```
pip install opendatasets
```

```
Collecting opendatasets
  Downloading opendatasets-0.1.22-py3-none-any.whl (15 kB)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-
packages (from opendatasets) (4.66.4)
Requirement already satisfied: kaggle in
/usr/local/lib/python3.10/dist-packages (from opendatasets) (1.6.12)
Requirement already satisfied: click in
/usr/local/lib/python3.10/dist-packages (from opendatasets) (8.1.7)
Requirement already satisfied: six>=1.10 in
/usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets)
(1.16.0)
Requirement already satisfied: certifi>=2023.7.22 in
/usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets)
(2024.2.2)
Requirement already satisfied: python-dateutil in
/usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets)
(2.8.2)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets)
(2.31.0)
Requirement already satisfied: python-slugify in
/usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets)
(8.0.4)
Requirement already satisfied: urllib3 in
/usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets)
(2.0.7)
Requirement already satisfied: bleach in
/usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets)
(6.1.0)
Requirement already satisfied: webencodings in
/usr/local/lib/python3.10/dist-packages (from bleach->kaggle-
>opendatasets) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in
/usr/local/lib/python3.10/dist-packages (from python-slugify->kaggle-
>opendatasets) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->kaggle-
>opendatasets) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests->kaggle-
>opendatasets) (3.7)
Installing collected packages: opendatasets
Successfully installed opendatasets-0.1.22
```

```
import opendatasets as od
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import warnings
warnings.simplefilter('ignore')
```

```
od.download("https://www.kaggle.com/datasets/zubairmustafa/shopping-mall-customer-segmentation-data")
```

```
Skipping, found downloaded files in "./shopping-mall-customer-segmentation-data" (use force=True to force download)
```

```
import pandas as pd

df = pd.read_csv("/content/shopping-mall-customer-segmentation-data/Shopping Mall Customer Segmentation Data .csv")

data
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 15079,\n  \"fields\": [\n    {\n      \"column\": \"Customer ID\",\n      \"properties\": {\n        \"dtype\": \"string\",\n        \"num_unique_values\": 15079,\n        \"samples\": [\n          \"73ebeee3-01c6-4ffc-8a7f-bf81391443a4\",\n          \"81a16720-019a-4bc7-91c4-1e04e69d81d5\",\n          \"e1034276-ecee-4b46-85e8-8d423f7f035e\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Age\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 21,\n        \"min\": 18,\n        \"max\": 90,\n        \"num_unique_values\": 73,\n        \"samples\": [\n          24,\n          34,\n          21\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Gender\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 2,\n        \"samples\": [\n          \"Female\",\n          \"Male\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Annual Income\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 52249,\n        \"min\": 20022,\n        \"max\": 199974,\n        \"num_unique_values\": 14441,\n        \"samples\": [\n          75279,\n          71541\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Spending Score\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 28,\n        \"min\": 1,\n        \"max\": 100,\n        \"num_unique_values\": 100,\n        \"samples\": [\n          100,\n          72\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe","variable_name":"data"}
```


```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15079 entries, 0 to 15078
```

```
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Customer ID     15079 non-null  object
 1   Age             15079 non-null  int64
 2   Gender          15079 non-null  object
 3   Annual Income   15079 non-null  int64
 4   Spending Score  15079 non-null  int64
dtypes: int64(3), object(2)
memory usage: 589.1+ KB
```

```
data.isnull().sum()
```

```
Customer ID       0
Age               0
Gender            0
Annual Income     0
Spending Score    0
dtype: int64
```

```
data.describe()
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 8,\n  \"fields\": [\
n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 5313.846781202327,\n
\"min\": 18.0,\n        \"max\": 15079.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
54.191590954307316,\n          54.0,\n          15079.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Annual Income\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
65113.71932206709,\n        \"min\": 15079.0,\n        \"max\":
199974.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
109742.88056237152,\n          109190.0,\n          15079.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Spending Score\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 5314.534466194705,\n        \"min\":
1.0,\n        \"max\": 15079.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          50.59161748126534,\n          51.0,\n
15079.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```


```
print(data.columns)
```

```
Index(['Customer ID', 'Age', 'Gender', 'Annual Income', 'Spending
Score'], dtype='object')
```

```
num_col=[]
for col in df.columns:
    if(df[col].dtypes!='object'):
```

```
        num_col.append(col)
print(f"There are total {len(num_col)} numerical columns in dataset")
print(num_col)
```

```
There are total 3 numerical columns in dataset
['Age', 'Annual Income', 'Spending Score']
```

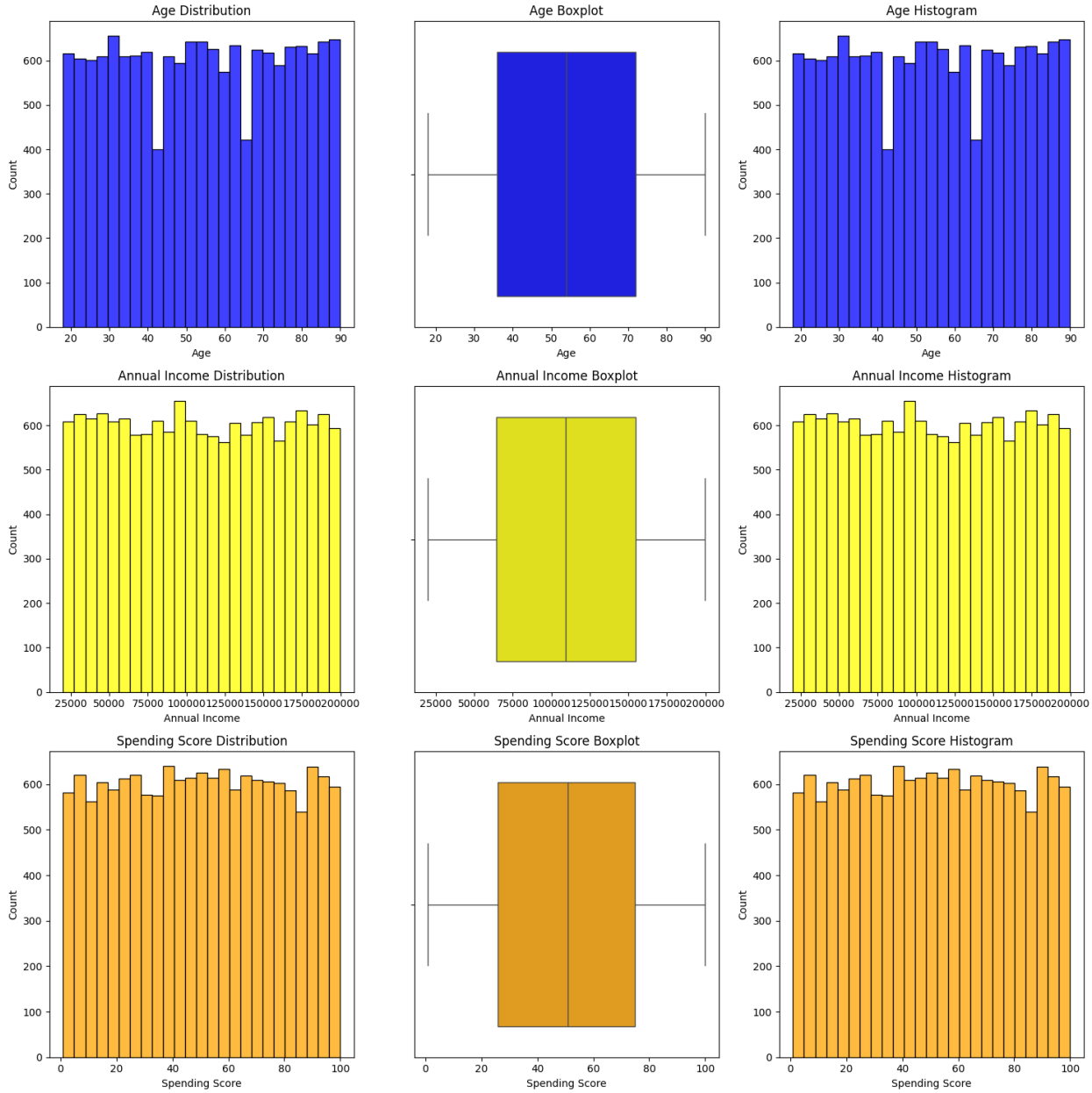
```
df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 15079,\n  \"fields\":
[\n    {\n      \"column\": \"Customer ID\",\n      \"properties\": {\
n        \"dtype\": \"string\",\n        \"num_unique_values\":
15079,\n        \"samples\": [\n          \"73ebeee3-01c6-4ffc-8a7f-
bf81391443a4\",\n          \"81a16720-019a-4bc7-91c4-1e04e69d81d5\",\n
\"e1034276-ecee-4b46-85e8-8d423f7f035e\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 21,\n        \"min\": 18,\n
\"max\": 90,\n        \"num_unique_values\": 73,\n        \"samples\":
[\n          24,\n          34,\n          21\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Gender\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"Female\",\n          \"Male\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Annual Income\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 52249,\n        \"min\": 20022,\n
\"max\": 199974,\n        \"num_unique_values\": 14441,\n
\"samples\": [\n          75279,\n          71541\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Spending Score\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
28,\n        \"min\": 1,\n        \"max\": 100,\n
\"num_unique_values\": 100,\n        \"samples\": [\n          100,\n
72\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


```
cat_col=[]
for col in df.columns:
    if(df[col].dtypes=='object'):
        cat_col.append(col)
print(f"There are total {len(cat_col)} categorical columns in
dataset")
print(cat_col)
```

```
There are total 2 categorical columns in dataset
['Customer ID', 'Gender']
```

```
fig, axes = plt.subplots(3, 3, figsize=(15, 15))

sns.histplot(df['Age'], ax=axes[0, 0], color='blue')
axes[0, 0].set_title('Age Distribution')
axes[0, 0].set_xlabel('Age')

sns.boxplot(df['Age'], ax=axes[0, 1], orient='h', color='blue')
axes[0, 1].set_title('Age Boxplot')
axes[0, 1].set_xlabel('Age')

sns.histplot(df['Age'], ax=axes[0, 2], color='blue')
axes[0, 2].set_title('Age Histogram')
axes[0, 2].set_xlabel('Age')

sns.histplot(df['Annual Income'], ax=axes[1, 0], color='yellow')
axes[1, 0].set_title('Annual Income Distribution')
axes[1, 0].set_xlabel('Annual Income')

sns.boxplot(df['Annual Income'], ax=axes[1, 1], orient='h',
color='yellow')
axes[1, 1].set_title('Annual Income Boxplot')
axes[1, 1].set_xlabel('Annual Income')

sns.histplot(df['Annual Income'], ax=axes[1, 2], color='yellow')
axes[1, 2].set_title('Annual Income Histogram')
axes[1, 2].set_xlabel('Annual Income')

sns.histplot(df['Spending Score'], ax=axes[2, 0], color='orange')
axes[2, 0].set_title('Spending Score Distribution')
axes[2, 0].set_xlabel('Spending Score')

sns.boxplot(df['Spending Score'], ax=axes[2, 1], orient='h',
color='orange')
axes[2, 1].set_title('Spending Score Boxplot')
axes[2, 1].set_xlabel('Spending Score')

sns.histplot(df['Spending Score'], ax=axes[2, 2], color='orange')
axes[2, 2].set_title('Spending Score Histogram')
axes[2, 2].set_xlabel('Spending Score')

plt.tight_layout()
plt.show()
```

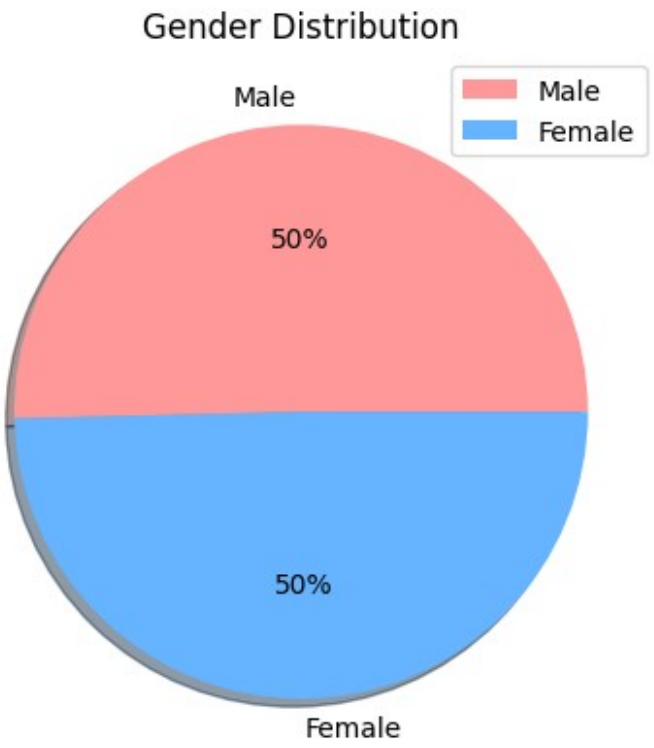



```
df.describe()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n{\n      \"column\": \"Age\",\n      \"properties\": {\n\"dtype\": \"number\",\n        \"std\": 5313.846781202327,\n\"min\": 18.0,\n        \"max\": 15079.0,\n\"num_unique_values\": 8,\n        \"samples\": [\n54.191590954307316,\n          54.0,\n          15079.0\n        ],\n\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Annual Income\",\n\"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 65113.71932206709,\n        \"min\": 15079.0,\n        \"max\":
```

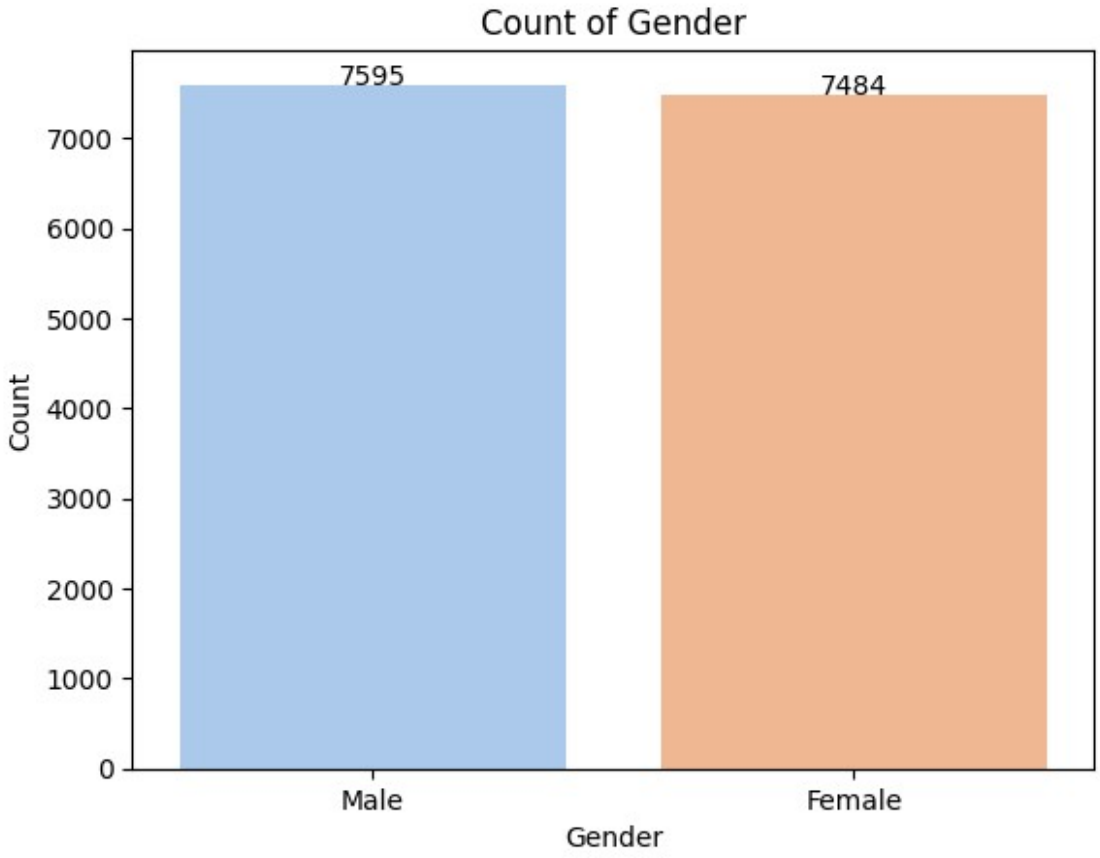
```
199974.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
109742.88056237152,\n          109190.0,\n          15079.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Spending Score\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 5314.534466194705,\n        \"min\":
1.0,\n        \"max\": 15079.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          50.59161748126534,\n          51.0,\n
15079.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```
colors = ['#ff9999', '#66b3ff']

explode = (0.0, 0)

df['Gender'].value_counts().plot(kind='pie', autopct="%0.0f%%",
colors=colors, explode=explode, shadow=True)

plt.title('Gender Distribution')
plt.ylabel('')
plt.legend(df['Gender'].unique(), loc='upper right')
plt.show()
```




```
gender_counts = df['Gender'].value_counts()
```

```
index = gender_counts.index
value = gender_counts.values

sns.barplot(x=index, y=value, palette='pastel')

plt.title('Count of Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
for i, v in enumerate(value):
    plt.text(i, v + 0.2, str(v), ha='center')

plt.show()
```




```
group = df.groupby('Gender')

group.get_group('Female')[[ 'Age','Annual Income', 'Spending
Score']].agg(['mean','max','min']).reset_index()
```


```
{"summary":"{\n  \"name\": \"group\",\n  \"rows\": 3,\n  \"fields\":
[\n    {\n      \"column\": \"index\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"mean\",\n          \"max\",\n
\"min\"\n        ],\n        \"semantic_type\": \"\",\n
```

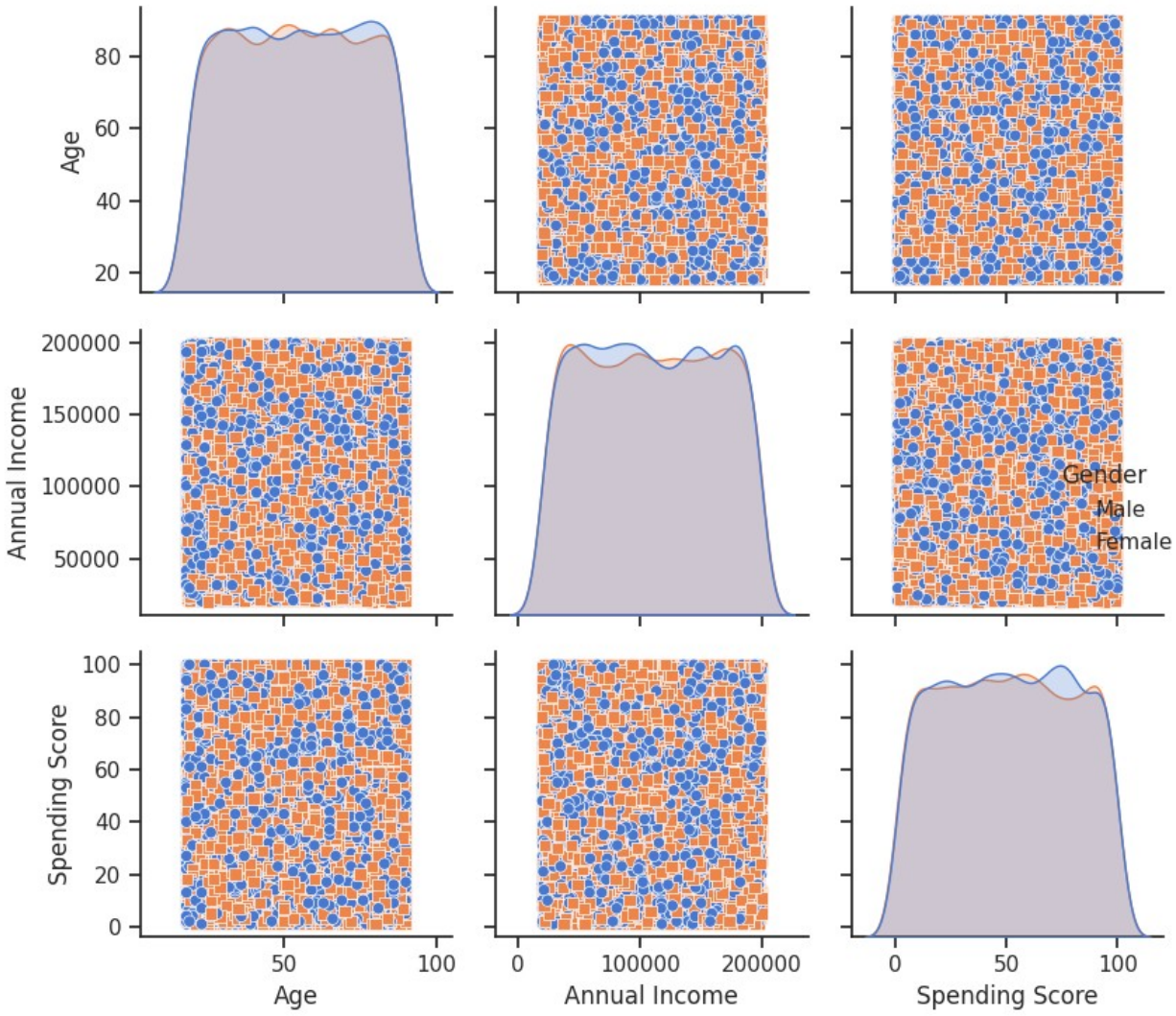
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 36.000101841323456,\n   \"min\": 18.0,\n   \"max\":
90.0,\n   \"num_unique_values\": 3,\n   \"samples\": [\n
54.148316408337784,\n   90.0,\n   18.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Annual Income\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
89970.51490519308,\n   \"min\": 20022.0,\n   \"max\":
199963.0,\n   \"num_unique_values\": 3,\n   \"samples\": [\n
109902.79957242117,\n   199963.0,\n   20022.0\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Spending Score\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 49.50003527226734,\n   \"min\":
1.0,\n   \"max\": 100.0,\n   \"num_unique_values\": 3,\n
\"samples\": [\n   50.397648316408336,\n   100.0,\n
1.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
group.get_group('Male')[['Age','Annual Income', 'Spending
Score']].agg(['mean','max','min']).reset_index()
```

{"summary":"{\n \"name\": \"group\",\n \"rows\": 3,\n \"fields\":
[\n   {\n   \"column\": \"index\",\n   \"properties\": {\n
\"dtype\": \"string\",\n   \"num_unique_values\": 3,\n
\"samples\": [\n   \"mean\",\n   \"max\",\n
\"min\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 36.00025400429281,\n   \"min\": 18.0,\n   \"max\":
90.0,\n   \"num_unique_values\": 3,\n   \"samples\": [\n
54.23423304805793,\n   90.0,\n   18.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Annual Income\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
89974.31856762875,\n   \"min\": 20026.0,\n   \"max\":
199974.0,\n   \"num_unique_values\": 3,\n   \"samples\": [\n
109585.29874917709,\n   199974.0,\n   20026.0\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Spending Score\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 49.50026918642728,\n   \"min\":
1.0,\n   \"max\": 100.0,\n   \"num_unique_values\": 3,\n
\"samples\": [\n   50.78275181040158,\n   100.0,\n
1.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
sns.set(style="ticks", palette="muted")
sns.pairplot(df, hue='Gender', markers=["o", "s"])
```

```
plt.tight_layout()
plt.show()
```




```
df[num_col].sample(10)
```


```
{"summary":"{\n  \"name\": \"df[num_col]\",\n  \"rows\": 10,\n
\"fields\": [\n    {\n      \"column\": \"Age\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
18,\n        \"min\": 27,\n        \"max\": 83,\n
\"num_unique_values\": 10,\n        \"samples\": [\n          57,\n
80,\n          61\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Annual Income\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 50589,\n        \"min\": 34808,\n
\"max\": 192792,\n        \"num_unique_values\": 10,\n
\"samples\": [\n          114715,\n          192792,\n          83995\
```

```
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Spending Score\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 29,\n        \"min\": 19,\n
\"max\": 94,\n        \"num_unique_values\": 9,\n        \"samples\":
[\n          45,\n          94,\n          28\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```


```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
df[num_col]=scaler.fit_transform(df[num_col])
```

```
df[num_col].sample(5)
```


```
{"summary":"{\n  \"name\": \"df[num_col]\",\n  \"rows\": 5,\n
\"fields\": [\n    {\n      \"column\": \"Age\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.8634408853093851,\n        \"min\": -0.43523866907561026,\n
\"max\": 1.458834603937702,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          1.222075444811038,\n
1.458834603937702,\n          0.03827964917771778\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Annual Income\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.9525957921701192,\n        \"min\": -1.0361200108164168,\n
\"max\": 1.3382814628093946,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          -0.6184173599348889,\n
1.3382814628093946,\n          -1.0361200108164168\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Spending Score\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.8265107083685672,\n        \"min\": -0.751639586442482,\n
\"max\": 1.1978121800729868,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          0.6756375997563434,\n          -
0.4731464769402721,\n          -0.751639586442482\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```


```
cat_col
```

```
['Customer ID', 'Gender']
```

```
df[cat_col].sample(5)
```


```
{"summary":"{\n  \"name\": \"df[cat_col]\",\n  \"rows\": 5,\n
\"fields\": [\n    {\n      \"column\": \"Customer ID\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"0eb4fece-e9b7-4371-8c31-2157486b13a7\",\n          \"6e2a80b2-7879-
4462-9a59-77cc2dfe3c61\",\n          \"7f64ea86-b7c0-4dca-a5b9-
```

```
77db48512957\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Gender\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Female\",\n          \"Male\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```


```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df['Gender']=encoder.fit_transform(df['Gender'])
```

```
df[cat_col].sample(5)
```


```
{"summary":"{\n  \"name\": \"df[cat_col]\",\n  \"rows\": 5,\n
\"fields\": [\n    {\n      \"column\": \"Customer ID\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"7a911220-fc1f-4873-9926-93441a2db0aa\",\n          \"21ab5613-f56e-
4a8d-aa58-86e5132a365e\",\n          \"ff04685c-4bdd-4f0f-ada2-
e9314ebfb0ae\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Gender\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

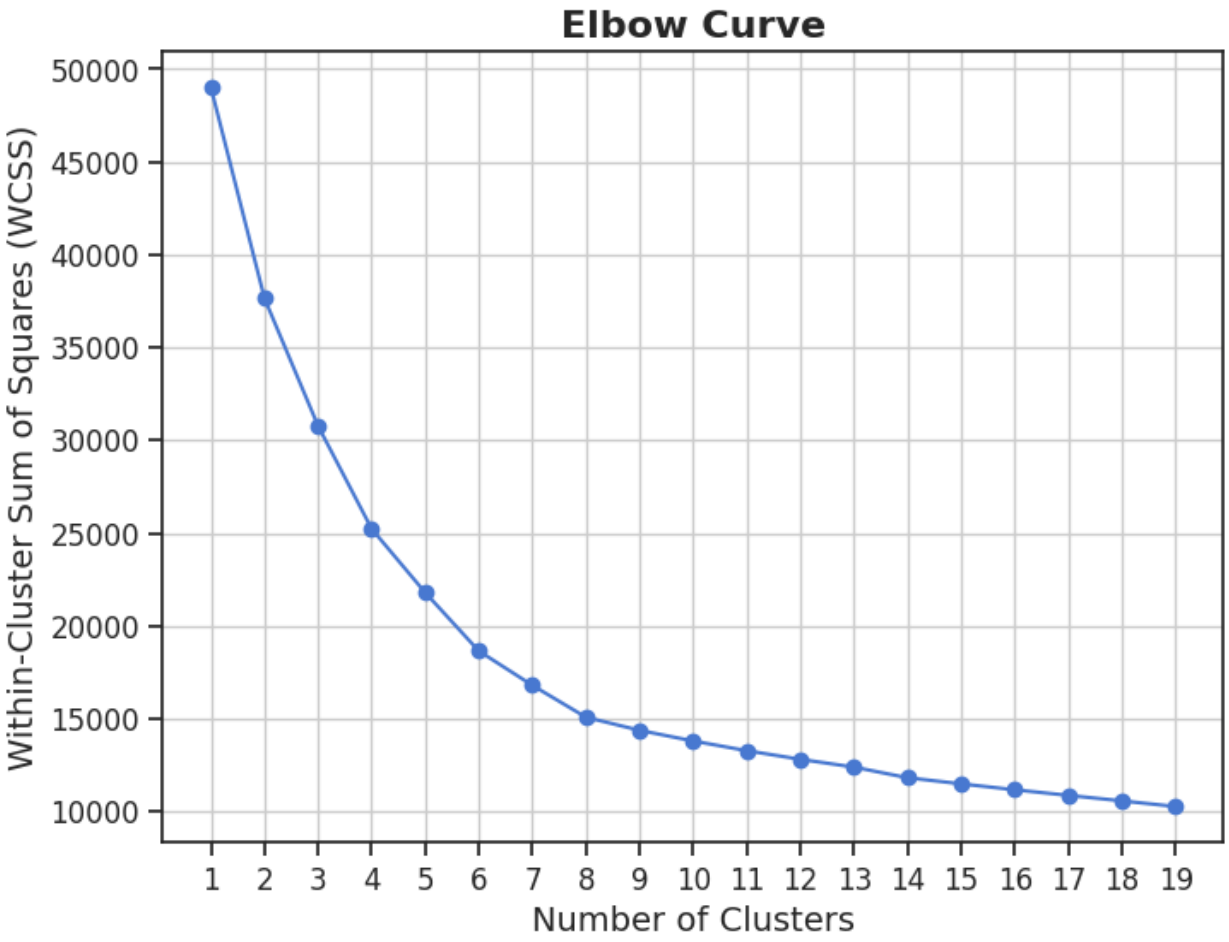

```
Feature=df.drop(columns={'Customer ID'})
```

```
Feature.sample(10)
```

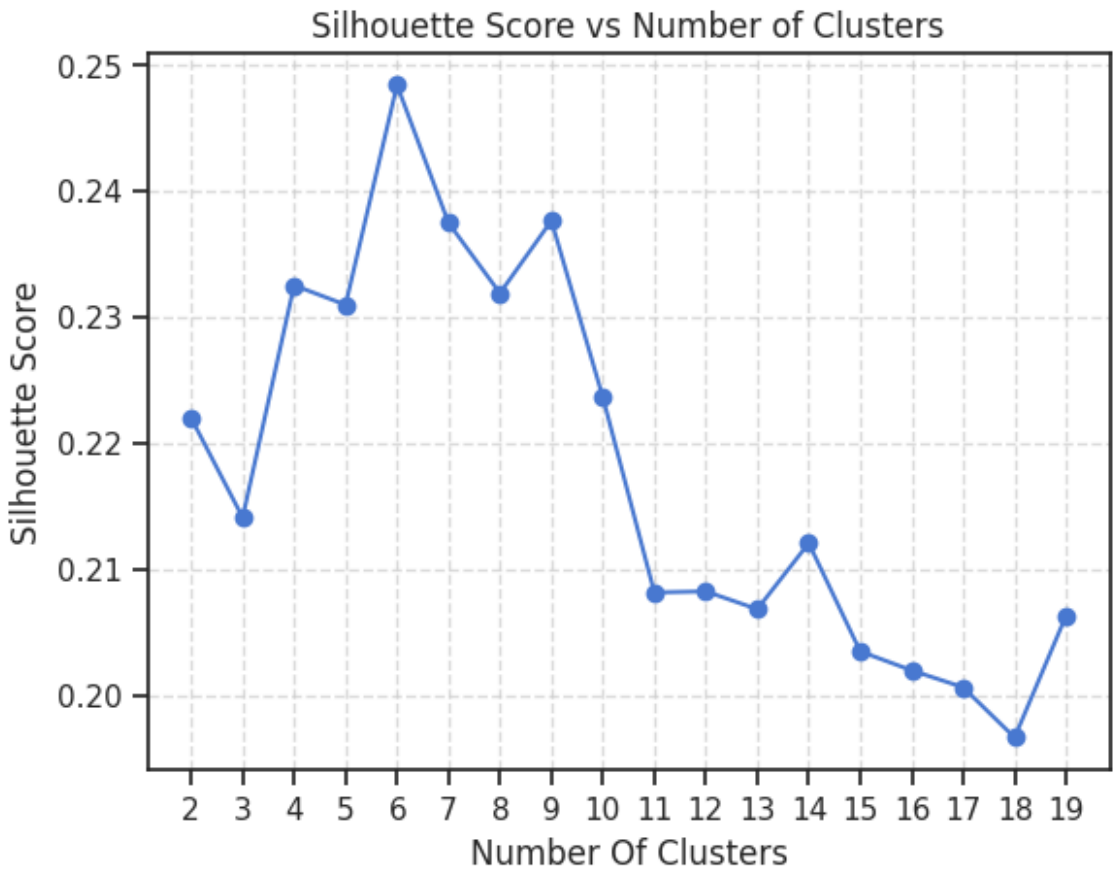

```
{"summary":"{\n  \"name\": \"Feature\",\n  \"rows\": 10,\n
\"fields\": [\n    {\n      \"column\": \"Age\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.0940561925710495,\n        \"min\": -1.713738128359596,\n
\"max\": 1.3167791084617035,\n        \"num_unique_values\": 10,\n
\"samples\": [\n          -1.0508124828049368,\n
1.0326681175097068,\n          -1.003460650979604\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Gender\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 0,\n        \"max\": 1,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          1,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Annual Income\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.9060867165174005,\n        \"min\": -1.45033925432714,\n
\"max\": 1.6599798753834745,\n        \"num_unique_values\": 10,\n
\"samples\": [\n          -0.8849363030033125,\n
0.5229344628298725\n        ],\n        \"semantic_type\": \"\",\n
```

\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Spending Score\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 1.037681430960551,\n   \"min\": -
1.69155383101244,\n   \"max\": 1.5459285669507492,\n
\"num_unique_values\": 10,\n   \"samples\": [\n   -
1.4478723601980066,\n   1.3022470961363155\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe"}

```
from sklearn.cluster import KMeans

wcss=[]
for i in range(1,20):
    KM=KMeans(n_clusters=i,n_init=10)
    KM.fit(Feature)
    wcss.append(KM.inertia_)

plt.figure(figsize=(8, 6))
plt.plot(range(1, 20), wcss, marker='o', linestyle='-', color='b')
plt.xlabel("Number of Clusters", fontsize=14)
plt.ylabel("Within-Cluster Sum of Squares (WCSS)", fontsize=14)
plt.title("Elbow Curve", fontsize=16, fontweight='bold')
plt.grid(True)
plt.xticks(range(1, 20), fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```



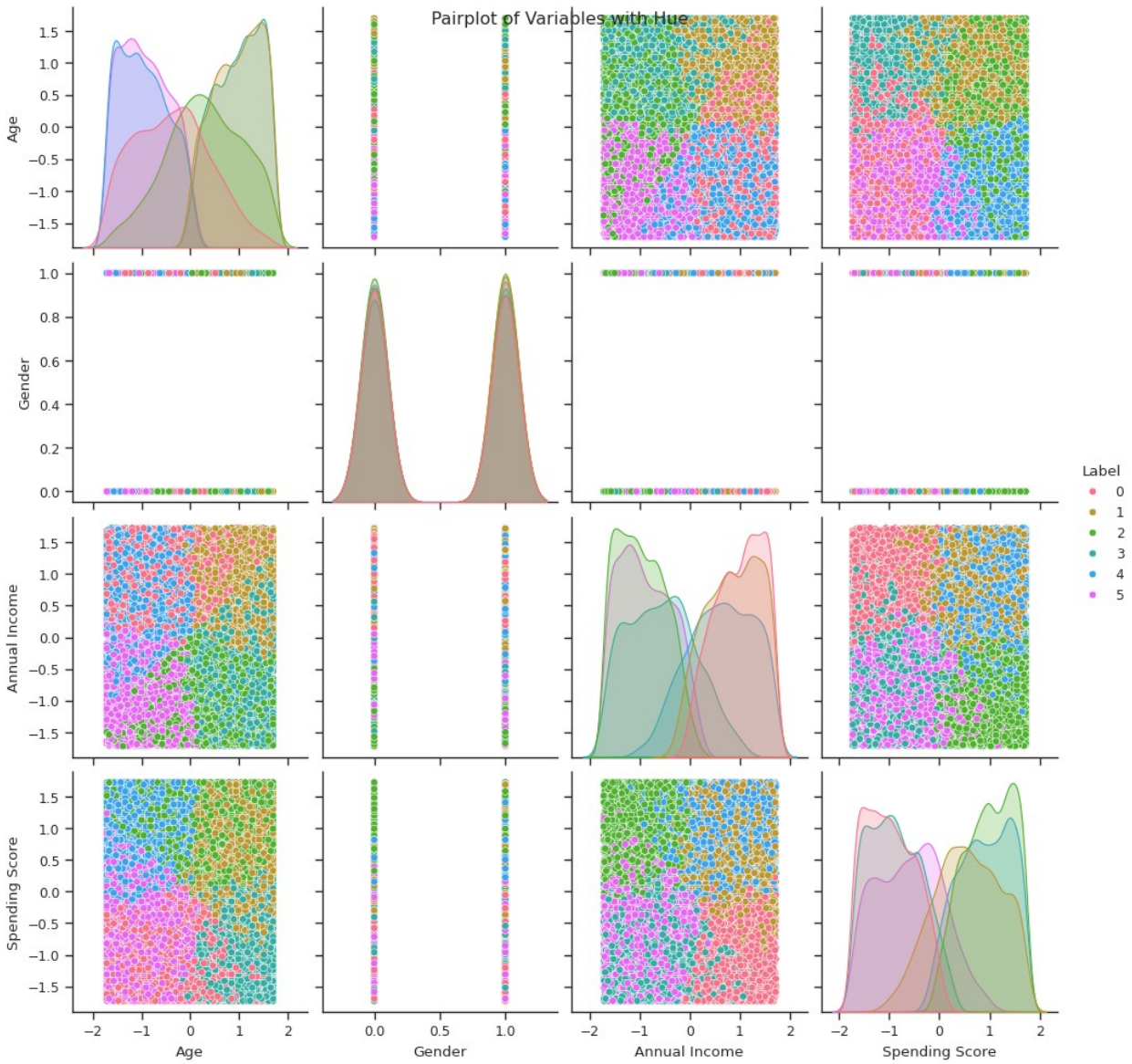

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

sil_scores = []

for i in range(2, 20):  # Starting from 2 clusters as silhouette score
requires at least 2 clusters
    km = KMeans(n_clusters=i, n_init=10)
    labels = km.fit_predict(Feature)
    sil_score = silhouette_score(Feature, labels)
    sil_scores.append(sil_score)

plt.plot(range(2, 20), sil_scores, marker='o', linestyle='-')
plt.xlabel("Number Of Clusters")
plt.ylabel("Silhouette Score")
plt.title("Silhouette Score vs Number of Clusters")
plt.grid(True, linestyle='--', alpha=0.7)
plt.xticks(range(2, 20))
plt.show()
```




```
model=KMeans(n_clusters=6,n_init=10)

df['Label']=model.fit_predict(Feature)

from sklearn.metrics import silhouette_score
silhouette_score(Feature,df['Label'])
```

```
0.2483586990932
```

```
sns.set(style="ticks")
sns.set_context("paper")

palette = sns.color_palette("husl")

sns.pairplot(df, hue='Label', palette=palette, diag_kind='kde',
markers='o', diag_kws=dict(shade=True))

plt.suptitle("Pairplot of Variables with Hue")
plt.show()
```




```
m=KMeans(n_clusters=6)
#Hyperpera meter tuning



param_grid=[{'init':['k-means++','random']},{'algorithm':
['lloyd','elkan']}]


from sklearn.model_selection import GridSearchCV
gsv=GridSearchCV(m,param_grid,cv=10,scoring=silhouette_score)

gsv.fit(Feature)
```

```
GridSearchCV(cv=10, estimator=KMeans(n_clusters=6),
             param_grid=[{'init': ['k-means++', 'random']},
                         {'algorithm': ['lloyd', 'elkan']}],
             scoring=<function silhouette_score at 0x7a9943dd3640>)
```

```
gsv.best_params_
```

```
{'init': 'k-means++'}
```

```
if df['Customer ID'].isin(df['Customer ID']).any():
    df['Label'] = df.loc[df['Customer ID'].isin(df['Customer ID']),
'Label']
```

```
df.sample(10)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 10,\n  \"fields\": [\n
{\n      \"column\": \"Customer ID\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 10,\n
\"samples\": [\n          \"f8a91903-05a8-49fa-820f-d04aadfbd09d\",\n
\"dd2af46f-e80e-4b0c-b5fe-a85b9e9ab3f7\",\n          \"a2aaab12-85b7-
4e8a-ac74-1b29185c9bde\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Age\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.8691780971639277,\n        \"min\": -
1.713738128359596,\n        \"max\": 0.8906126220337083,\n
\"num_unique_values\": 9,\n        \"samples\": [\n
0.8906126220337083,\n          0.1803351446537162,\n          -
0.43523866907561026\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Gender\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Annual Income\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.803482786802926,\n        \"min\": -
1.0543791901117328,\n        \"max\": 1.3453248359547576,\n
\"num_unique_values\": 10,\n        \"samples\": [\n          -
1.0543791901117328,\n          1.228994341450637\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Spending Score\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.9896687587518641,\n        \"min\": -1.2390025280713493,\n
\"max\": 1.5807402056385254,\n        \"num_unique_values\": 10,\n
\"samples\": [\n          -0.8212628638180344,\n
0.1882746581274762\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Label\",\n      \"properties\": {\n        \"dtype\": \"int32\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n          1,\n
```

```
3\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```