Assignment-5

Name: K Naga Sai Krishna

Assignment:

Market Basket Magic: Extracting Insights for Retail Success

Customer segmentation is a crucial aspect of retail and marketing strategy. Mall Customer Segmentation is a common data analysis project that involves categorizing mall customers into distinct groups or segments based on various characteristics and behaviors. This segmentation is valuable for tailoring marketing efforts, optimizing store layouts, and enhancing customer experiences.

Dataset link: Here

Task:

- Understand the data
- Data Preprocessing
- Machine Learning approach with clustering algorithm

Data Preprocessing and data understanding

```
[3] import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    from sklearn.cluster import KMeans
```

```
[4] from google.colab import files
    uploaded = files.upload()
```

Browse... Mall_Customers.csv

Mall_Customers.csv(application/vnd.ms-excel) - 3981 bytes, last modified: n/a - 100% done

```
Saving Mall_Customers.csv to Mall_Customers.csv
```

```
[14] import io
     df = pd.read_csv(io.BytesIO(uploaded['Mall_Customers.csv']))
```

```
[15] 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

```
[16] df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
CustomerID	200.0	100.50	57.879185	1.0	50.75	100.5	150.25	200.0
Age	200.0	38.85	13.969007	18.0	28.75	36.0	49.00	70.0
Annual Income (k$)	200.0	60.56	26.264721	15.0	41.50	61.5	78.00	137.0
Spending Score (1-100)	200.0	50.20	25.823522	1.0	34.75	50.0	73.00	99.0

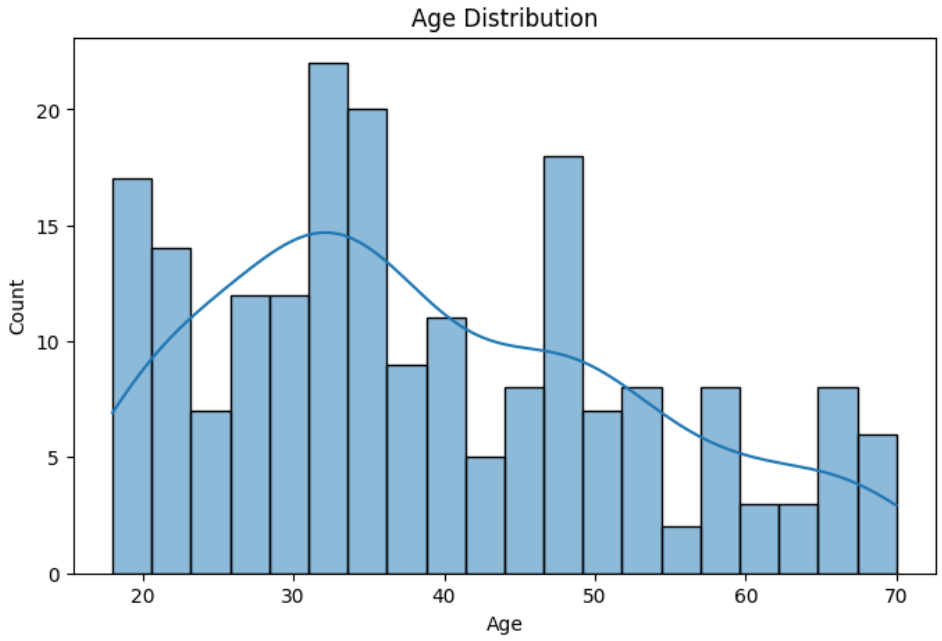
```
[17] df.isnull().sum()
```

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

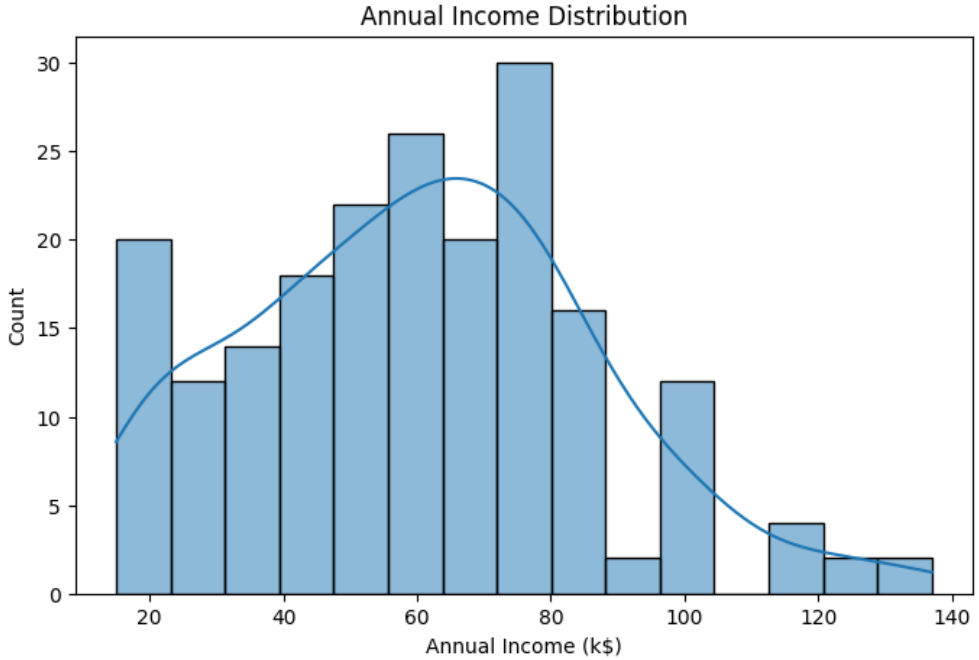
```
[18] df.drop(df.columns[[0]], axis=1, inplace=True)
```

Data Visualization:

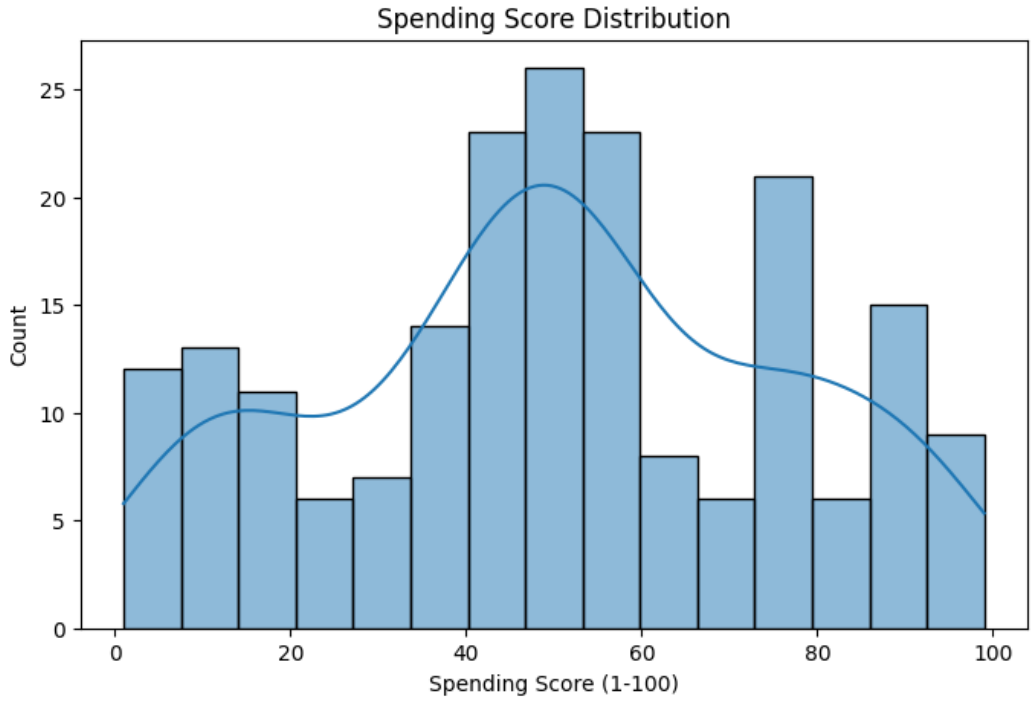
```
[19] plt.figure(figsize=(8, 5))
     sns.histplot(data=df, x='Age', bins=20, kde=True)
     plt.title('Age Distribution')
     plt.xlabel('Age')
     plt.ylabel('Count')
     plt.show()
```



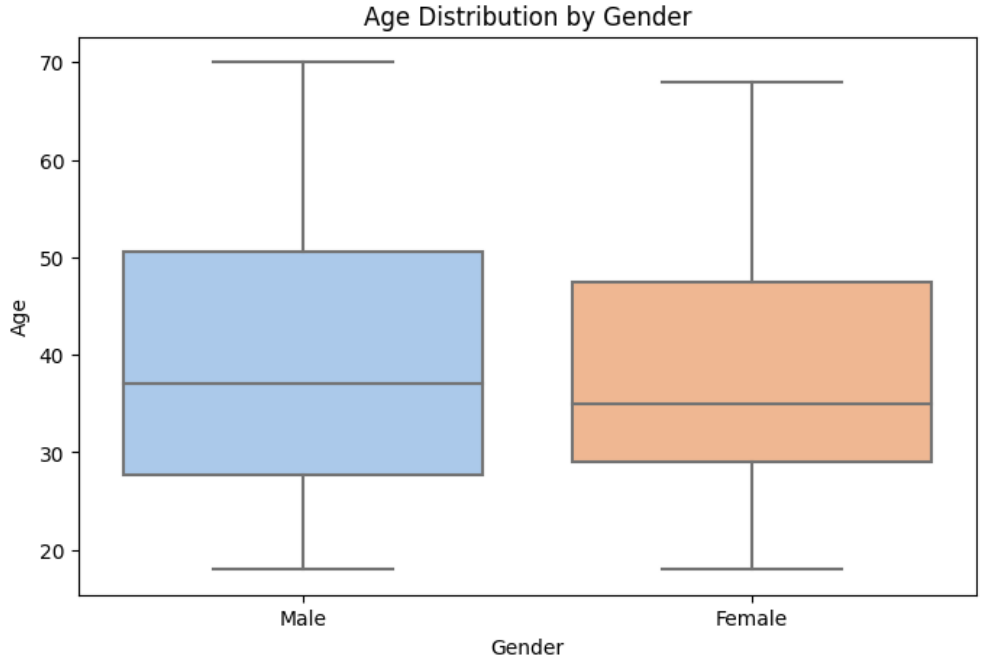

```
plt.figure(figsize=(8, 5))
sns.histplot(data=df, x='Annual Income (k$)', bins=15, kde=True)
plt.title('Annual Income Distribution')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Count')
plt.show()
```



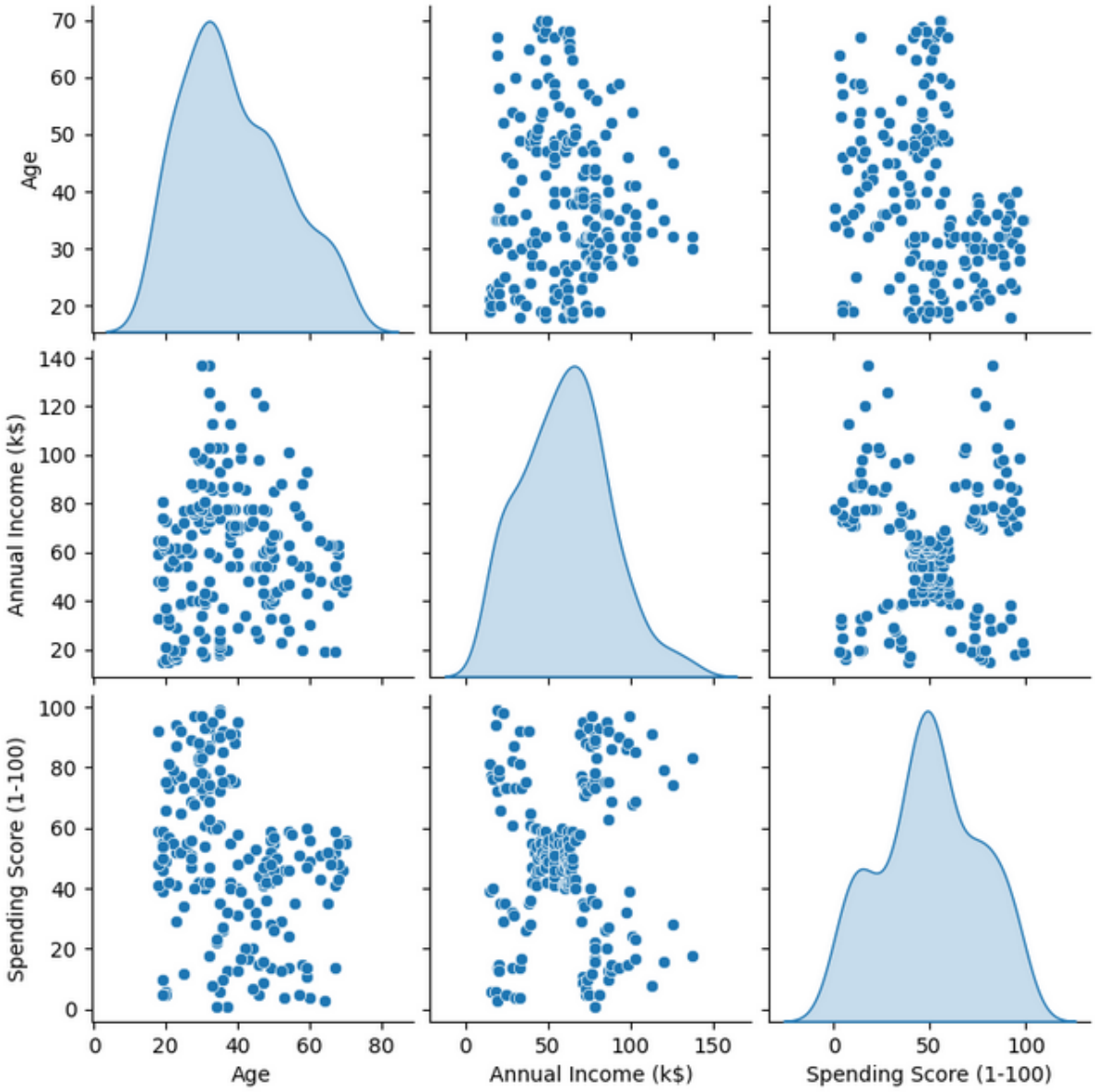
```
plt.figure(figsize=(8, 5))
sns.histplot(data=df, x='Spending Score (1-100)', bins=15, kde=True)
plt.title('Spending Score Distribution')
plt.xlabel('Spending Score (1-100)')
plt.ylabel('Count')
plt.show()
```



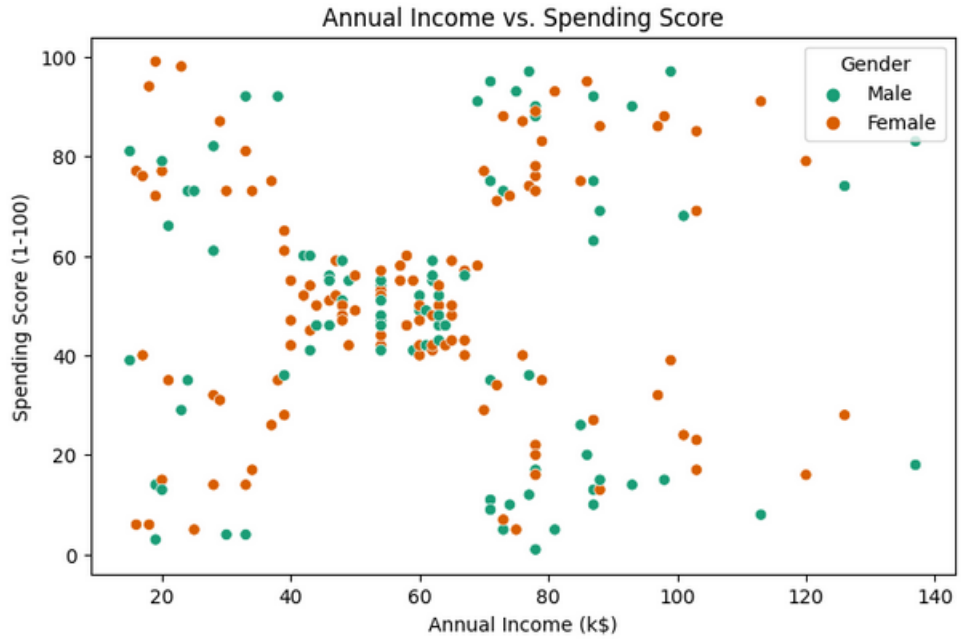

```
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='Gender', y='Age', palette='pastel')
plt.title('Age Distribution by Gender')
plt.xlabel('Gender')
plt.ylabel('Age')
plt.show()
```



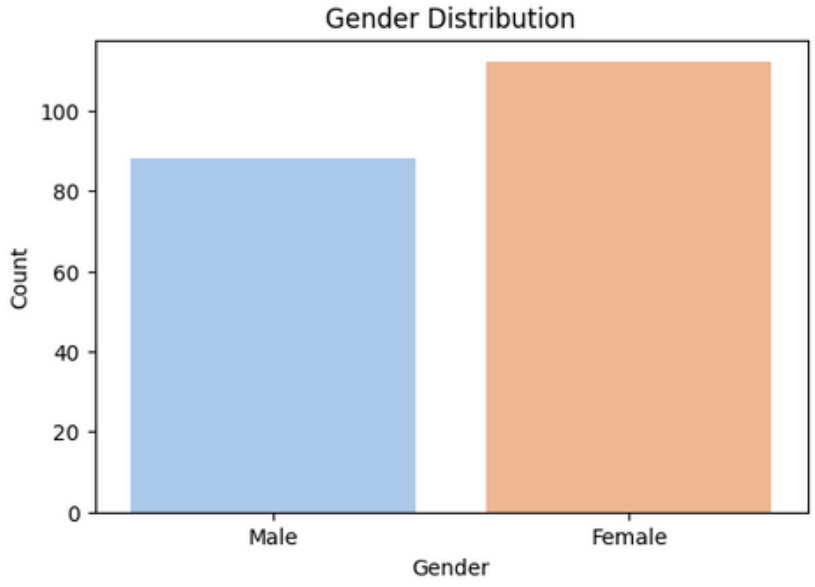
```
sns.pairplot(df[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']], diag_kind='kde')
plt.show()
```



```
plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x='Annual Income (k$)',
                y='Spending Score (1-100)', hue='Gender', palette='Dark2')
plt.title('Annual Income vs. Spending Score')
plt.show()
```




```
plt.figure(figsize=(6, 4))
sns.countplot(data=df, x='Gender', palette='pastel')
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```



Label Encoding

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

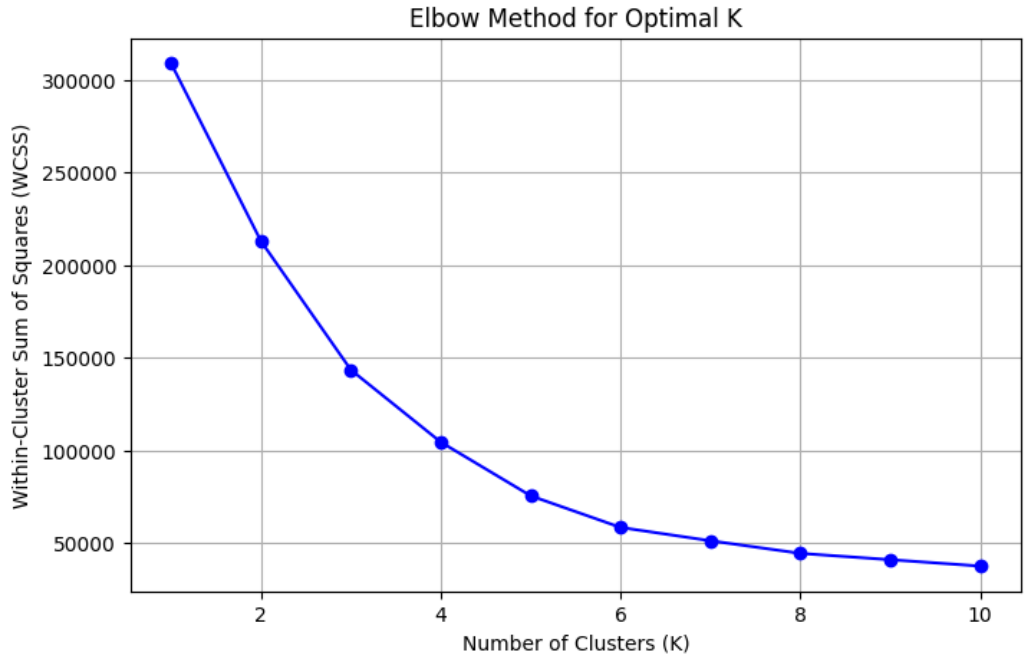
Applying K Means

```
[27] a = []
     k_values = range(1, 11)
     for k in k_values:
         kmeans = KMeans(n_clusters=k, init='k-means++', random_state=42,n_init=10)
         kmeans.fit(df[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']])
         a.append(kmeans.inertia_)
```

```
[28] a
```

```
[308812.78,
 212840.1698209719,
 143342.751571706,
 104366.15145556197,
 75378.76464074483,
 58302.40630860368,
 51118.949931647294,
 44312.46881207722,
 40894.98978213978,
 37468.51571576572]
```

```
plt.figure(figsize=(8, 5))
plt.plot(k_values, a, marker='o', linestyle='-', color='b')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Within-Cluster Sum of Squares (WCSS)')
plt.grid(True)
plt.show()
```



```
[30] from sklearn import cluster
     km_model = cluster.KMeans(n_clusters=3,init = 'k-means++',random_state=0)
```

```
[31] km_model.fit(df)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: Fut
  warnings.warn(
▾ KMeans
KMeans(n_clusters=3, random_state=0)
```

```
pred = km_model.predict(df)
pred
```

```
array([2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1], dtype=int32)
```

Predicting with random values

```
[36] print(km_model.predict([[1,20,19,40]]))

     [2]
     /usr/local/lib/python3.10/dist-packages/skle
       warnings.warn(
```

```
[47] print(km_model.predict([[1,20,115,40]]))

     [0]
     /usr/local/lib/python3.10/dist-packages/skle
       warnings.warn(
```

```
[48] print(km_model.predict([[0,43,190,89]]))

     [1]
     /usr/local/lib/python3.10/dist-packages/skle
       warnings.warn(
```

```
[51] print(km_model.predict([[0,27,69,96]]))

     [1]
     /usr/local/lib/python3.10/dist-packages/skle
       warnings.warn(
```