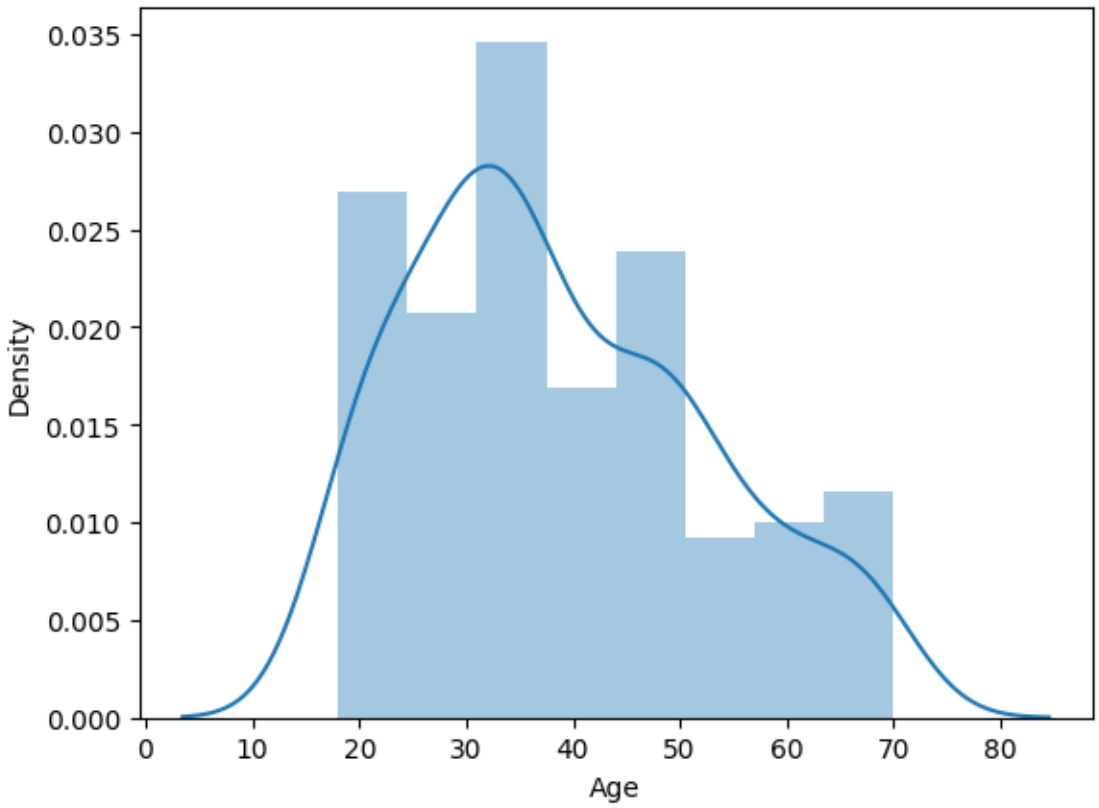
```
<ipython-input-13-7408ab7b8e79>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  df.corr().Age.sort_values(ascending =False)
```

[13]:
```
Age                     1.000000
Annual Income (k$)     -0.012398
CustomerID             -0.026763
Spending Score (1-100) -0.327227
Name: Age, dtype: float64
```

[14]:
```
sns.distplot(df.Age)
```

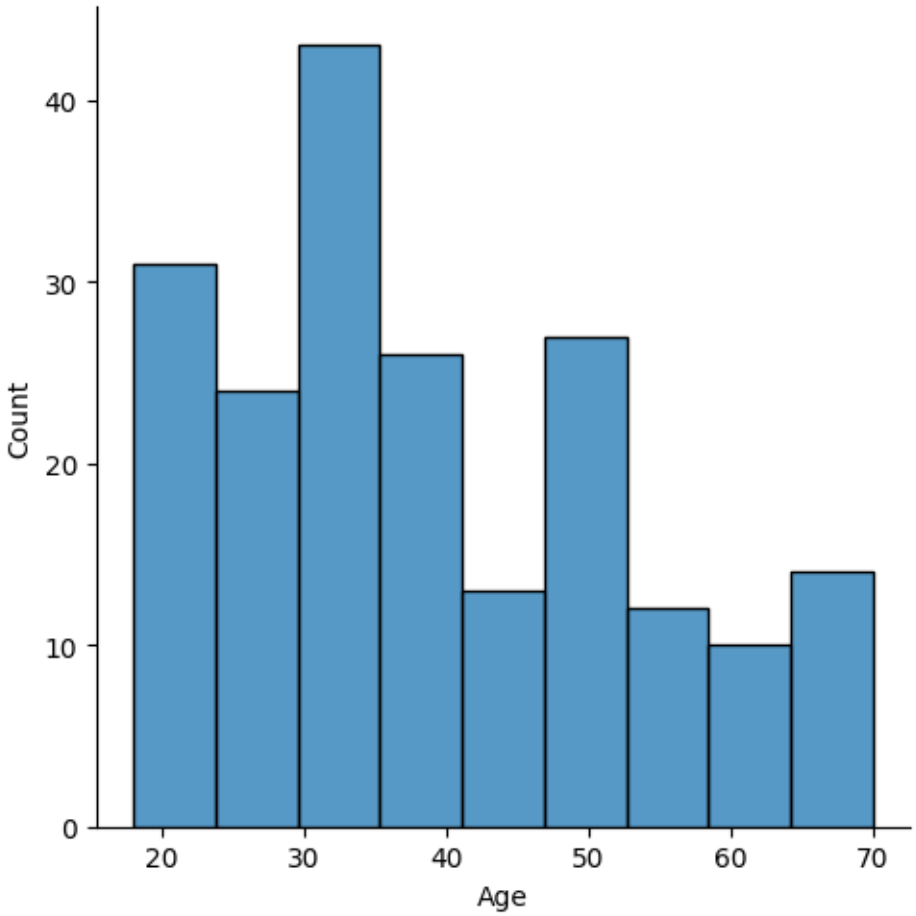
```
<ipython-input-14-b2378c9d8a20>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.Age)
```

[14]: <Axes: xlabel='Age', ylabel='Density'>



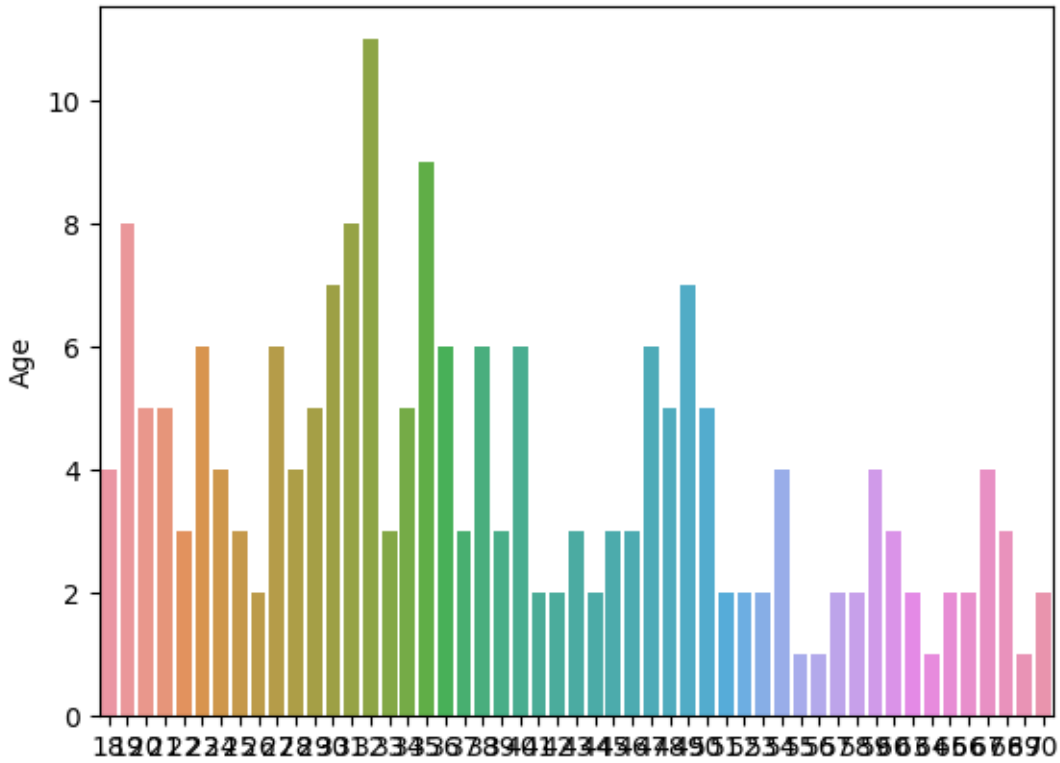
[15]:
```
sns.displot(df.Age)
```

[15]: <seaborn.axisgrid.FacetGrid at 0x788056589600>



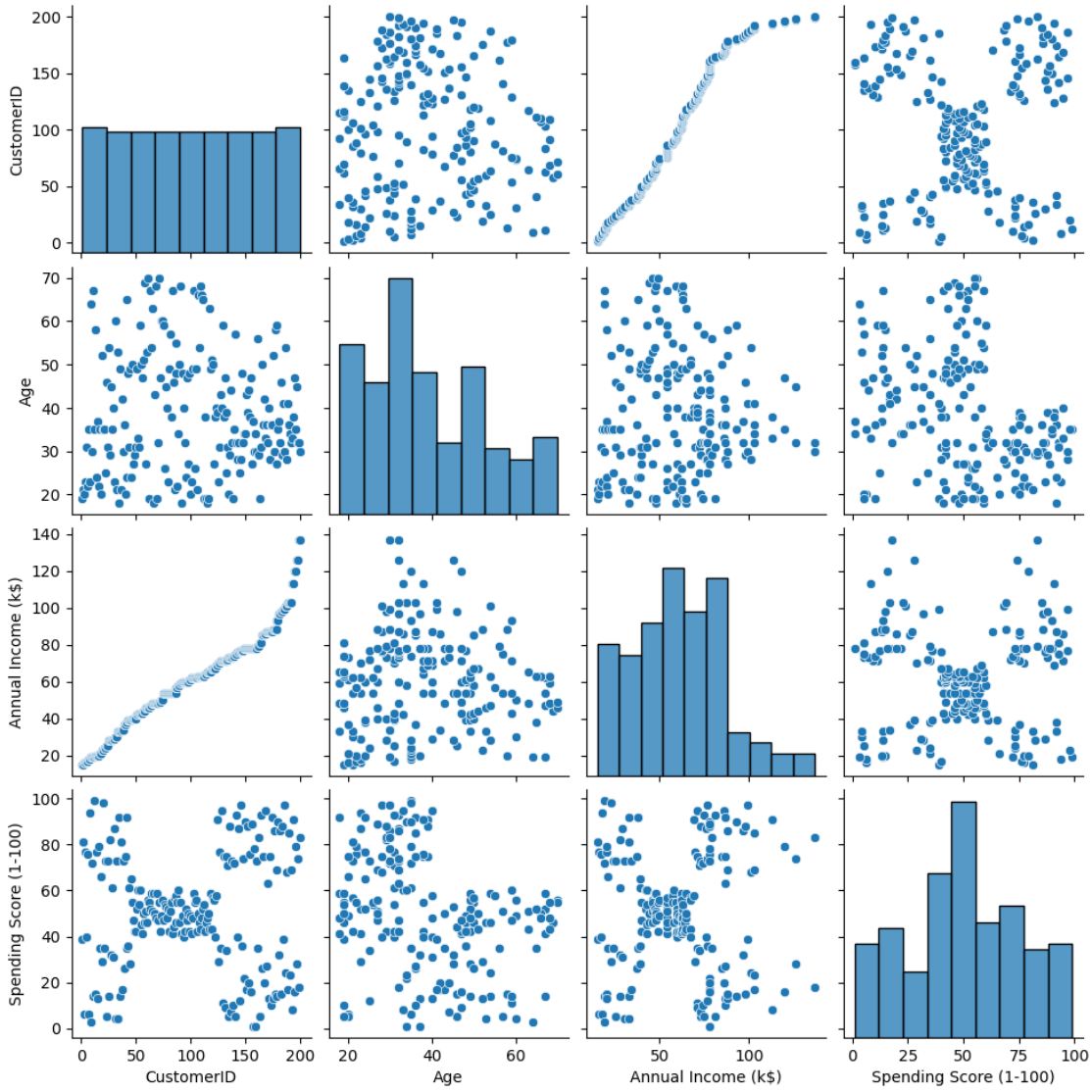
[16]:
```
sns.barplot(x =df.Age.value_counts().index,y =df.Age.value_counts() )
```

[16]:
```
<Axes: ylabel='Age'>
```



[17]:
```
sns.pairplot(df)
```

[17]: <seaborn.axisgrid.PairGrid at 0x788052261d80>



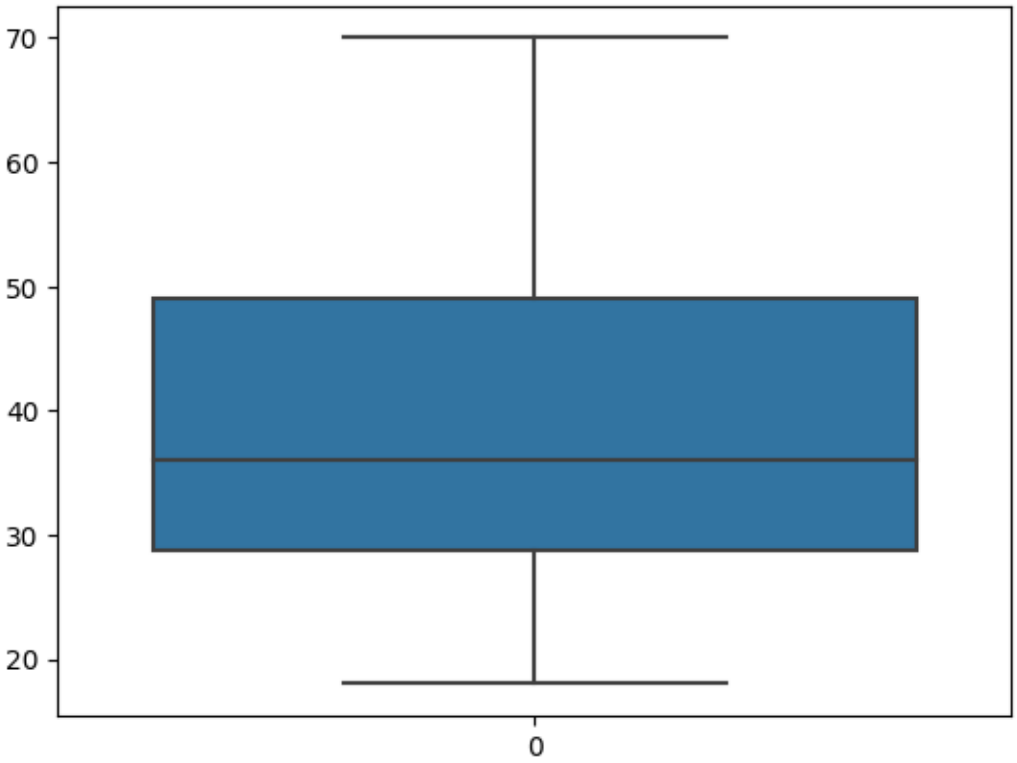
[18]:
```
sns.heatmap(df.corr(),annot=True)
```

```
<ipython-input-18-8df7bcac526d>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(df.corr(),annot=True)
```

[18]: <Axes: >

[19]:
```
sns.boxplot(df.Age)
```

[19]: <Axes: >



[20]:
```
from sklearn.preprocessing import LabelEncoder
```

[21]:
```
le = LabelEncoder()
```

[22]:
```
df.Age = le.fit_transform(df.Age)
```

[23]:
```
df.head()
```

[23]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	1	15	39
1	2	Male	3	15	81
2	3	Female	2	16	6
3	4	Female	5	16	77
4	5	Female	13	17	40

[24]:
```
df_main = pd.get_dummies(df,columns =['Age'])
df_main.head()
```

[24]:

	CustomerID	Gender	Annual Income (k$)	Spending Score (1-100)	Age_0	\
0	1	Male	15	39	0	
1	2	Male	15	81	0	
2	3	Female	16	6	0	

```
3           4  Female                  16                      77      0
4           5  Female                  17                      40      0

   Age_1  Age_2  Age_3  Age_4  Age_5  …  Age_41  Age_42  Age_43  Age_44  \
0      1      0      0      0      0  …       0       0       0       0
1      0      0      1      0      0  …       0       0       0       0
2      0      1      0      0      0  …       0       0       0       0
3      0      0      0      0      1  …       0       0       0       0
4      0      0      0      0      0  …       0       0       0       0

   Age_45  Age_46  Age_47  Age_48  Age_49  Age_50
0       0       0       0       0       0       0
1       0       0       0       0       0       0
2       0       0       0       0       0       0
3       0       0       0       0       0       0
4       0       0       0       0       0       0

[5 rows x 55 columns]
```

[25]:
```
df_main.corr()
```

```
<ipython-input-25-b764c75a6398>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  df_main.corr()
```

[25]:
```
                        CustomerID  Annual Income (k$)  \
CustomerID                1.000000            0.977548
Annual Income (k$)        0.977548            1.000000
Spending Score (1-100)    0.013835            0.009903
Age_0                    -0.058767           -0.050765
Age_1                    -0.012375           -0.027737
Age_2                    -0.114547           -0.115884
Age_3                    -0.133962           -0.132997
Age_4                    -0.136440           -0.137665
Age_5                    -0.129966           -0.127942
Age_6                    -0.126194           -0.116199
Age_7                    -0.001069           -0.013628
Age_8                    -0.018278           -0.009821
Age_9                     0.028938            0.017497
Age_10                    0.152794            0.134629
Age_11                    0.030232            0.018580
Age_12                    0.105792            0.113275
Age_13                   -0.090157           -0.094937
Age_14                    0.217477            0.245143
Age_15                    0.079441            0.093137
```

Age_16	0.130634	0.112705
Age_17	-0.118435	-0.115114
Age_18	0.156365	0.137205
Age_19	0.036693	0.020913
Age_20	0.097474	0.093574
Age_21	0.075879	0.057025
Age_22	0.010154	0.007429
Age_23	0.150579	0.155135
Age_24	0.002611	-0.002148
Age_25	0.030993	0.024054
Age_26	0.077466	0.057312
Age_27	-0.000356	0.041325
Age_28	-0.008194	-0.007348
Age_29	0.052291	0.064486
Age_30	-0.019137	-0.014424
Age_31	-0.078932	-0.069493
Age_32	-0.014145	-0.011980
Age_33	-0.021760	-0.019411
Age_34	-0.006093	-0.019411
Age_35	-0.094003	-0.080790
Age_36	-0.011135	-0.004417
Age_37	-0.016576	-0.009633
Age_38	0.074284	0.049894
Age_39	0.018278	0.015115
Age_40	-0.009574	-0.025165
Age_41	0.021651	0.025574
Age_42	-0.087991	-0.081142
Age_43	-0.016538	-0.015575
Age_44	-0.112347	-0.112451
Age_45	-0.042650	-0.038592
Age_46	0.013926	0.009360
Age_47	-0.087841	-0.082119
Age_48	-0.023868	-0.018339
Age_49	-0.052183	-0.044807
Age_50	-0.060058	-0.050100

	Spending Score (1-100)	Age_0	Age_1	Age_2	\
CustomerID	0.013835	-0.058767	-0.012375	-0.114547	
Annual Income (k$)	0.009903	-0.050765	-0.027737	-0.115884	
Spending Score (1-100)	1.000000	0.054350	-0.082810	-0.062164	
Age_0	0.054350	1.000000	-0.029161	-0.022875	
Age_1	-0.082810	-0.029161	1.000000	-0.032686	
Age_2	-0.062164	-0.022875	-0.032686	1.000000	
Age_3	0.103193	-0.022875	-0.032686	-0.025641	
Age_4	0.094856	-0.017629	-0.025190	-0.019760	
Age_5	0.089665	-0.025123	-0.035898	-0.028161	
Age_6	0.118128	-0.020408	-0.029161	-0.022875	

```
Age_7                          -0.050462 -0.017629 -0.025190 -0.019760
Age_8                           0.016777 -0.014358 -0.020515 -0.016093
Age_9                           0.069183 -0.025123 -0.035898 -0.028161
Age_10                          0.109810 -0.020408 -0.029161 -0.022875
Age_11                          0.164114 -0.022875 -0.032686 -0.025641
Age_12                          0.222435 -0.027206 -0.038874 -0.030496
Age_13                          0.108366 -0.029161 -0.041667 -0.032686
Age_14                          0.147977 -0.034464 -0.049245 -0.038631
Age_15                          0.019802 -0.017629 -0.025190 -0.019760
Age_16                         -0.068381 -0.022875 -0.032686 -0.025641
Age_17                          0.115357 -0.031010 -0.044310 -0.034759
Age_18                          0.015703 -0.025123 -0.035898 -0.028161
Age_19                         -0.167036 -0.017629 -0.025190 -0.019760
Age_20                          0.088527 -0.025123 -0.035898 -0.028161
Age_21                          0.165120 -0.017629 -0.025190 -0.019760
Age_22                         -0.018434 -0.025123 -0.035898 -0.028161
Age_23                         -0.086618 -0.014358 -0.020515 -0.016093
Age_24                         -0.123684 -0.014358 -0.020515 -0.016093
Age_25                         -0.077610 -0.017629 -0.025190 -0.019760
Age_26                         -0.143193 -0.014358 -0.020515 -0.016093
Age_27                         -0.060044 -0.017629 -0.025190 -0.019760
Age_28                         -0.138292 -0.017629 -0.025190 -0.019760
Age_29                         -0.148152 -0.025123 -0.035898 -0.028161
Age_30                         -0.052218 -0.022875 -0.032686 -0.025641
Age_31                         -0.055345 -0.027206 -0.038874 -0.030496
Age_32                         -0.027352 -0.022875 -0.032686 -0.025641
Age_33                         -0.014436 -0.014358 -0.020515 -0.016093
Age_34                         -0.113930 -0.014358 -0.020515 -0.016093
Age_35                         -0.098323 -0.014358 -0.020515 -0.016093
Age_36                         -0.080139 -0.020408 -0.029161 -0.022875
Age_37                          0.021466 -0.010127 -0.014470 -0.011351
Age_38                         -0.041830 -0.010127 -0.014470 -0.011351
Age_39                         -0.086618 -0.014358 -0.020515 -0.016093
Age_40                         -0.137340 -0.014358 -0.020515 -0.016093
Age_41                         -0.095390 -0.020408 -0.029161 -0.022875
Age_42                         -0.066431 -0.017629 -0.025190 -0.019760
Age_43                         -0.012485 -0.014358 -0.020515 -0.016093
Age_44                         -0.129894 -0.010127 -0.014470 -0.011351
Age_45                         -0.026141 -0.014358 -0.020515 -0.016093
Age_46                         -0.004682 -0.014358 -0.020515 -0.016093
Age_47                         -0.048250 -0.020408 -0.029161 -0.022875
Age_48                         -0.007346 -0.017629 -0.025190 -0.019760
Age_49                         -0.011558 -0.010127 -0.014470 -0.011351
Age_50                          0.020679 -0.014358 -0.020515 -0.016093

                                   Age_3     Age_4     Age_5     Age_6  …    Age_41  \
CustomerID                  -0.133962 -0.136440 -0.129966 -0.126194  …  0.021651
```

Annual Income (k$)	-0.132997	-0.137665	-0.127942	-0.116199	…	0.025574
Spending Score (1-100)	0.103193	0.094856	0.089665	0.118128	…	-0.095390
Age_0	-0.022875	-0.017629	-0.025123	-0.020408	…	-0.020408
Age_1	-0.032686	-0.025190	-0.035898	-0.029161	…	-0.029161
Age_2	-0.025641	-0.019760	-0.028161	-0.022875	…	-0.022875
Age_3	1.000000	-0.019760	-0.028161	-0.022875	…	-0.022875
Age_4	-0.019760	1.000000	-0.021702	-0.017629	…	-0.017629
Age_5	-0.028161	-0.021702	1.000000	-0.025123	…	-0.025123
Age_6	-0.022875	-0.017629	-0.025123	1.000000	…	-0.020408
Age_7	-0.019760	-0.015228	-0.021702	-0.017629	…	-0.017629
Age_8	-0.016093	-0.012403	-0.017675	-0.014358	…	-0.014358
Age_9	-0.028161	-0.021702	-0.030928	-0.025123	…	-0.025123
Age_10	-0.022875	-0.017629	-0.025123	-0.020408	…	-0.020408
Age_11	-0.025641	-0.019760	-0.028161	-0.022875	…	-0.022875
Age_12	-0.030496	-0.023502	-0.033492	-0.027206	…	-0.027206
Age_13	-0.032686	-0.025190	-0.035898	-0.029161	…	-0.029161
Age_14	-0.038631	-0.029771	-0.042427	-0.034464	…	-0.034464
Age_15	-0.019760	-0.015228	-0.021702	-0.017629	…	-0.017629
Age_16	-0.025641	-0.019760	-0.028161	-0.022875	…	-0.022875
Age_17	-0.034759	-0.026787	-0.038175	-0.031010	…	-0.031010
Age_18	-0.028161	-0.021702	-0.030928	-0.025123	…	-0.025123
Age_19	-0.019760	-0.015228	-0.021702	-0.017629	…	-0.017629
Age_20	-0.028161	-0.021702	-0.030928	-0.025123	…	-0.025123
Age_21	-0.019760	-0.015228	-0.021702	-0.017629	…	-0.017629
Age_22	-0.028161	-0.021702	-0.030928	-0.025123	…	-0.025123
Age_23	-0.016093	-0.012403	-0.017675	-0.014358	…	-0.014358
Age_24	-0.016093	-0.012403	-0.017675	-0.014358	…	-0.014358
Age_25	-0.019760	-0.015228	-0.021702	-0.017629	…	-0.017629
Age_26	-0.016093	-0.012403	-0.017675	-0.014358	…	-0.014358
Age_27	-0.019760	-0.015228	-0.021702	-0.017629	…	-0.017629
Age_28	-0.019760	-0.015228	-0.021702	-0.017629	…	-0.017629
Age_29	-0.028161	-0.021702	-0.030928	-0.025123	…	-0.025123
Age_30	-0.025641	-0.019760	-0.028161	-0.022875	…	-0.022875
Age_31	-0.030496	-0.023502	-0.033492	-0.027206	…	-0.027206
Age_32	-0.025641	-0.019760	-0.028161	-0.022875	…	-0.022875
Age_33	-0.016093	-0.012403	-0.017675	-0.014358	…	-0.014358
Age_34	-0.016093	-0.012403	-0.017675	-0.014358	…	-0.014358
Age_35	-0.016093	-0.012403	-0.017675	-0.014358	…	-0.014358
Age_36	-0.022875	-0.017629	-0.025123	-0.020408	…	-0.020408
Age_37	-0.011351	-0.008748	-0.012467	-0.010127	…	-0.010127
Age_38	-0.011351	-0.008748	-0.012467	-0.010127	…	-0.010127
Age_39	-0.016093	-0.012403	-0.017675	-0.014358	…	-0.014358
Age_40	-0.016093	-0.012403	-0.017675	-0.014358	…	-0.014358
Age_41	-0.022875	-0.017629	-0.025123	-0.020408	…	1.000000
Age_42	-0.019760	-0.015228	-0.021702	-0.017629	…	-0.017629
Age_43	-0.016093	-0.012403	-0.017675	-0.014358	…	-0.014358
Age_44	-0.011351	-0.008748	-0.012467	-0.010127	…	-0.010127

```
Age_45                  -0.016093 -0.012403 -0.017675 -0.014358  … -0.014358
Age_46                  -0.016093 -0.012403 -0.017675 -0.014358  … -0.014358
Age_47                  -0.022875 -0.017629 -0.025123 -0.020408  … -0.020408
Age_48                  -0.019760 -0.015228 -0.021702 -0.017629  … -0.017629
Age_49                  -0.011351 -0.008748 -0.012467 -0.010127  … -0.010127
Age_50                  -0.016093 -0.012403 -0.017675 -0.014358  … -0.014358

                           Age_42    Age_43    Age_44    Age_45    Age_46  \
CustomerID              -0.087991 -0.016538 -0.112347 -0.042650  0.013926
Annual Income (k$)      -0.081142 -0.015575 -0.112451 -0.038592  0.009360
Spending Score (1-100)  -0.066431 -0.012485 -0.129894 -0.026141 -0.004682
Age_0                   -0.017629 -0.014358 -0.010127 -0.014358 -0.014358
Age_1                   -0.025190 -0.020515 -0.014470 -0.020515 -0.020515
Age_2                   -0.019760 -0.016093 -0.011351 -0.016093 -0.016093
Age_3                   -0.019760 -0.016093 -0.011351 -0.016093 -0.016093
Age_4                   -0.015228 -0.012403 -0.008748 -0.012403 -0.012403
Age_5                   -0.021702 -0.017675 -0.012467 -0.017675 -0.017675
Age_6                   -0.017629 -0.014358 -0.010127 -0.014358 -0.014358
Age_7                   -0.015228 -0.012403 -0.008748 -0.012403 -0.012403
Age_8                   -0.012403 -0.010101 -0.007125 -0.010101 -0.010101
Age_9                   -0.021702 -0.017675 -0.012467 -0.017675 -0.017675
Age_10                  -0.017629 -0.014358 -0.010127 -0.014358 -0.014358
Age_11                  -0.019760 -0.016093 -0.011351 -0.016093 -0.016093
Age_12                  -0.023502 -0.019140 -0.013500 -0.019140 -0.019140
Age_13                  -0.025190 -0.020515 -0.014470 -0.020515 -0.020515
Age_14                  -0.029771 -0.024246 -0.017102 -0.024246 -0.024246
Age_15                  -0.015228 -0.012403 -0.008748 -0.012403 -0.012403
Age_16                  -0.019760 -0.016093 -0.011351 -0.016093 -0.016093
Age_17                  -0.026787 -0.021817 -0.015388 -0.021817 -0.021817
Age_18                  -0.021702 -0.017675 -0.012467 -0.017675 -0.017675
Age_19                  -0.015228 -0.012403 -0.008748 -0.012403 -0.012403
Age_20                  -0.021702 -0.017675 -0.012467 -0.017675 -0.017675
Age_21                  -0.015228 -0.012403 -0.008748 -0.012403 -0.012403
Age_22                  -0.021702 -0.017675 -0.012467 -0.017675 -0.017675
Age_23                  -0.012403 -0.010101 -0.007125 -0.010101 -0.010101
Age_24                  -0.012403 -0.010101 -0.007125 -0.010101 -0.010101
Age_25                  -0.015228 -0.012403 -0.008748 -0.012403 -0.012403
Age_26                  -0.012403 -0.010101 -0.007125 -0.010101 -0.010101
Age_27                  -0.015228 -0.012403 -0.008748 -0.012403 -0.012403
Age_28                  -0.015228 -0.012403 -0.008748 -0.012403 -0.012403
Age_29                  -0.021702 -0.017675 -0.012467 -0.017675 -0.017675
Age_30                  -0.019760 -0.016093 -0.011351 -0.016093 -0.016093
Age_31                  -0.023502 -0.019140 -0.013500 -0.019140 -0.019140
Age_32                  -0.019760 -0.016093 -0.011351 -0.016093 -0.016093
Age_33                  -0.012403 -0.010101 -0.007125 -0.010101 -0.010101
Age_34                  -0.012403 -0.010101 -0.007125 -0.010101 -0.010101
Age_35                  -0.012403 -0.010101 -0.007125 -0.010101 -0.010101
```

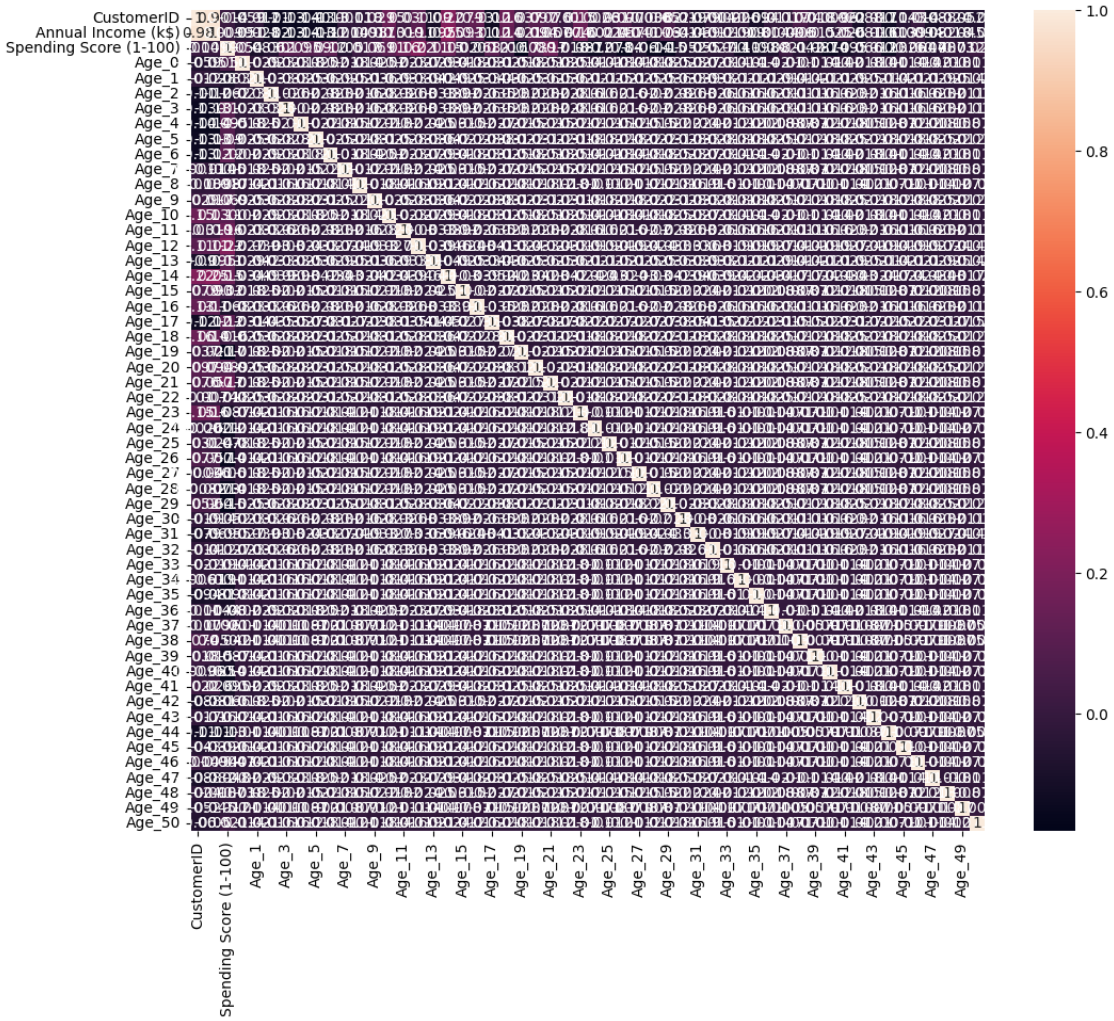
```
Age_36                 -0.017629 -0.014358 -0.010127 -0.014358 -0.014358
Age_37                 -0.008748 -0.007125 -0.005025 -0.007125 -0.007125
Age_38                 -0.008748 -0.007125 -0.005025 -0.007125 -0.007125
Age_39                 -0.012403 -0.010101 -0.007125 -0.010101 -0.010101
Age_40                 -0.012403 -0.010101 -0.007125 -0.010101 -0.010101
Age_41                 -0.017629 -0.014358 -0.010127 -0.014358 -0.014358
Age_42                  1.000000 -0.012403 -0.008748 -0.012403 -0.012403
Age_43                 -0.012403  1.000000 -0.007125 -0.010101 -0.010101
Age_44                 -0.008748 -0.007125  1.000000 -0.007125 -0.007125
Age_45                 -0.012403 -0.010101 -0.007125  1.000000 -0.010101
Age_46                 -0.012403 -0.010101 -0.007125 -0.010101  1.000000
Age_47                 -0.017629 -0.014358 -0.010127 -0.014358 -0.014358
Age_48                 -0.015228 -0.012403 -0.008748 -0.012403 -0.012403
Age_49                 -0.008748 -0.007125 -0.005025 -0.007125 -0.007125
Age_50                 -0.012403 -0.010101 -0.007125 -0.010101 -0.010101

                          Age_47    Age_48    Age_49    Age_50
CustomerID             -0.087841 -0.023868 -0.052183 -0.060058
Annual Income (k$)     -0.082119 -0.018339 -0.044807 -0.050100
Spending Score (1-100) -0.048250 -0.007346 -0.011558  0.020679
Age_0                  -0.020408 -0.017629 -0.010127 -0.014358
Age_1                  -0.029161 -0.025190 -0.014470 -0.020515
Age_2                  -0.022875 -0.019760 -0.011351 -0.016093
Age_3                  -0.022875 -0.019760 -0.011351 -0.016093
Age_4                  -0.017629 -0.015228 -0.008748 -0.012403
Age_5                  -0.025123 -0.021702 -0.012467 -0.017675
Age_6                  -0.020408 -0.017629 -0.010127 -0.014358
Age_7                  -0.017629 -0.015228 -0.008748 -0.012403
Age_8                  -0.014358 -0.012403 -0.007125 -0.010101
Age_9                  -0.025123 -0.021702 -0.012467 -0.017675
Age_10                 -0.020408 -0.017629 -0.010127 -0.014358
Age_11                 -0.022875 -0.019760 -0.011351 -0.016093
Age_12                 -0.027206 -0.023502 -0.013500 -0.019140
Age_13                 -0.029161 -0.025190 -0.014470 -0.020515
Age_14                 -0.034464 -0.029771 -0.017102 -0.024246
Age_15                 -0.017629 -0.015228 -0.008748 -0.012403
Age_16                 -0.022875 -0.019760 -0.011351 -0.016093
Age_17                 -0.031010 -0.026787 -0.015388 -0.021817
Age_18                 -0.025123 -0.021702 -0.012467 -0.017675
Age_19                 -0.017629 -0.015228 -0.008748 -0.012403
Age_20                 -0.025123 -0.021702 -0.012467 -0.017675
Age_21                 -0.017629 -0.015228 -0.008748 -0.012403
Age_22                 -0.025123 -0.021702 -0.012467 -0.017675
Age_23                 -0.014358 -0.012403 -0.007125 -0.010101
Age_24                 -0.014358 -0.012403 -0.007125 -0.010101
Age_25                 -0.017629 -0.015228 -0.008748 -0.012403
Age_26                 -0.014358 -0.012403 -0.007125 -0.010101
```

```
Age_27                -0.017629 -0.015228 -0.008748 -0.012403
Age_28                -0.017629 -0.015228 -0.008748 -0.012403
Age_29                -0.025123 -0.021702 -0.012467 -0.017675
Age_30                -0.022875 -0.019760 -0.011351 -0.016093
Age_31                -0.027206 -0.023502 -0.013500 -0.019140
Age_32                -0.022875 -0.019760 -0.011351 -0.016093
Age_33                -0.014358 -0.012403 -0.007125 -0.010101
Age_34                -0.014358 -0.012403 -0.007125 -0.010101
Age_35                -0.014358 -0.012403 -0.007125 -0.010101
Age_36                -0.020408 -0.017629 -0.010127 -0.014358
Age_37                -0.010127 -0.008748 -0.005025 -0.007125
Age_38                -0.010127 -0.008748 -0.005025 -0.007125
Age_39                -0.014358 -0.012403 -0.007125 -0.010101
Age_40                -0.014358 -0.012403 -0.007125 -0.010101
Age_41                -0.020408 -0.017629 -0.010127 -0.014358
Age_42                -0.017629 -0.015228 -0.008748 -0.012403
Age_43                -0.014358 -0.012403 -0.007125 -0.010101
Age_44                -0.010127 -0.008748 -0.005025 -0.007125
Age_45                -0.014358 -0.012403 -0.007125 -0.010101
Age_46                -0.014358 -0.012403 -0.007125 -0.010101
Age_47                 1.000000 -0.017629 -0.010127 -0.014358
Age_48                -0.017629  1.000000 -0.008748 -0.012403
Age_49                -0.010127 -0.008748  1.000000 -0.007125
Age_50                -0.014358 -0.012403 -0.007125  1.000000

[54 rows x 54 columns]
```

[44]:
```
plt.figure(figsize=(12,10))
sns.heatmap(df_main.corr(),annot =True)
```

```
<ipython-input-44-65b4d4cae710>:2: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future
version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
```

[44]: <Axes: >



[27]:
```
df_main.head()
```

[27]:

	CustomerID	Gender	Annual Income (k$)	Spending Score (1-100)	Age_0	\
0	1	Male	15	39	0	
1	2	Male	15	81	0	
2	3	Female	16	6	0	
3	4	Female	16	77	0	
4	5	Female	17	40	0	

	Age_1	Age_2	Age_3	Age_4	Age_5	…	Age_41	Age_42	Age_43	Age_44	\
0	1	0	0	0	0	…	0	0	0	0	
1	0	0	1	0	0	…	0	0	0	0	
2	0	1	0	0	0	…	0	0	0	0	
3	0	0	0	0	1	…	0	0	0	0	
4	0	0	0	0	0	…	0	0	0	0	

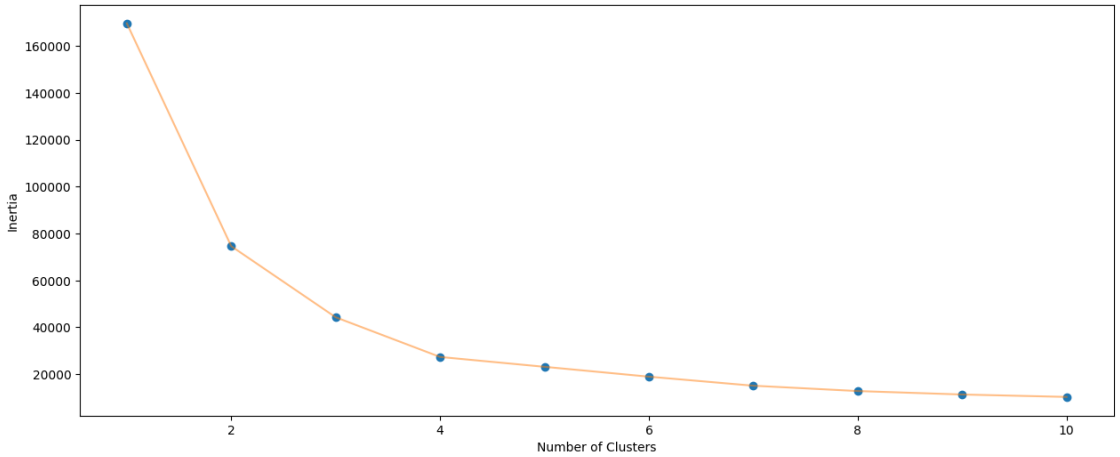
```
   Age_45  Age_46  Age_47  Age_48  Age_49  Age_50
0       0       0       0       0       0       0
1       0       0       0       0       0       0
2       0       0       0       0       0       0
3       0       0       0       0       0       0
4       0       0       0       0       0       0

[5 rows x 55 columns]
```

0.2.2 Using K-means

[28]:
```
X1 = df[['Age' , 'Spending Score (1-100)']].iloc[: , :].values
inertia = []
for n in range(1 , 11):
    algorithm = (KMeans(n_clusters = n ,init='k-means++', n_init = 10␣
 ↪,max_iter=300,
                        tol=0.0001,  random_state= 111  , algorithm='elkan') )
    algorithm.fit(X1)
    inertia.append(algorithm.inertia_)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1373:
RuntimeWarning: algorithm='elkan' doesn't make sense for a single cluster. Using
'lloyd' instead.
  warnings.warn(
```

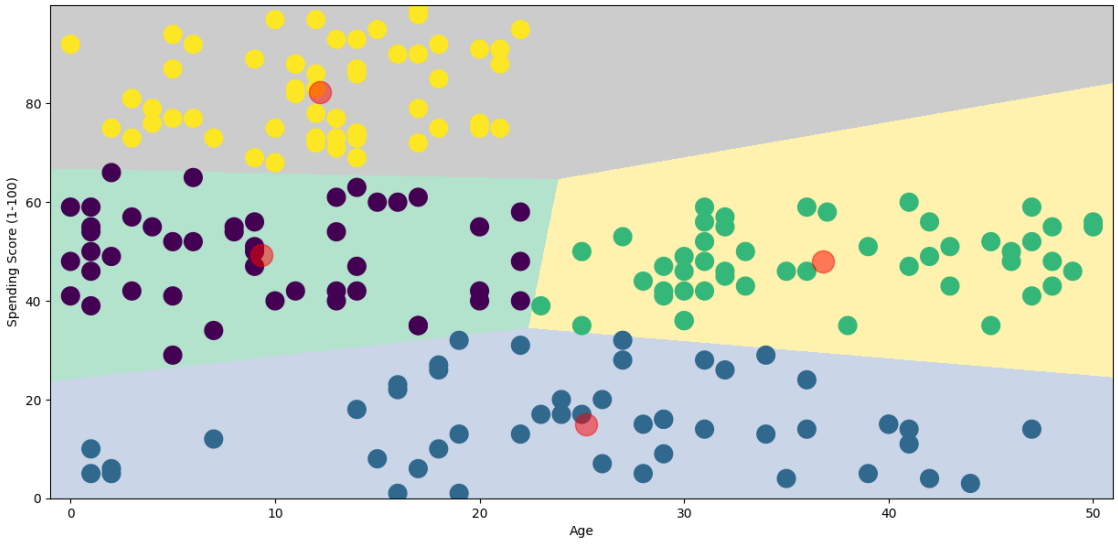
[29]:
```
plt.figure(1 , figsize = (15 ,6))
plt.plot(np.arange(1 , 11) , inertia , 'o')
plt.plot(np.arange(1 , 11) , inertia , '-' , alpha = 0.5)
plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
plt.show()
```



[30]:
```
algorithm = (KMeans(n_clusters = 4 ,init='k-means++', n_init = 10 ,max_iter=300,
                    tol=0.0001,  random_state= 111  , algorithm='elkan') )
algorithm.fit(X1)
labels1 = algorithm.labels_
centroids1 = algorithm.cluster_centers_
```

[31]:
```
h = 0.02
x_min, x_max = X1[:, 0].min() - 1, X1[:, 0].max() + 1
y_min, y_max = X1[:, 1].min() - 1, X1[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
Z = algorithm.predict(np.c_[xx.ravel(), yy.ravel()])
```

[32]:
```
plt.figure(1 , figsize = (15 , 7) )
plt.clf()
Z = Z.reshape(xx.shape)
plt.imshow(Z , interpolation='nearest',
           extent=(xx.min(), xx.max(), yy.min(), yy.max()),
           cmap = plt.cm.Pastel2, aspect = 'auto', origin='lower')

plt.scatter( x = 'Age' ,y = 'Spending Score (1-100)' , data = df , c = labels1 ,
            s = 200 )
plt.scatter(x = centroids1[: , 0] , y =  centroids1[: , 1] , s = 300 , c =␣
 ↪'red' , alpha = 0.5)
plt.ylabel('Spending Score (1-100)') , plt.xlabel('Age')
plt.show()
```

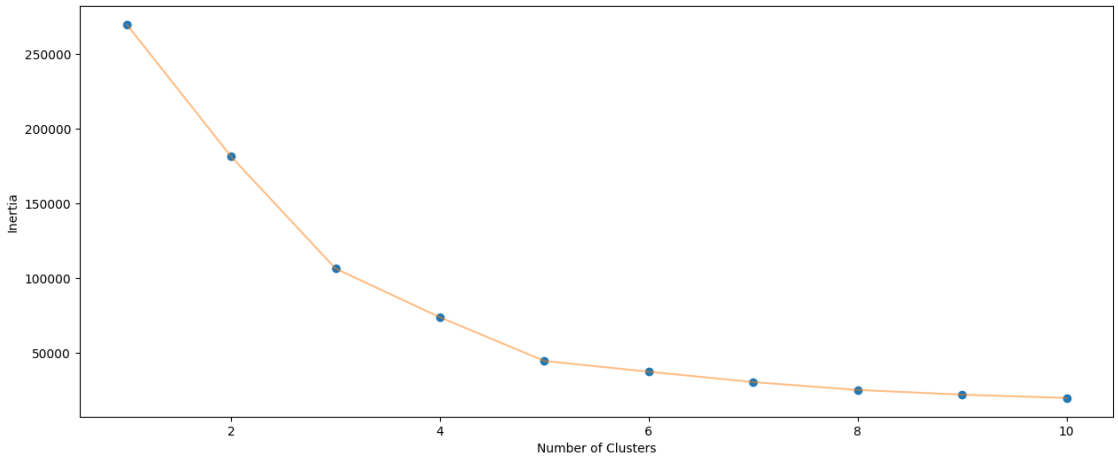


[33]:
```
X2 = df[['Annual Income (k$)' , 'Spending Score (1-100)']].iloc[: , :].values
inertia = []
for n in range(1 , 11):
```

```
    algorithm = (KMeans(n_clusters = n ,init='k-means++', n_init = 10␣
↪,max_iter=300,
                        tol=0.0001,  random_state= 111  , algorithm='elkan') )
    algorithm.fit(X2)
    inertia.append(algorithm.inertia_)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1373:
RuntimeWarning: algorithm='elkan' doesn't make sense for a single cluster. Using
'lloyd' instead.
  warnings.warn(
```

[34]:
```
plt.figure(1 , figsize = (15 ,6))
plt.plot(np.arange(1 , 11) , inertia , 'o')
plt.plot(np.arange(1 , 11) , inertia , '-' , alpha = 0.5)
plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
plt.show()
```

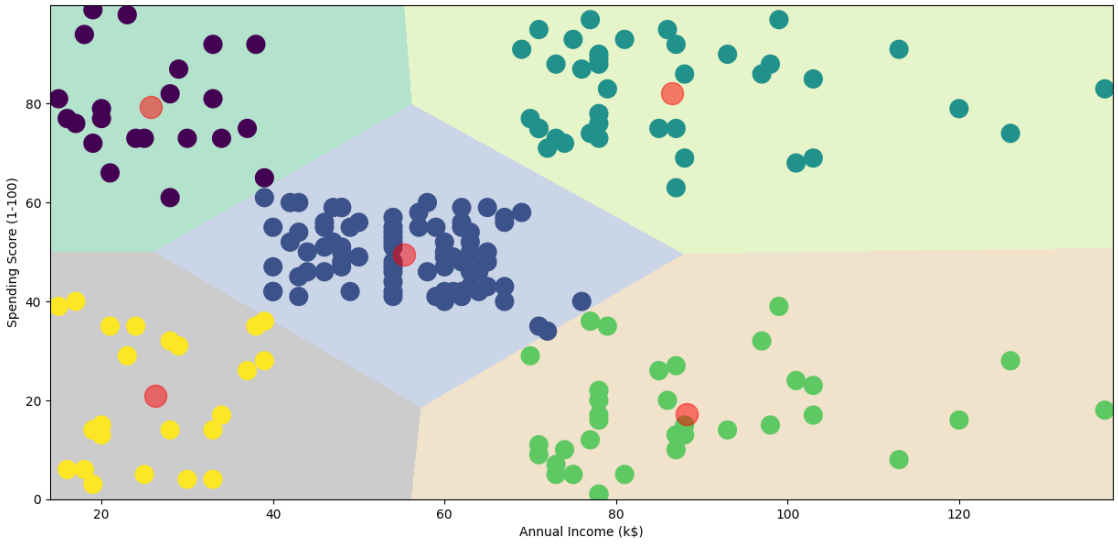


[35]:
```
algorithm = (KMeans(n_clusters = 5 ,init='k-means++', n_init = 10 ,max_iter=300,
                        tol=0.0001,  random_state= 111  , algorithm='elkan') )
algorithm.fit(X2)
labels2 = algorithm.labels_
centroids2 = algorithm.cluster_centers_
```

[36]:
```
h = 0.02
x_min, x_max = X2[:, 0].min() - 1, X2[:, 0].max() + 1
y_min, y_max = X2[:, 1].min() - 1, X2[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
Z2 = algorithm.predict(np.c_[xx.ravel(), yy.ravel()])
```

[37]:
```
plt.figure(1 , figsize = (15 , 7) )
plt.clf()
```

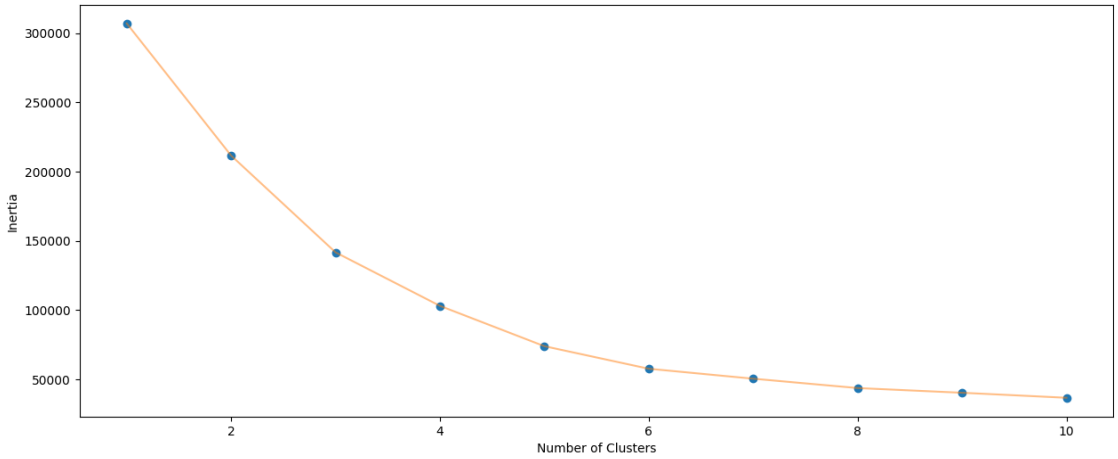
```
Z2 = Z2.reshape(xx.shape)
plt.imshow(Z2 , interpolation='nearest',
           extent=(xx.min(), xx.max(), yy.min(), yy.max()),
           cmap = plt.cm.Pastel2, aspect = 'auto', origin='lower')

plt.scatter( x = 'Annual Income (k$)' ,y = 'Spending Score (1-100)' , data = df␣
 ↪, c = labels2 ,
            s = 200 )
plt.scatter(x = centroids2[: , 0] , y =  centroids2[: , 1] , s = 300 , c =␣
 ↪'red' , alpha = 0.5)
plt.ylabel('Spending Score (1-100)') , plt.xlabel('Annual Income (k$)')
plt.show()
```



[38]:
```
X3 = df[['Age' , 'Annual Income (k$)' ,'Spending Score (1-100)']].iloc[: , :].
 ↪values
inertia = []
for n in range(1 , 11):
    algorithm = (KMeans(n_clusters = n ,init='k-means++', n_init = 10␣
 ↪,max_iter=300,
                        tol=0.0001,  random_state= 111  , algorithm='elkan') )
    algorithm.fit(X3)
    inertia.append(algorithm.inertia_)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1373:
RuntimeWarning: algorithm='elkan' doesn't make sense for a single cluster. Using
'lloyd' instead.
  warnings.warn(
```

[39]:
```
plt.figure(1 , figsize = (15 ,6))
plt.plot(np.arange(1 , 11) , inertia , 'o')
plt.plot(np.arange(1 , 11) , inertia , '-' , alpha = 0.5)
plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
plt.show()
```



[45]:
```
algorithm = (KMeans(n_clusters = 6 ,init='k-means++', n_init = 10 ,max_iter=300,
                    tol=0.0001,  random_state= 111  , algorithm='elkan') )
algorithm.fit(X3)
labels3 = algorithm.labels_
centroids3 = algorithm.cluster_centers_
```

[46]:
```
df['label3'] =  labels3
trace1 = go.Scatter3d(
    x= df['Age'],
    y= df['Spending Score (1-100)'],
    z= df['Annual Income (k$)'],
    mode='markers',
     marker=dict(
        color = df['label3'],
        size= 20,
        line=dict(
            color= df['label3'],
            width= 12
        ),
        opacity=0.8
     )
)
data = [trace1]
layout = go.Layout(
#     margin=dict(
```

```
#         l=0,
#         r=0,
#         b=0,
#         t=0
#     )
    title= 'Clusters',
    scene = dict(
            xaxis = dict(title  = 'Age'),
            yaxis = dict(title  = 'Spending Score'),
            zaxis = dict(title  = 'Annual Income')
        )
)
fig = go.Figure(data=data, layout=layout)
py.offline.iplot(fig)
```

[ ]: