```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('/content/drive/MyDrive/Dataset_ML/Mall_Customers.csv')
df
```

	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
...	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

```
df.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

```
df.tail()
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

```
df.columns
```

```
Index(['CustomerID', 'Genre', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

```
df.isna().sum()
```

```
CustomerID                0
Genre                     0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

```
x=df.iloc[:,[3,4]]
x
```

	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40
...	...	...
195	120	79
196	126	28
197	126	74
198	137	18
199	137	83

200 rows × 2 columns

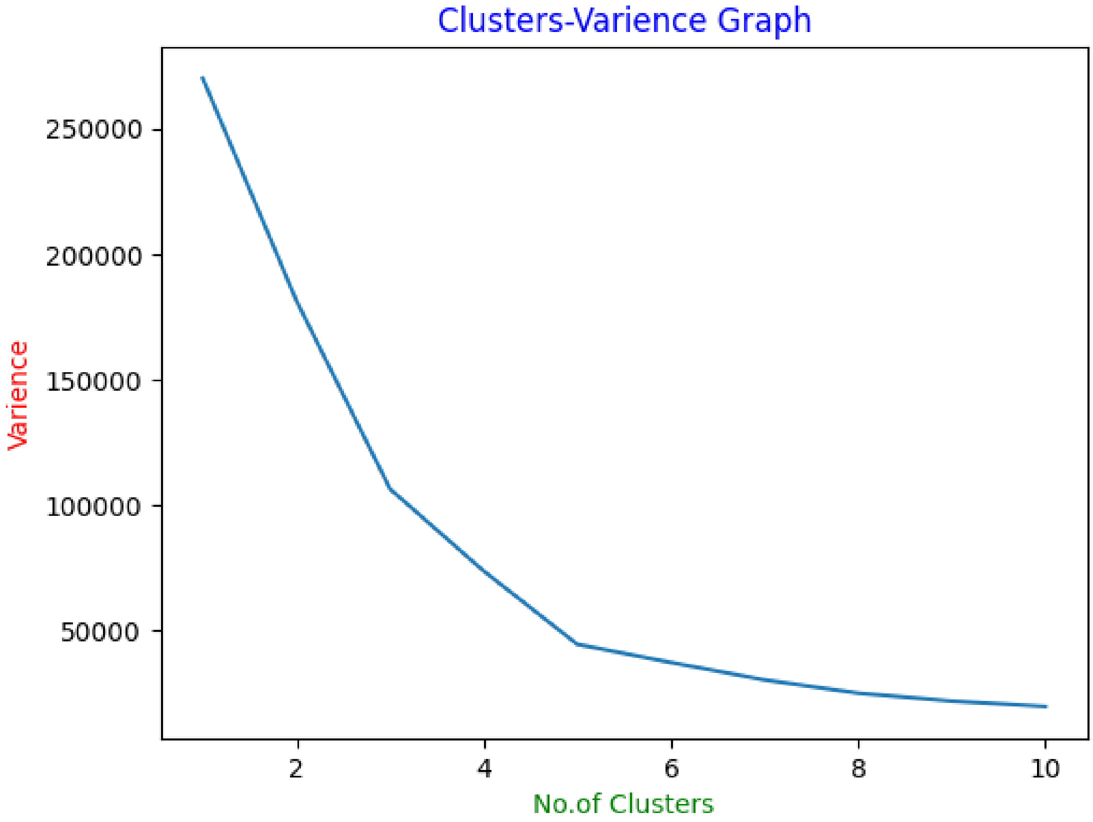
```
#Number of clusters
from sklearn.cluster import KMeans
WCSS=[]  #Varience
for i in range(1,11):  #cluster
  kmeans=KMeans(n_clusters=i,init='k-means++',random_state=42)
  kmeans.fit(x)
  WCSS.append(kmeans.inertia_)
WCSS
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
```

```
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
  warnings.warn(
[269981.28,
 181363.59595959593,
 106348.37306211122,
 73679.78903948836,
 44448.4554479337,
 37233.814510710006,
 30259.65720728547,
 25011.839349156588,
 21862.092672182895,
 19672.072849014323]
```

```
#ploting the cluster-varience graph
plt.plot(range(1,11),WCSS)
plt.xlabel('No.of Clusters',color='g')
plt.ylabel('Varience',color='r')
plt.title('Clusters-Varience Graph',color='blue')
```

```
Text(0.5, 1.0, 'Clusters-Varience Graph')
```



```
#Number of Clusters is 5
kmeans=KMeans(n_clusters=5,init='k-means++',random_state=42)
kmeans.fit(x)
```

```
y_pred=kmeans.predict(x)
y_pred
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The
  warnings.warn(
array([2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 0,
       2, 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, 0, 0, 0, 0, 4, 1, 4, 0, 4, 1, 4, 1, 4,
       0, 4, 1, 4, 1, 4, 1, 4, 1, 4, 0, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4,
       1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4,
       1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4,
       1, 4], dtype=int32)
```

```
x['cluster']=y_pred
x
```

```
<ipython-input-11-84aa8cab0bce>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gui
  x['cluster']=y_pred
```

	Annual Income (k$)	Spending Score (1-100)	cluster
0	15	39	2
1	15	81	3
2	16	6	2
3	16	77	3
4	17	40	2
...	...	...	...
195	120	79	4
196	126	28	1
197	126	74	4
198	137	18	1
199	137	83	4

200 rows × 3 columns

```
z=x.iloc[:,:-1]
y=x.iloc[:,-1]
y
```

```
0      2
1      3
2      2
3      3
4      2
      ..
195    4
```

```
196    1
197    4
198    1
199    4
Name: cluster, Length: 200, dtype: int32
```

```
from sklearn.model_selection import train_test_split
z_train,z_test,y_train,y_test=train_test_split(z,y,test_size=.30,random_state=42)
z_train
```

	Annual Income (k$)	Spending Score (1-100)
169	87	63
97	60	50
31	30	73
12	20	15
35	33	81
...	...	...
106	63	50
14	20	13
92	60	49
179	93	90
102	62	59

140 rows × 2 columns

```
z_test
```

124	70	29
16	21	35
148	78	22
93	60	40
65	48	59
60	46	56
84	54	57
67	48	48
125	70	77
132	72	34
9	19	72
18	23	29
55	43	41
75	54	54
150	78	17
104	62	56
135	73	88
137	73	73
164	85	26
76	54	53
79	54	42
197	126	74
38	37	26
24	28	14
122	69	58
195	120	79
29	29	87
19	23	98
143	76	87
86	57	58
114	65	48
173	87	92
5	17	76
126	71	35
117	65	59
73	50	56

140	75	5
98	61	42
172	87	10
96	60	47

```
y_train
```

```
169    4
97     0
31     3
12     2
35     3
      ..
106    0
14     2
92     0
179    4
102    0
Name: cluster, Length: 140, dtype: int32
```

```
y_test
```

```
158    1
128    1
115    0
69     0
170    1
174    1
45     3
66     0
182    1
165    4
78     0
186    1
177    4
56     0
152    1
82     0
68     0
124    1
16     2
148    1
93     0
65     0
```

```
125    4
132    0
9      3
18     2
55     0
75     0
150    1
104    0
135    4
137    4
164    1
76     0
79     0
197    4
38     2
24     2
122    0
195    4
29     3
19     3
143    4
86     0
114    0
173    4
5      3
126    0
117    0
73     0
140    1
98     0
172    1
96     0
Name: cluster, dtype: int32
```

```
from sklearn.tree import DecisionTreeRegressor
der=DecisionTreeRegressor()
der.fit(z_train,y_train)
y_pred=der.predict(z_test)
y_pred
```

```
array([0., 3., 2., 1., 1., 0., 0., 1., 1., 3., 0., 1., 4., 0., 1., 4., 0.,
       1., 0., 0., 1., 2., 1., 0., 0., 0., 0., 0., 4., 1., 3., 2., 0., 0.,
       1., 0., 4., 4., 1., 0., 0., 4., 2., 2., 0., 4., 3., 3., 4., 0., 0.,
       4., 3., 1., 0., 0., 1., 0., 1., 0.])
```

```
from sklearn.metrics import mean_absolute_error
print('MAE is',mean_absolute_error(y_test,y_pred))
```

```
MAE is 0.03333333333333333
```

```
from sklearn.metrics import mean_absolute_percentage_error
print('"ERROR percentage is',mean_absolute_percentage_error(y_test,y_pred))
```

```
from sklearn.metrics import mean_squared_error
z=mean_squared_error(y_test,y_pred)
z
```

```
0.03333333333333333
```