EXPERIMENT – 10

K-MEANS CLUSTERING

Aim:

To perform model clustering using K-means Clustering technique

Procedure:

- Upload the given dataset
- Import all the necessities
- Read the dataset as DataFrame
- Using seaborn visualize the trends
- Using sklearn train the model for predicting

Program:

```
from google.colab import files
uploaded=files.upload()
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
file=next(iter(uploaded))
df=pd.read_csv(file)
df.info
```

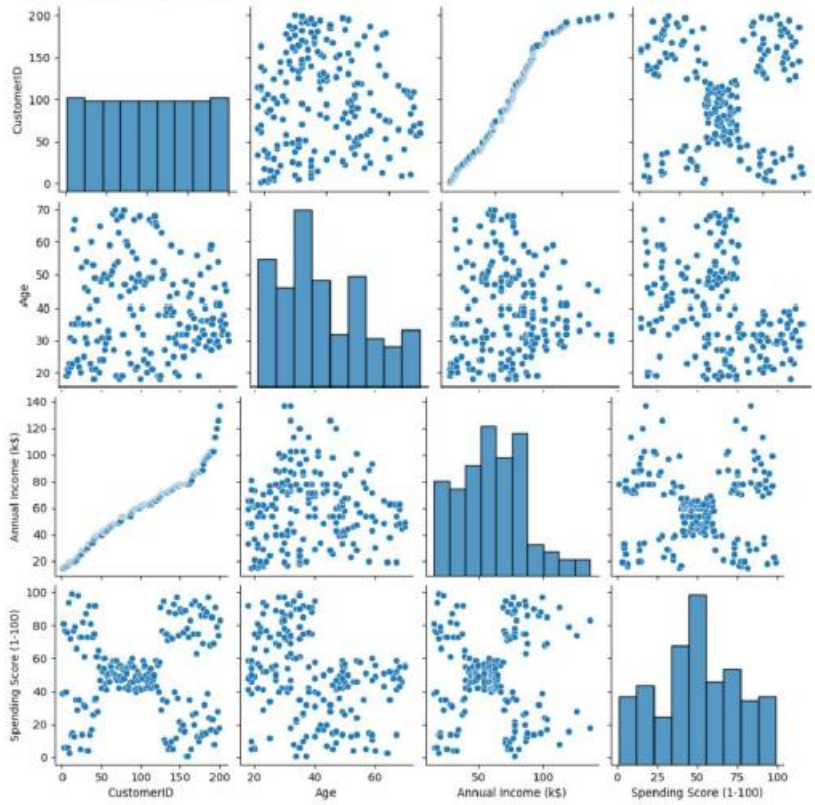
```
Choose Files  No file chosen    Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving Mall_Customers - Mall_Customers.csv to Mall_Customers - Mall_Customers.csv
pandas.core.frame.DataFrame.info
def info(verbose: bool | None=None, buf: WriteBuffer[str] | None=None, max_cols: int | None=None,
memory_usage: bool | str | None=None, show_counts: bool | None=None) -> None

Print a concise summary of a DataFrame.

This method prints information about a DataFrame including
the index dtype and columns, non-null values and memory usage.

Parameters
```

```
df.head()
```

	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

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7ef83f0bb860>
```




```
features=df.iloc[:,[3,4]].values
from sklearn.cluster import KMeans
model=KMeans(n_clusters=5)
model.fit(features)
KMeans(n_clusters=5)
```

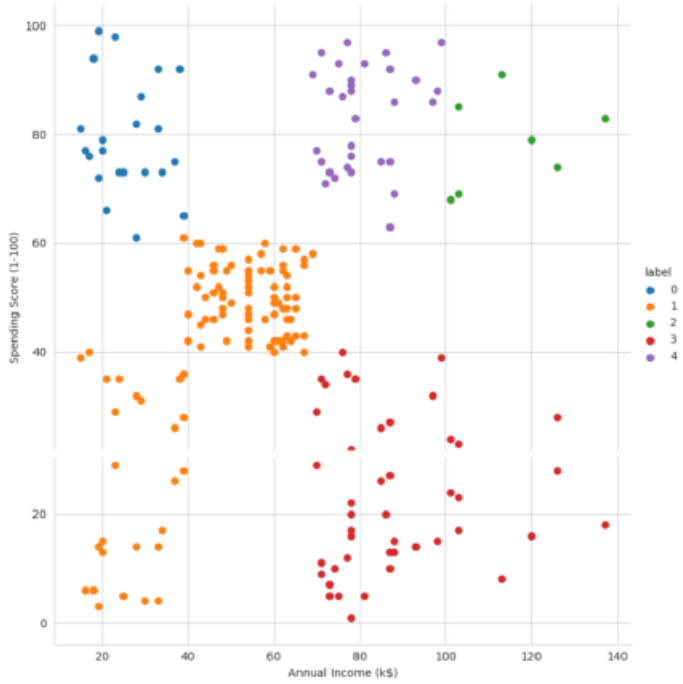
```
KMeans(n_clusters=5)
```

```
Final=df.iloc[:,[3,4]]
Final['label']=model.predict(features)
Final.head()
```

```
/tmp/ipython-input-470183701.py:2: 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_guide/indexing.html#returning-a-view-versus-a-copy
  Final['label']=model.predict(features)
```

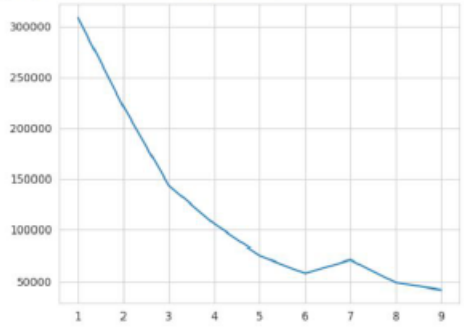
	Annual Income (k$)	Spending Score (1-100)	label
0	15	39	1
1	15	81	0
2	16	6	1
3	16	77	0
4	17	40	1

```
sns.set_style("whitegrid")
sns.FacetGrid(Final,hue="label",height=8)\
.map(plt.scatter,"Annual Income (k$)","Spending Score (1-100)")\
.add_legend();
plt.show()
```




```
features_el=df.iloc[:,[2,3,4]].values
from sklearn.cluster import KMeans
wcss=[]
for i in range(1,10):
  model=KMeans(n_clusters=i)
  model.fit(features_el)
  wcss.append(model.inertia_)
plt.plot(range(1,10),wcss)
```

```
[<matplotlib.lines.Line2D at 0x7ef8315c1bb0>]
```



Result:

Thus the python program to make prediction model using K-means Clustering is executed and verified successfully