In [4]:
```
random.seed(42)
Customer_ID = []
age = []
gender = []
MaritalStatuss = []
AnnualIncome = []
TotalPurchase = []
PreferedCategory = []
```

In [5]:
```
for CustomerIDs in range(1001, 1501):
    Customer_ID.append(CustomerIDs)
    age.append(random.randint(18, 65))
    gender.append(random.choice(['Male', 'Female']))
    MaritalStatuss.append(random.choice(['Married', 'Single', 'Divorced']))
    AnnualIncome.append(random.randint(25000, 90000))
    TotalPurchase.append(random.randint(18, 90))
    PreferedCategory.append(random.choice(['Electronics', 'Appliances']))
```

In [6]:
```
data = {
    'CustomerID': Customer_ID,
    'Age': age,
    'Gender': gender,
    'MaritalStatus': MaritalStatuss,
    'AnnualIncome (USD)': AnnualIncome,
    'TotalPurchases': TotalPurchase,
    'PreferredCategory': PreferedCategory
}
```

In [7]:
```
df = pd.DataFrame(data)
df.to_csv('TechElectroCustomerData.csv', index=False)
```

In [8]:
```
df.head()
```

Out[8]:

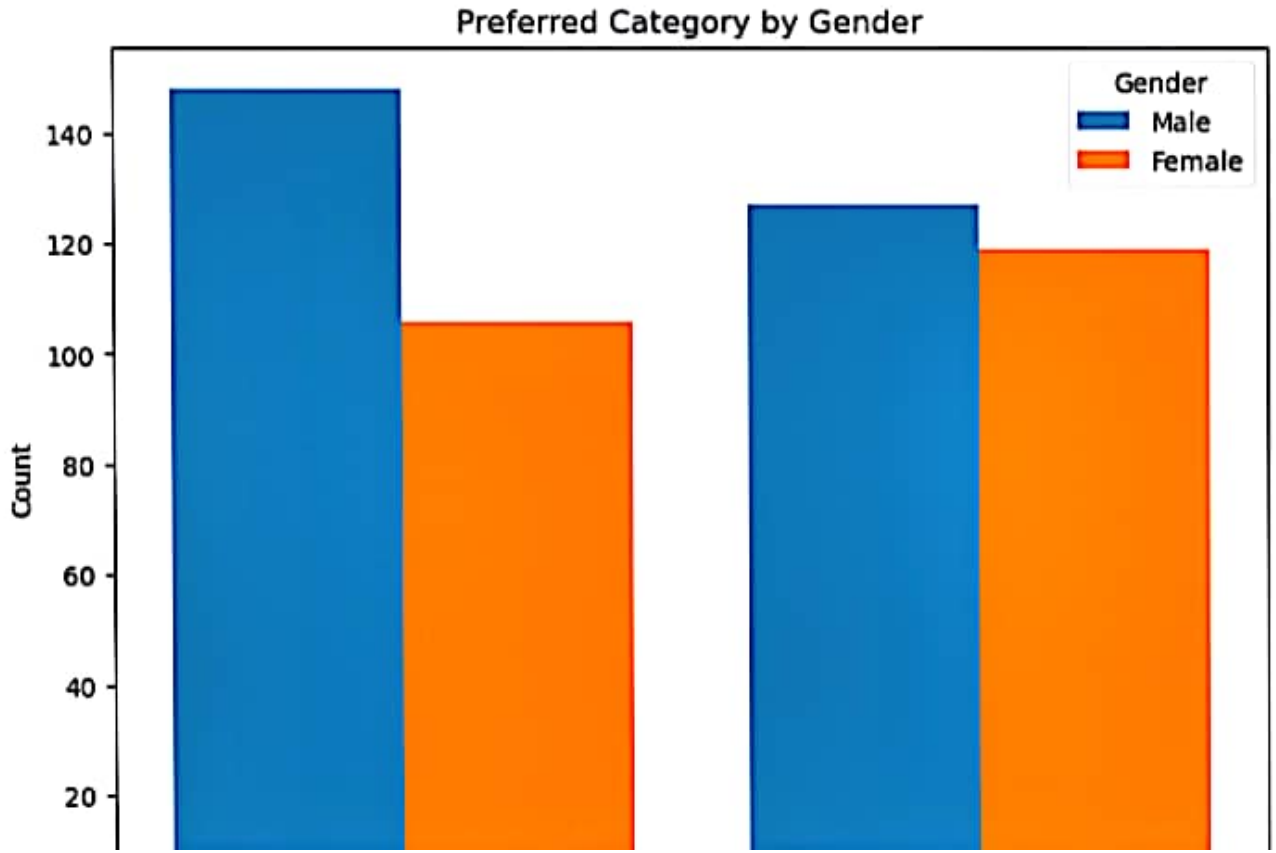
	CustomerID	Age	Gender	MaritalStatus	AnnualIncome (USD)	TotalPurchases	PreferredCategory
0	1001	58	Male	Married	73598	53	Electronics
1	1002	32	Male	Divorced	31717	87	Electronics
2	1003	55	Female	Married	26952	29	Electronics
3	1004	32	Male	Divorced	38031	87	Appliances
4	1005	32	Female	Divorced	43231	18	Electronics

In [9]:
```
num_cols = ['Age', 'AnnualIncome (USD)', 'TotalPurchases']
```

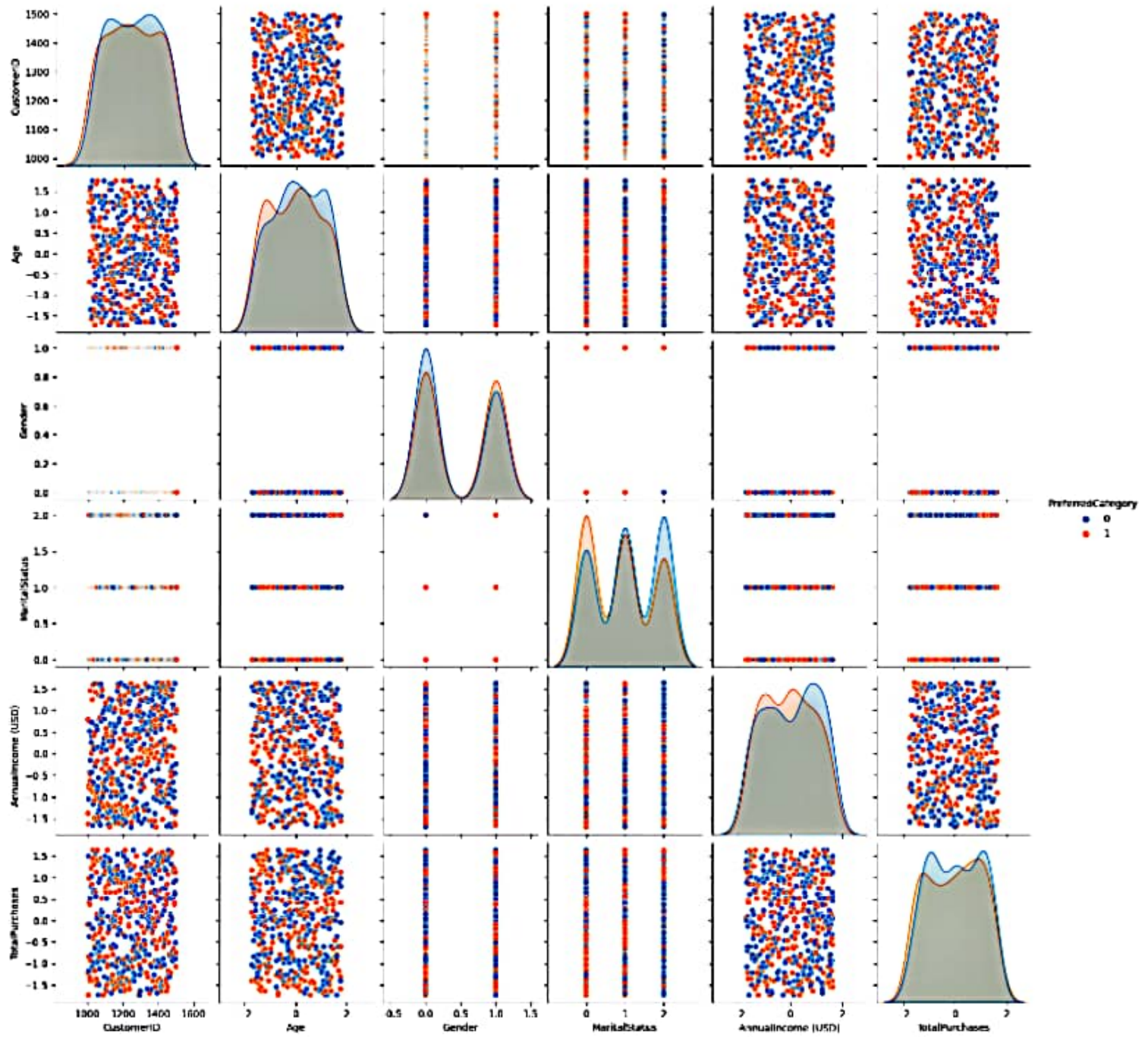
In [9]:
```
num_cols = ['Age', 'AnnualIncome (USD)', 'TotalPurchases']
scaler = StandardScaler()
df[num_cols] = scaler.fit_transform(df[num_cols])
```

In [10]:
```
l = LabelEncoder()
df['Gender'] = l.fit_transform(df['Gender'])
df['MaritalStatus'] = l.fit_transform(df['MaritalStatus'])
df['PreferredCategory'] = l.fit_transform(df['PreferredCategory'])
```

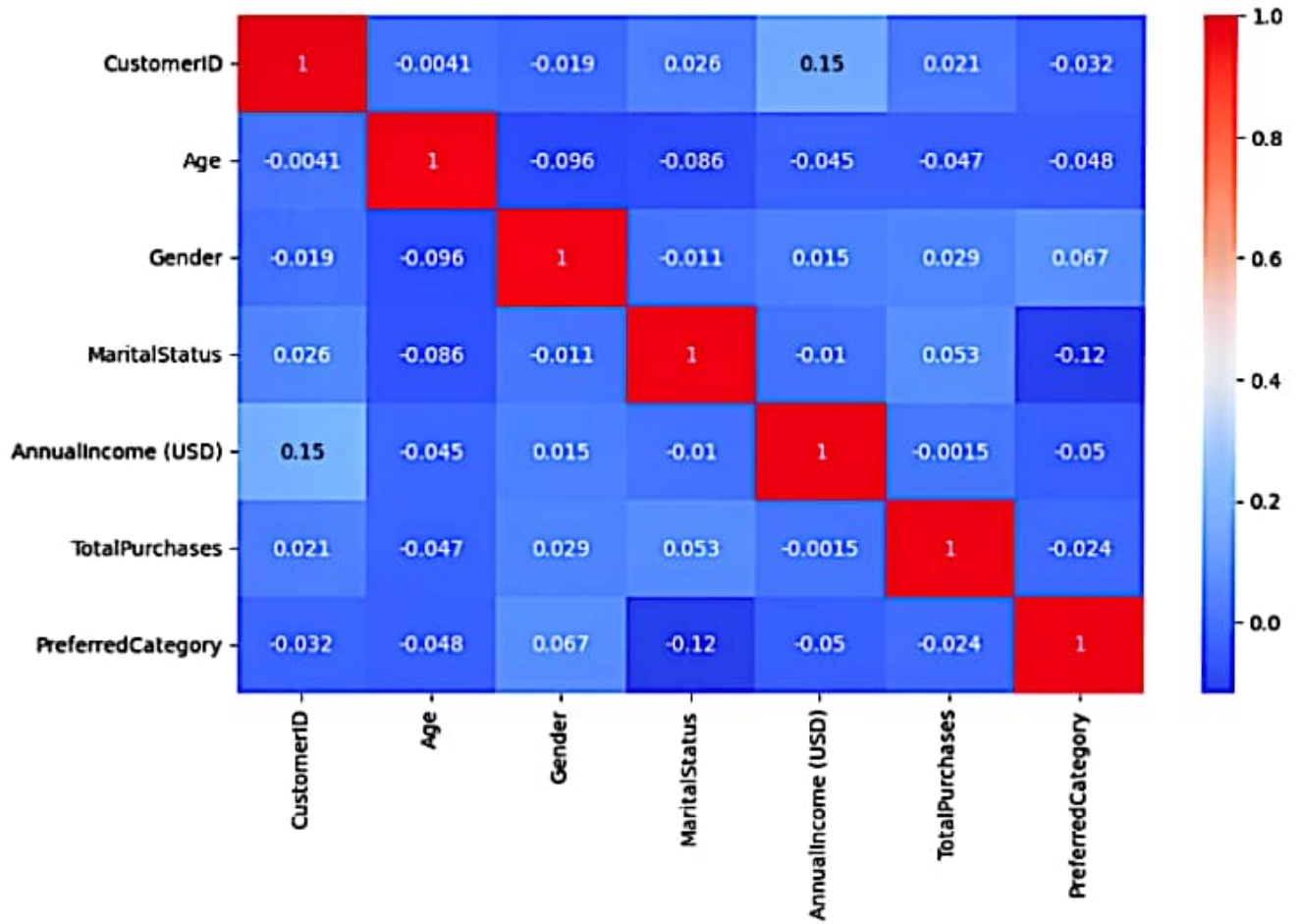
In [11]:
```
#Exploratory Data Analysis
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='PreferredCategory', hue='Gender')
plt.title('Preferred Category by Gender')
plt.xlabel('Preferred Category')
plt.ylabel('Count')
plt.legend(title='Gender', loc='upper right', labels=['Male', 'Female'])
plt.show()
```



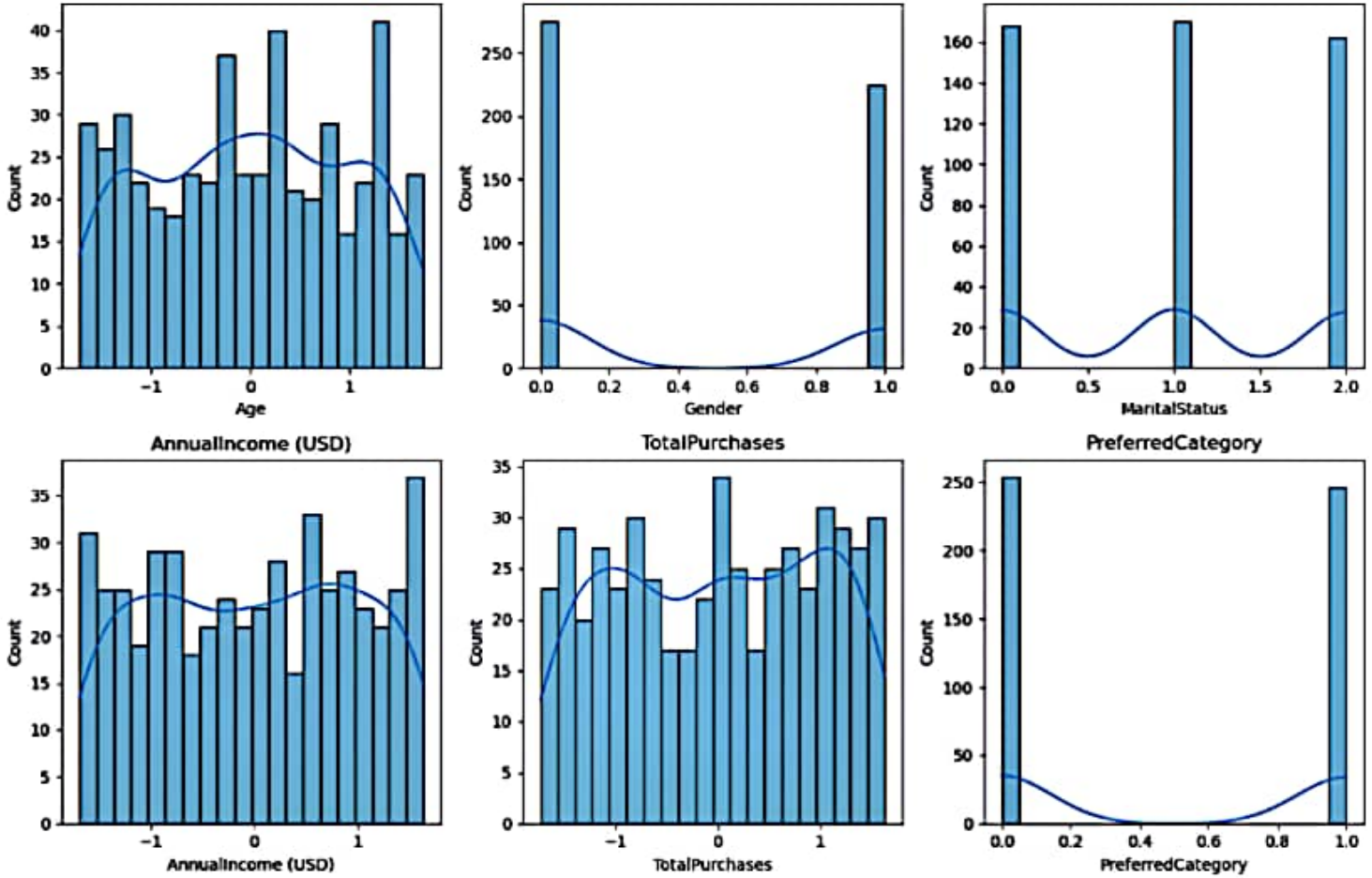
```
In [12]: # Pairplot for visualizing relationships
sns.pairplot(df, hue='PreferredCategory')
plt.show()
```



In [13]:
```
# Correlation heatmap
correlation_matrix = df.corr()
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.show()
```



In [14]:
```
# Distribution plots for numerical features
plt.figure(figsize=(12, 8))
for i, column in enumerate(df.columns[1:]):
    plt.subplot(2, 3, i + 1)
    sns.histplot(df[column], bins=20, kde=True)
    plt.title(column)
plt.tight_layout()
plt.show()
```



In [15]:
```
# Feature scaling/normalization
scaler = StandardScaler()
data_scaled = scaler.fit_transform(df.drop(['CustomerID'], axis=1))
```

In [18]:
```
from sklearn.cluster import KMeans

# Determine the optimal number of clusters using the elbow method
inertia = []
for n_clusters in range(1, 11):
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    kmeans.fit(data_scaled)
    inertia.append(kmeans.inertia_)

# Plot the elbow method
plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), inertia, marker='o')
plt.title('Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
```

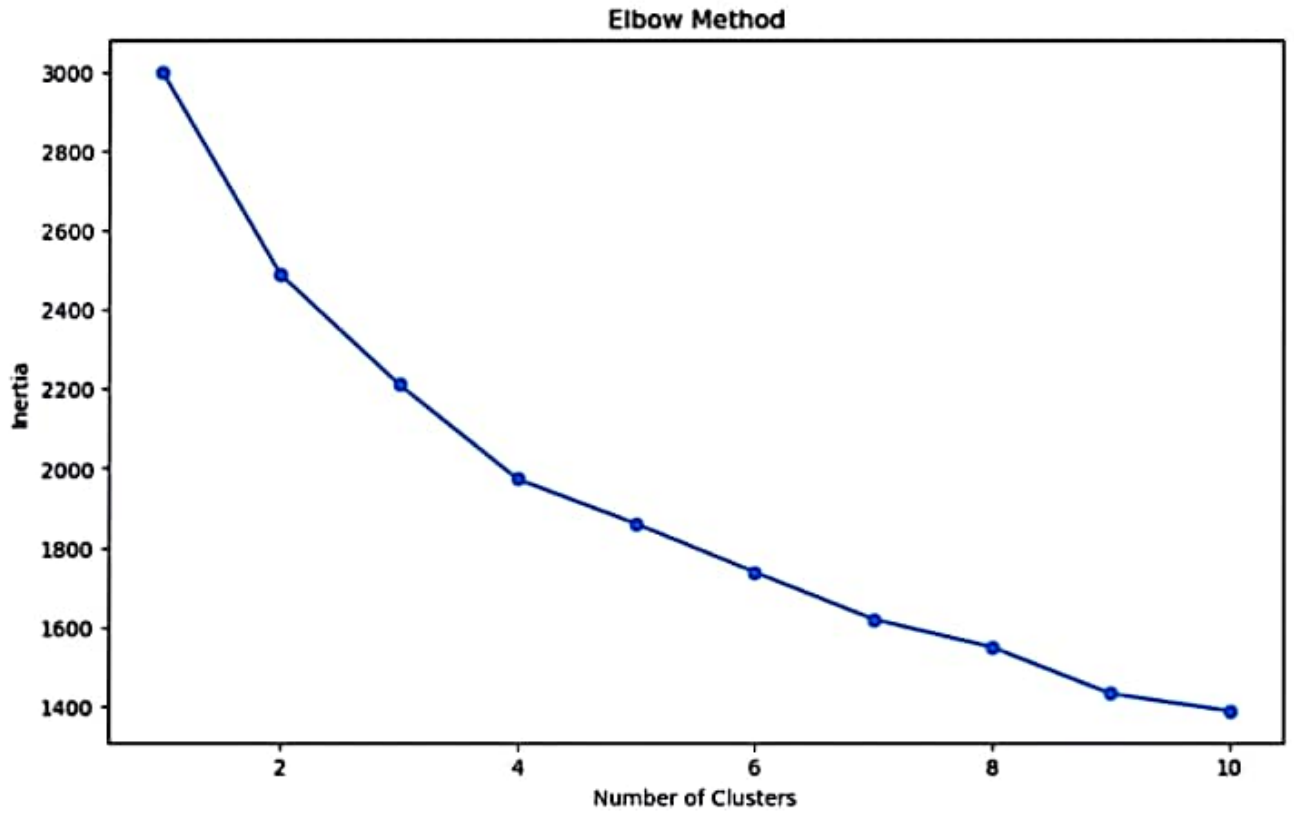
In [18]:
```
from sklearn.cluster import KMeans

# Determine the optimal number of clusters using the elbow method
inertia = []
for n_clusters in range(1, 11):
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    kmeans.fit(data_scaled)
    inertia.append(kmeans.inertia_)

# Plot the elbow method
plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), inertia, marker='o')
plt.title('Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.show()

# Based on the elbow method, choose the optimal number of clusters
optimal_clusters = 3

# Apply K-means clustering
kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)
df['Cluster'] = kmeans.fit_predict(data_scaled)
```



```python
# #pip install streamlit
# #streamlit run techelectro_customer_data.py
import streamlit as st
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer

# Load the preprocessed DataFrame
df = pd.read_csv('TechElectroCustomerData.csv')

# Apply K-means clustering
optimal_clusters = 3
num_cols = ['Age', 'AnnualIncome (USD)', 'TotalPurchases']
data_scaled = StandardScaler().fit_transform(df[num_cols])
kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)
df['Cluster'] = kmeans.fit_predict(data_scaled)

# Perform one-hot encoding for categorical columns
categorical_cols = ['Gender', 'MaritalStatus', 'PreferredCategory']
transformer = ColumnTransformer(
    transformers=[('cat', OneHotEncoder(), categorical_cols)],
    remainder='passthrough'
)
data_encoded = transformer.fit_transform(df)
df_encoded = pd.DataFrame(data_encoded, columns = transformer.get_feature_names_out(df.columns))

# Streamlit Dashboard
st.title('TechElectro Customer Segmentation Dashboard')

# Sidebar filters
st.sidebar.title('Filters')
selected_cluster = st.sidebar.selectbox('Select a Cluster', df['Cluster'].unique())
selected_gender = st.sidebar.selectbox('Select Gender', df['Gender'].unique())
selected_category = st.sidebar.selectbox('Select Preferred Category', df['PreferredCategory'].unique())
```

```
29
30 # Streamlit Dashboard
31 st.title('TechElectro Customer Segmentation Dashboard')
32
33 # Sidebar filters
34 st.sidebar.title('Filters')
35 selected_cluster = st.sidebar.selectbox('Select a Cluster', df['Cluster'].unique())
36 selected_gender = st.sidebar.selectbox('Select Gender', df['Gender'].unique())
37 selected_category = st.sidebar.selectbox('Select Preferred Category', df['PreferredCategory'].unique())
38 # Display EDA Visualizations
39 st.header('Exploratory Data Analysis')
40
41
42
43 # Count plot of Preferred Category by Gender
44 plt.figure(figsize=(8, 6))
45 sns.countplot(data=df, x='PreferredCategory', hue='Gender')
46 plt.title('Preferred Category by Gender')
47 plt.xlabel('Preferred Category')
48 plt.ylabel('Count')
49 plt.legend(title='Gender', loc='upper right', labels=['Male', 'Female'])
50 # Pass the figure to st.pyplot()
51 st.pyplot(plt.gcf())  # Get the current figure
52
53 # Correlation heatmap
54 correlation_matrix = df.corr()
55 plt.figure(figsize=(10, 6))
56 sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
57 # Pass the figure to st.pyplot()
58 st.pyplot(plt.gcf())  # Get the current figure
59
60 # Distribution plots for numerical features
61 plt.figure(figsize=(12, 8))
62 for column in num_cols:
63     sns.histplot(df[column], bins=20, kde=True)
64     plt.title(column)
65     # Pass the figure to st.pyplot()
66     st.pyplot(plt.gcf())  # Get the current figure
67
68 # Display Customer Segmentation Results
69 st.header('Customer Segmentation')
70 st.subheader('Cluster Distribution')
71 cluster_counts = df['Cluster'].value_counts()
72 st.bar_chart(cluster_counts)
73
```

techelectro-customer-data · Stre | Music/internship/week 5/day1/ | techelectro-main - Jupyter Notel | techelectro-customer-data.py - J | techelectro-customer-data · Stre

localhost:8501

Filters

Select a Cluster

0

Select Gender

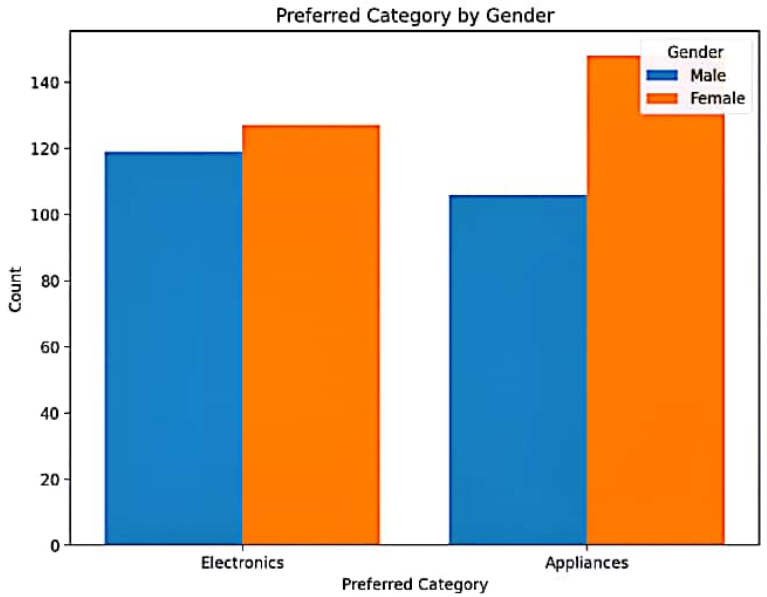
Male

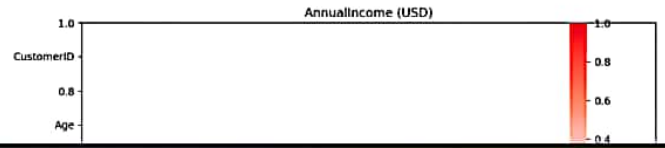
Select Preferred Category

Electronics

TechElectro Customer Segmentation Dashboard

Exploratory Data Analysis





Type here to search

29°C Smoke

2:03 AM
8/23/2023

localhost:8501

Filters

Select a Cluster

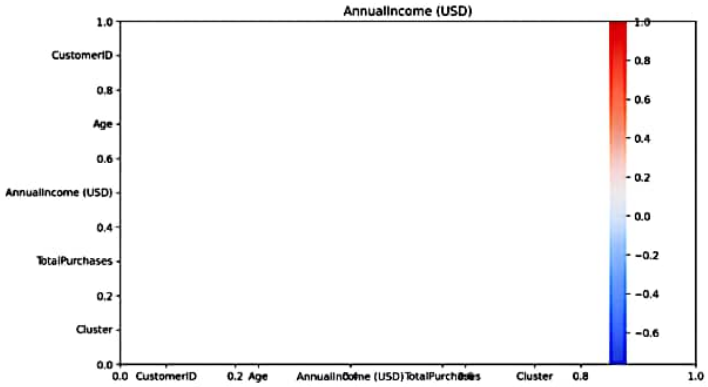
0

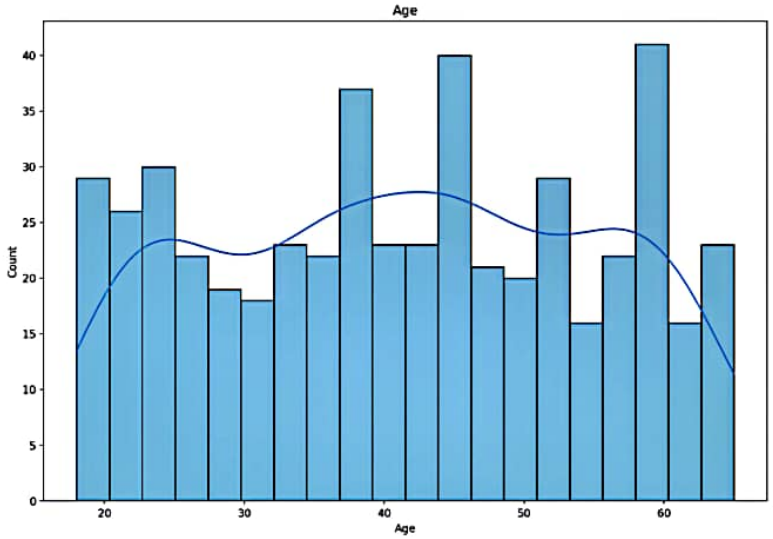
Select Gender

Male

Select Preferred Category

Electronics





AnnualIncome (USD)

techelectro-customer-data · Stre | Music/internship/week 5/day1/ | techelectro-main - Jupyter Note | techelectro-customer-data.py - J | techelectro-customer-data · Stre

localhost:8501

Filters

Select a Cluster

0

Select Gender

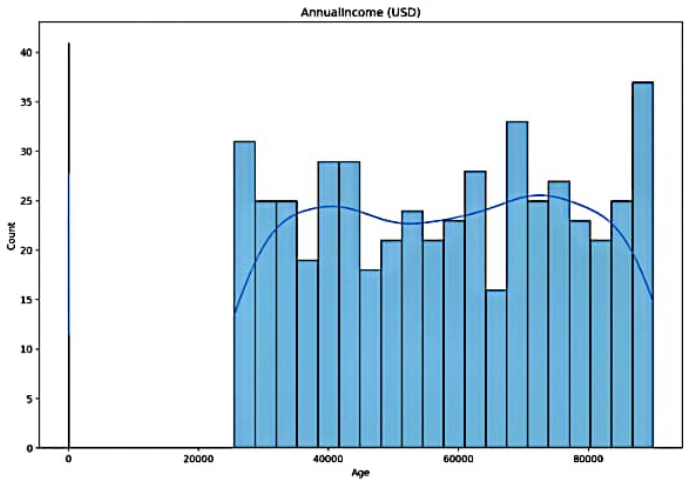
Male

Select Preferred Category

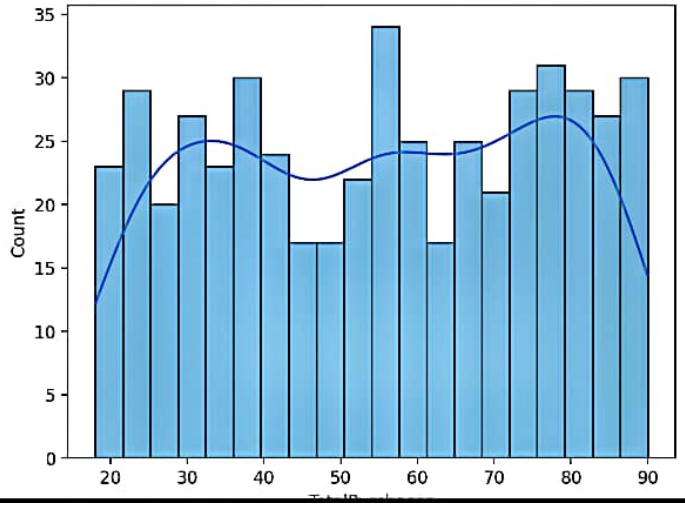
Electronics

Age

AnnualIncome (USD)



TotalPurchases



techelectro-customer-data · Stre | Music/internship/week 5/day1/ | techelectro-main - Jupyter Notel | techelectro-customer-data.py - J | techelectro-customer-data · Stre

localhost:8501

Filters

Select a Cluster

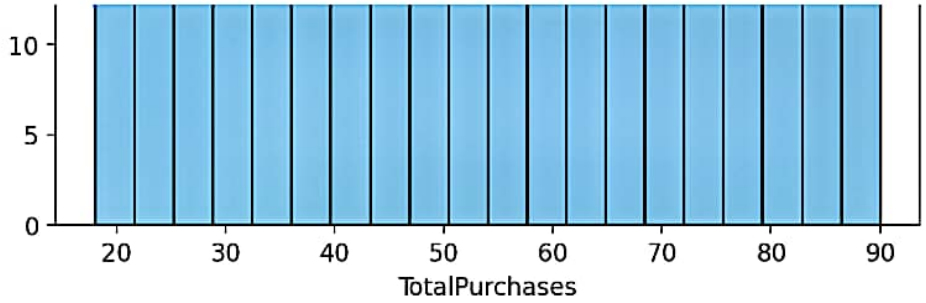
0

Select Gender

Male

Male

Female



Customer Segmentation

Cluster Distribution