DHANRAJ RAJESH TUNGAR
PRN - 2022018100094671

DATA ANALYTICS MINI PROJECT

(Customer behavior prediction)

```
!pip install pandas numpy matplotlib seaborn faker scikit-learn

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from faker import Faker
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
np.random.seed(42)
fake = Faker()
```

```
categories = ["Electronics", "Fashion", "Groceries", "Entertainment"]
data = {
    "Customer_ID": range(1, 101),
    "Age": np.random.randint(18, 65, 100),
    "Gender": np.random.choice(["Male", "Female"], 100),
    "City": np.random.choice(["New York", "Los Angeles", "Chicago", "Houston"], 100),
    "Annual_Income": np.random.randint(30000, 150000, 100),
    "Spending_Score": np.random.randint(1, 100, 100),
    "Purchase_Category": np.random.choice(categories, 100)
}

df = pd.DataFrame(data)
df.to_csv("customer_data.csv", index=False)
df.head()
```

	Customer_ID	Age	Gender	City	Annual_Income	Spending_Score	Purchase_Category
0	1	57	Male	Chicago	124270	16	Fashion
1	2	54	Female	Los Angeles	34757	68	Entertainment
2	3	53	Female	Los Angeles	49097	37	Fashion
3	4	41	Female	Chicago	40395	54	Groceries
4	5	48	Male	Houston	46241	85	Electronics

CUSTOMER BEHAVIOR PREDICTION

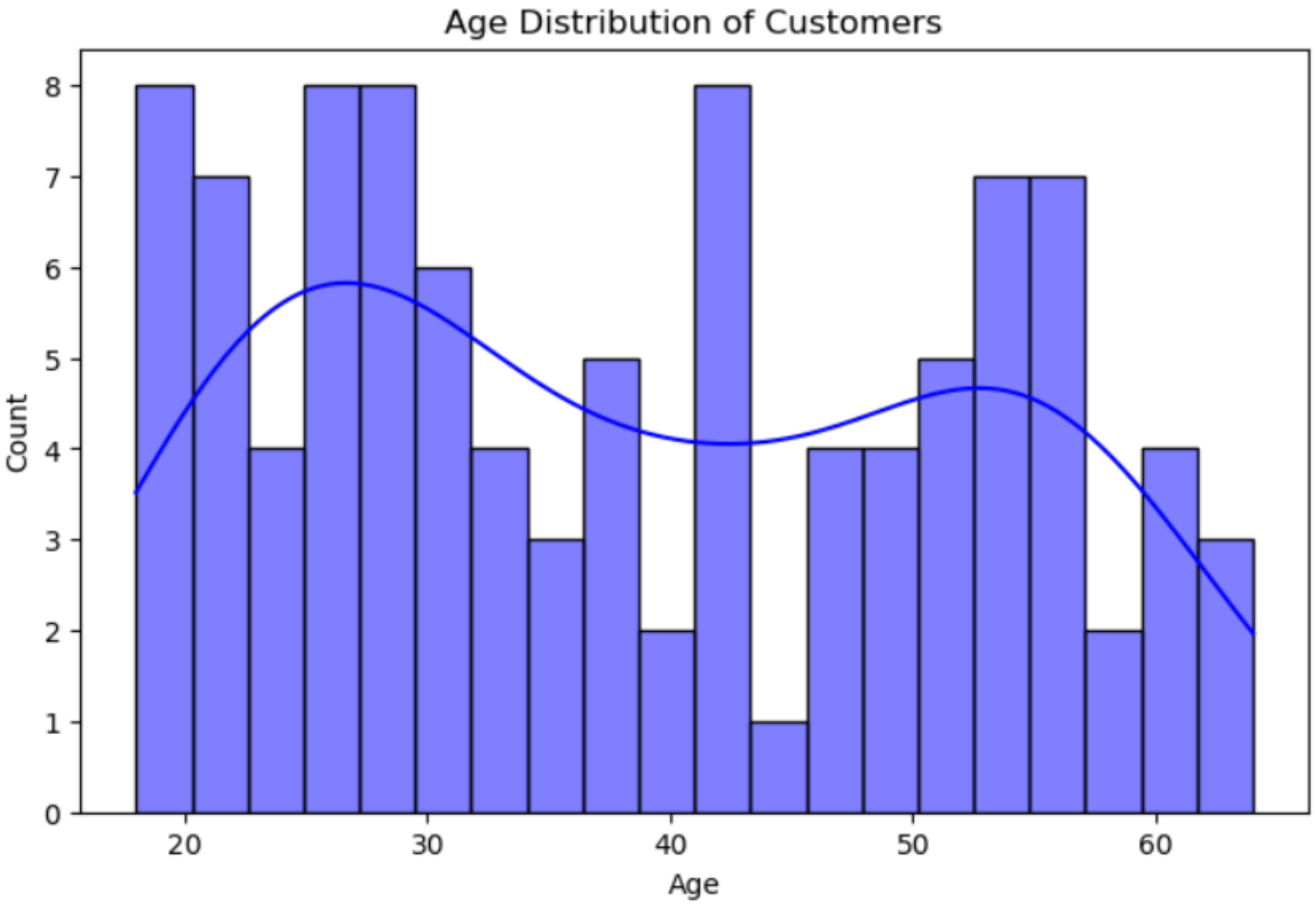
[75]:
```
df = pd.read_csv("customer_data.csv")
print(df.head())
print("Dataset Shape:", df.shape)
print(df.info())
print("Missing Values:\n", df.isnull().sum())
print("Summary Statistics:\n", df.describe())
```

```
   Customer_ID  Age  Gender         City  Annual_Income  Spending_Score  \
0            1   57    Male      Chicago         124270              16
1            2   54  Female  Los Angeles          34757              68
2            3   53  Female  Los Angeles          49097              37
3            4   41  Female      Chicago          40395              54
4            5   48    Male      Houston          46241              85

  Purchase_Category
0           Fashion
1     Entertainment
2           Fashion
3         Groceries
4       Electronics
Dataset Shape: (100, 7)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Customer_ID        100 non-null    int64
 1   Age                100 non-null    int64
 2   Gender             100 non-null    object
 3   City               100 non-null    object
 4   Annual_Income      100 non-null    int64
 5   Spending_Score     100 non-null    int64
 6   Purchase_Category  100 non-null    object
dtypes: int64(4), object(3)
memory usage: 5.6+ KB
None
```

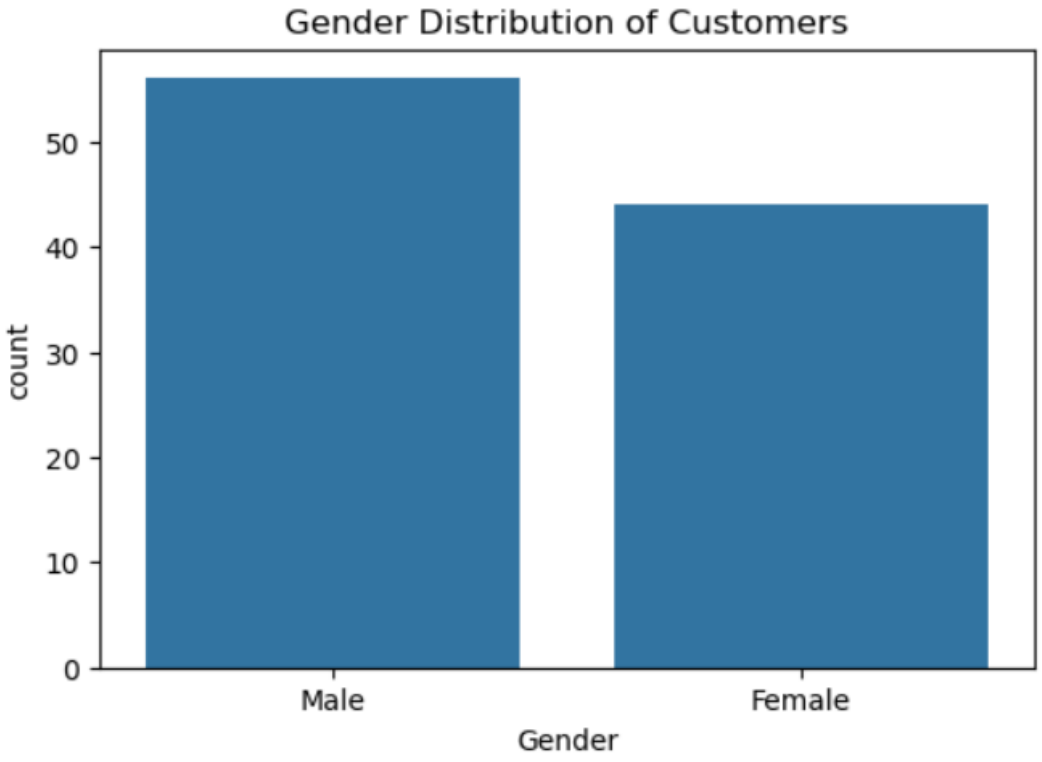
```
Missing Values:
 Customer_ID          0
Age                   0
Gender                0
City                  0
Annual_Income         0
Spending_Score        0
Purchase_Category     0
dtype: int64
Summary Statistics:
        Customer_ID         Age  Annual_Income  Spending_Score
count   100.000000  100.000000     100.000000      100.000000
mean     50.500000   38.540000   90324.050000       48.790000
std      29.011492   13.807786   33485.684339       30.031732
min       1.000000   18.000000   31759.000000        1.000000
25%      25.750000   26.750000   61279.500000       22.750000
50%      50.500000   37.000000   91679.000000       44.500000
75%      75.250000   52.000000  115807.250000       77.250000
max     100.000000   64.000000  149771.000000       99.000000
```

```
df.drop_duplicates(inplace=True)


df.fillna(df.select_dtypes(include=[np.number]).mean(), inplace=True)


df.rename(columns={"Annual_Income": "Income", "Spending_Score": "Score"}, inplace=True)
```

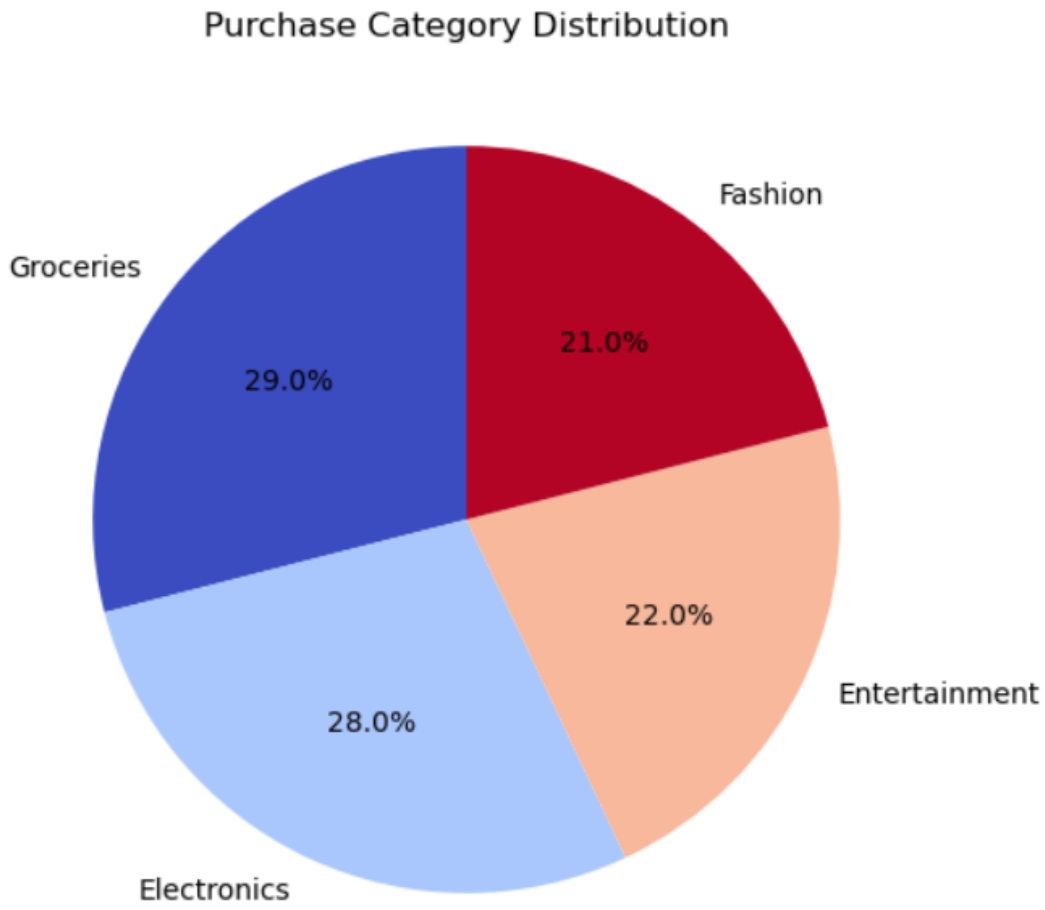
```
plt.figure(figsize=(8,5))
sns.histplot(df["Age"], bins=20, kde=True, color="blue")
plt.title("Age Distribution of Customers")
plt.show()


plt.figure(figsize=(8,5))
sns.scatterplot(x=df["Income"], y=df["Score"], hue=df["Gender"])
plt.title("Spending Score vs Annual Income")
plt.show()

plt.figure(figsize=(6,4))
sns.countplot(x=df["Gender"])
plt.title("Gender Distribution of Customers")
plt.show()


plt.figure(figsize=(6,6))
df["Purchase_Category"].value_counts().plot.pie(autopct="%1.1f%%", startangle=90, cmap="coolwarm")
plt.title("Purchase Category Distribution")
plt.ylabel("")
plt.show()
```



CUSTOMER BEHAVIOR PREDICTION



CUSTOMER BEHAVIOR PREDICTION

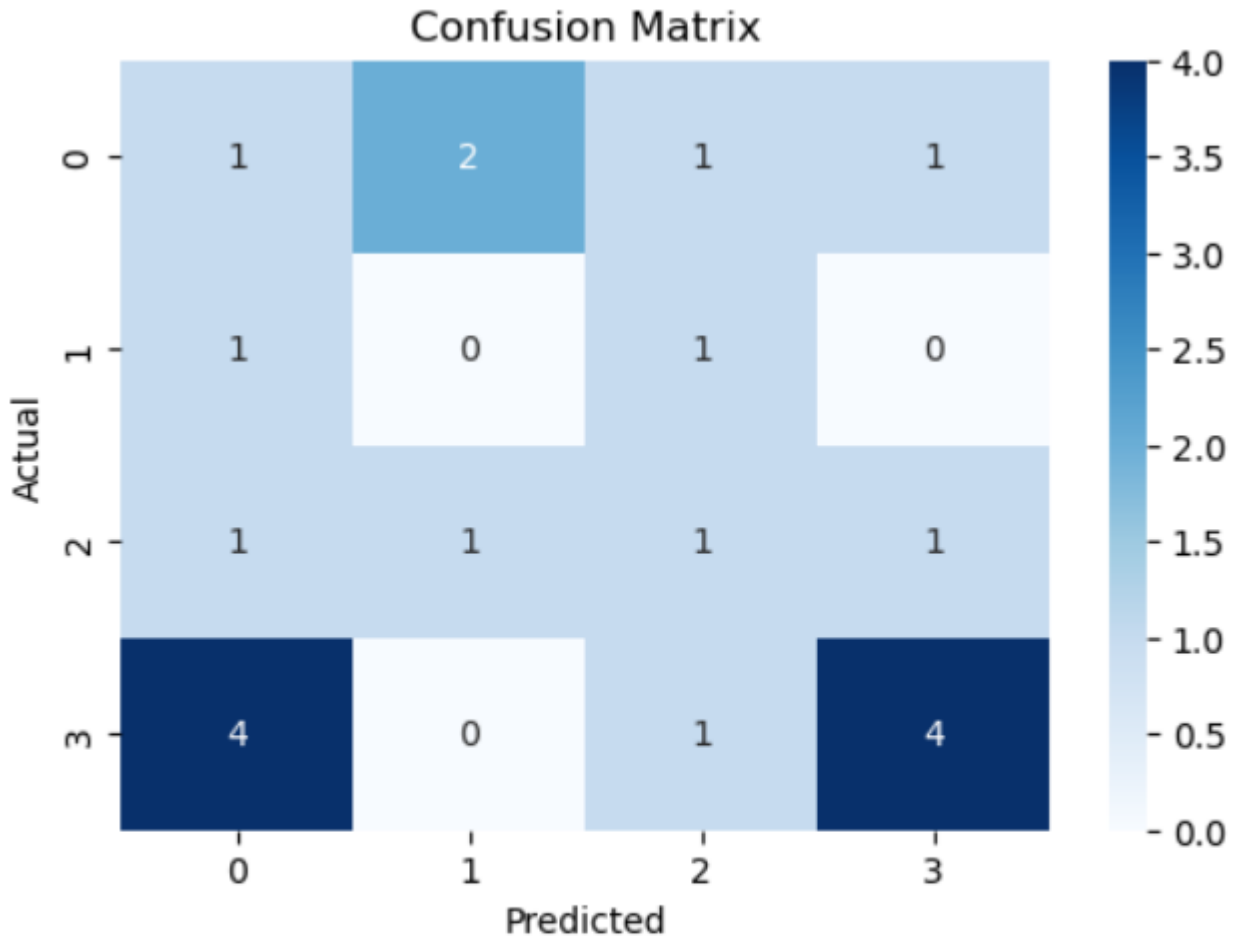


DHANRAJ RAJESH TUNGAR
PRN - 2022018100094671

CUSTOMER BEHAVIOR PREDICTION

```
df["Gender"] = df["Gender"].map({"Male": 0, "Female": 1})
df["Purchase_Category"] = df["Purchase_Category"].astype("category").cat.codes


X = df[["Age", "Income", "Score", "Gender"]]
y = df["Purchase_Category"]


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)


y_pred = model.predict(X_test)


print("Model Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))


plt.figure(figsize=(6,4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```

```
Model Accuracy: 0.3
Classification Report:
              precision    recall  f1-score   support

           0       0.14      0.20      0.17         5
           1       0.00      0.00      0.00         2
           2       0.25      0.25      0.25         4
           3       0.67      0.44      0.53         9

    accuracy                           0.30        20
   macro avg       0.26      0.22      0.24        20
weighted avg       0.39      0.30      0.33        20
```




```
df.to_csv("Final_Customer_Data.csv", index=False)
print("Project Completed! File Saved Successfully.")
```

```
Project Completed! File Saved Successfully.
```

DHANRAJ RAJESH TUNGAR
PRN - 2022018100094671

CUSTOMER BEHAVIOR PREDICTION

PROF : Shaikh Mohammed Umar