```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

customer_data = pd.read_csv('mall_customer.csv')
print(customer_data.info());
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   CustomerID           500 non-null    object 
 1   Name                 500 non-null    object 
 2   Age                  500 non-null    int64  
 3   Gender               500 non-null    object 
 4   MembershipLevel      500 non-null    object 
 5   IncomeLevel          500 non-null    float64
 6   ElectronicsSpending  500 non-null    float64
 7   ClothingSpending     500 non-null    float64
 8   GrocerySpending      500 non-null    float64
 9   HomeSpending         500 non-null    float64
 10  Visits               500 non-null    int64  
 11  PurchaseFrequency    500 non-null    int64  
 12  OnlineActivity       500 non-null    float64
 13  EmailOpens           500 non-null    float64
 14  AppUsage             500 non-null    float64
 15  LoyaltyPoints        500 non-null    float64
dtypes: float64(9), int64(3), object(4)
memory usage: 62.6+ KB
None
```

```python
numeric_feature_names = [
    'Age', 'IncomeLevel', 'ElectronicsSpending', 'ClothingSpending',
    'GrocerySpending', 'HomeSpending', 'Visits', 'PurchaseFrequency',
    'OnlineActivity', 'EmailOpens', 'AppUsage', 'LoyaltyPoints'
]
categorical_feature_names = ['Gender', 'MembershipLevel']

# normalize data
scaler = StandardScaler()
numeric_data_scaled = pd.DataFrame(
    scaler.fit_transform(customer_data[numeric_feature_names]),
    columns=numeric_feature_names
)
print(numeric_data_scaled);
```

```
          Age  IncomeLevel  ElectronicsSpending  ClothingSpending  \
0   -2.022622    -1.076784            -1.561765         -1.303146
```

```
1   -2.217499    -0.549924            -1.457846        -1.254796
2   -2.022622    -1.178372            -1.237088        -0.945250
3   -2.412375    -1.252029            -1.413130        -1.262717
4   -1.730308    -0.846171            -1.668460        -1.609342
..        ...          ...                  ...              ...
495  1.290277     1.326355             1.042406         1.756637
496  1.095400     1.262265             1.077671         1.980948
497  1.290277     0.930803             1.400057         1.462284
498  0.900524     1.545417             1.630423         1.561354
499  0.900524     1.239021             1.374612         1.505411

     GrocerySpending  HomeSpending    Visits  PurchaseFrequency  \
0          -0.924782     -1.478844  0.901474           0.302569
1          -0.802603     -1.258900 -0.182897          -1.416574
2          -1.324214     -1.399553 -0.544354          -1.416574
3          -0.584143     -1.884581  0.178560          -0.843527
4          -2.215617     -1.884581  0.178560          -1.416574
..               ...           ...       ...                ...
495        -1.761730      0.188825  0.178560           0.875617
496        -1.194257      1.297081  0.178560          -0.843527
497        -0.198839     -0.014037  0.178560           0.302569
498         0.402067      0.473768 -0.544354           0.875617
499         0.111258      0.274206  0.178560          -0.270479

     OnlineActivity  EmailOpens  AppUsage  LoyaltyPoints
0         -1.622613    0.713398 -0.528785      -1.207362
1         -0.531652    0.760257 -0.479183      -1.180360
2         -0.764920    1.418271 -0.271860      -1.461425
3         -0.368898    0.756074  0.631222      -0.952202
4         -0.367671    1.249417 -0.321725      -1.050133
..              ...         ...       ...            ...
495        1.013074    1.636709  1.074382       1.758903
496        1.910496    1.113341  1.217708       1.834472
497        2.033284    1.493938  0.944123       1.682072
498        1.508789    0.715367  1.077459       1.644895
499        0.841951    0.533986  0.415132       1.468352

[500 rows x 12 columns]
```

```
# encode categorical data (0s or 1s for categories, etc...)
categorical_data_encoded = pd.get_dummies(
    customer_data[categorical_feature_names],
    drop_first=False
);
print(categorical_data_encoded);
```

```
     Gender_Female  Gender_Male  MembershipLevel_Bronze
MembershipLevel_Gold  \
0            False         True                    True
False
```

```
1               True        False                        True
False
2               True        False                       False
False
3               True        False                       False
False
4              False         True                       False
False
..               ...          ...                         ...
...
495            False         True                        True
False
496             True        False                       False
True
497            False         True                        True
False
498             True        False                        True
False
499             True        False                        True
False

     MembershipLevel_Silver
0                     False
1                     False
2                      True
3                      True
4                      True
..                      ...
495                   False
496                   False
497                   False
498                   False
499                   False

[500 rows x 5 columns]

# combine numerical and categorical data
clustering_features = pd.concat([numeric_data_scaled,
categorical_data_encoded], axis=1)
print(clustering_features);

          Age  IncomeLevel  ElectronicsSpending  ClothingSpending  \
0   -2.022622    -1.076784            -1.561765         -1.303146
1   -2.217499    -0.549924            -1.457846         -1.254796
2   -2.022622    -1.178372            -1.237088         -0.945250
3   -2.412375    -1.252029            -1.413130         -1.262717
4   -1.730308    -0.846171            -1.668460         -1.609342
..        ...          ...                  ...               ...
495  1.290277     1.326355             1.042406          1.756637
496  1.095400     1.262265             1.077671          1.980948
```

```
497  1.290277     0.930803             1.400057          1.462284
498  0.900524     1.545417             1.630423          1.561354
499  0.900524     1.239021             1.374612          1.505411

     GrocerySpending  HomeSpending    Visits  PurchaseFrequency  \
0          -0.924782     -1.478844  0.901474           0.302569
1          -0.802603     -1.258900 -0.182897          -1.416574
2          -1.324214     -1.399553 -0.544354          -1.416574
3          -0.584143     -1.884581  0.178560          -0.843527
4          -2.215617     -1.884581  0.178560          -1.416574
..               ...           ...       ...                ...
495        -1.761730      0.188825  0.178560           0.875617
496        -1.194257      1.297081  0.178560          -0.843527
497        -0.198839     -0.014037  0.178560           0.302569
498         0.402067      0.473768 -0.544354           0.875617
499         0.111258      0.274206  0.178560          -0.270479

     OnlineActivity  EmailOpens  AppUsage  LoyaltyPoints
Gender_Female  \
0         -1.622613    0.713398 -0.528785      -1.207362
False
1         -0.531652    0.760257 -0.479183      -1.180360
True
2         -0.764920    1.418271 -0.271860      -1.461425
True
3         -0.368898    0.756074  0.631222      -0.952202
True
4         -0.367671    1.249417 -0.321725      -1.050133
False
..              ...         ...       ...            ...              ..
.
495        1.013074    1.636709  1.074382       1.758903
False
496        1.910496    1.113341  1.217708       1.834472
True
497        2.033284    1.493938  0.944123       1.682072
False
498        1.508789    0.715367  1.077459       1.644895
True
499        0.841951    0.533986  0.415132       1.468352
True

     Gender_Male  MembershipLevel_Bronze  MembershipLevel_Gold  \
0           True                    True                 False
1          False                    True                 False
2          False                   False                 False
3          False                   False                 False
4           True                   False                 False
..           ...                     ...                   ...
495         True                    True                 False
```

```
496        False                       False                    True
497         True                        True                   False
498        False                        True                   False
499        False                        True                   False

     MembershipLevel_Silver
0                     False
1                     False
2                      True
3                      True
4                      True
..                      ...
495                   False
496                   False
497                   False
498                   False
499                   False

[500 rows x 17 columns]

k_vals = [2, 3, 4, 5];

# kmeans
for k in k_vals:

    print("-----------------------------");
    print(f"analysis for K = {k} clusters");

    kmeans_model = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans_model.fit(clustering_features)

    cluster_labels = kmeans_model.labels_

    analysis_df = customer_data.copy()
    analysis_df['Cluster_ID'] = cluster_labels

    for cluster_id in range(k):
        # filter data for the current cluster
        cluster_segment = analysis_df[analysis_df['Cluster_ID'] ==
cluster_id]

        customer_count = len(cluster_segment)

        # calc key metrics
        avg_income = cluster_segment['IncomeLevel'].mean()
        avg_loyalty = cluster_segment['LoyaltyPoints'].mean()

        # calc spending habits
        avg_electronics =
cluster_segment['ElectronicsSpending'].mean()
```

```
        avg_clothing = cluster_segment['ClothingSpending'].mean()
        avg_grocery = cluster_segment['GrocerySpending'].mean()
        avg_home = cluster_segment['HomeSpending'].mean()

        print(f"\n[ Cluster {cluster_id} ] - {customer_count}
Customers")
        print(f"  -> avg income:         ${avg_income:,.2f}")
        print(f"  -> avg loyalty points: {avg_loyalty:.2f}")
        print(f"  -> avg spending:")
        print(f"      - electronics: ${avg_electronics:,.2f}")
        print(f"      - clothing:    ${avg_clothing:,.2f}")
        print(f"      - grocery:     ${avg_grocery:,.2f}")
        print(f"      - home:        ${avg_home:,.2f}")

----------------------------
analysis for K = 2 clusters

[ Cluster 0 ] - 200 Customers
  -> avg income:         $44,598.70
  -> avg loyalty points: 161.79
  -> avg spending:
      - electronics: $728.51
      - clothing:    $546.14
      - grocery:     $236.86
      - home:        $291.79

[ Cluster 1 ] - 300 Customers
  -> avg income:         $75,503.31
  -> avg loyalty points: 404.43
  -> avg spending:
      - electronics: $1,364.83
      - clothing:    $866.32
      - grocery:     $397.96
      - home:        $825.79
----------------------------
analysis for K = 3 clusters

[ Cluster 0 ] - 100 Customers
  -> avg income:         $82,392.57
  -> avg loyalty points: 539.76
  -> avg spending:
      - electronics: $1,617.17
      - clothing:    $1,320.66
      - grocery:     $251.12
      - home:        $824.88

[ Cluster 1 ] - 200 Customers
  -> avg income:         $44,598.70
  -> avg loyalty points: 161.79
  -> avg spending:
```

```
      - electronics: $728.51
      - clothing:    $546.14
      - grocery:     $236.86
      - home:        $291.79

[ Cluster 2 ] - 200 Customers
  -> avg income:         $72,058.68
  -> avg loyalty points: 336.77
  -> avg spending:
      - electronics: $1,238.66
      - clothing:    $639.15
      - grocery:     $471.37
      - home:        $826.24
----------------------------
analysis for K = 4 clusters

[ Cluster 0 ] - 100 Customers
  -> avg income:         $43,704.85
  -> avg loyalty points: 181.65
  -> avg spending:
      - electronics: $969.13
      - clothing:    $911.21
      - grocery:     $275.09
      - home:        $448.76

[ Cluster 1 ] - 200 Customers
  -> avg income:         $72,058.68
  -> avg loyalty points: 336.77
  -> avg spending:
      - electronics: $1,238.66
      - clothing:    $639.15
      - grocery:     $471.37
      - home:        $826.24

[ Cluster 2 ] - 100 Customers
  -> avg income:         $45,492.55
  -> avg loyalty points: 141.93
  -> avg spending:
      - electronics: $487.90
      - clothing:    $181.07
      - grocery:     $198.62
      - home:        $134.81

[ Cluster 3 ] - 100 Customers
  -> avg income:         $82,392.57
  -> avg loyalty points: 539.76
  -> avg spending:
      - electronics: $1,617.17
      - clothing:    $1,320.66
      - grocery:     $251.12
```

```
      - home:        $824.88
----------------------------
analysis for K = 5 clusters

[ Cluster 0 ] - 100 Customers
  -> avg income:         $43,704.85
  -> avg loyalty points: 181.65
  -> avg spending:
      - electronics: $969.13
      - clothing:    $911.21
      - grocery:     $275.09
      - home:        $448.76

[ Cluster 1 ] - 100 Customers
  -> avg income:         $60,324.04
  -> avg loyalty points: 368.41
  -> avg spending:
      - electronics: $898.78
      - clothing:    $768.02
      - grocery:     $491.03
      - home:        $619.98

[ Cluster 2 ] - 100 Customers
  -> avg income:         $45,492.55
  -> avg loyalty points: 141.93
  -> avg spending:
      - electronics: $487.90
      - clothing:    $181.07
      - grocery:     $198.62
      - home:        $134.81

[ Cluster 3 ] - 100 Customers
  -> avg income:         $82,392.57
  -> avg loyalty points: 539.76
  -> avg spending:
      - electronics: $1,617.17
      - clothing:    $1,320.66
      - grocery:     $251.12
      - home:        $824.88

[ Cluster 4 ] - 100 Customers
  -> avg income:         $83,793.32
  -> avg loyalty points: 305.13
  -> avg spending:
      - electronics: $1,578.53
      - clothing:    $510.28
      - grocery:     $451.72
      - home:        $1,032.49
```