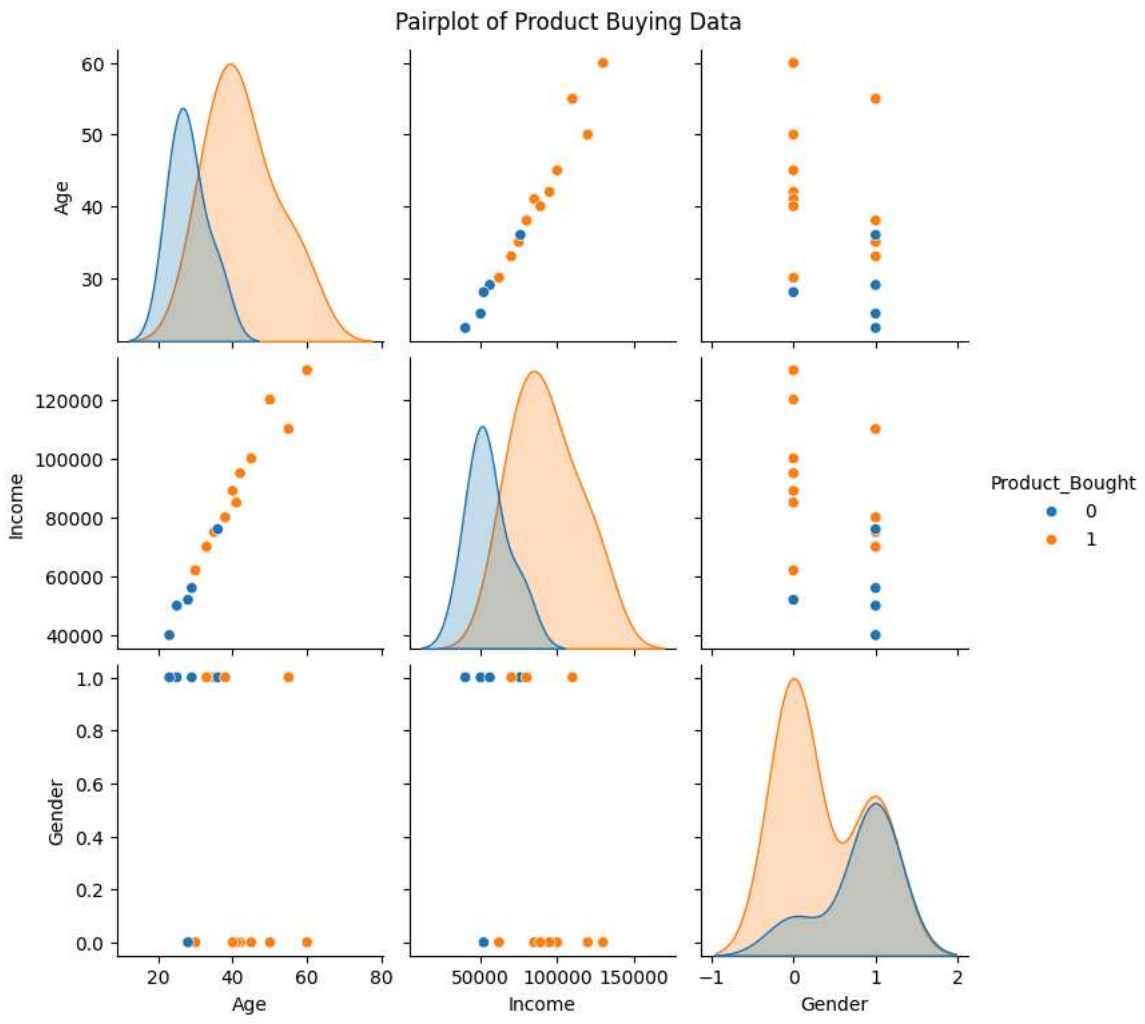
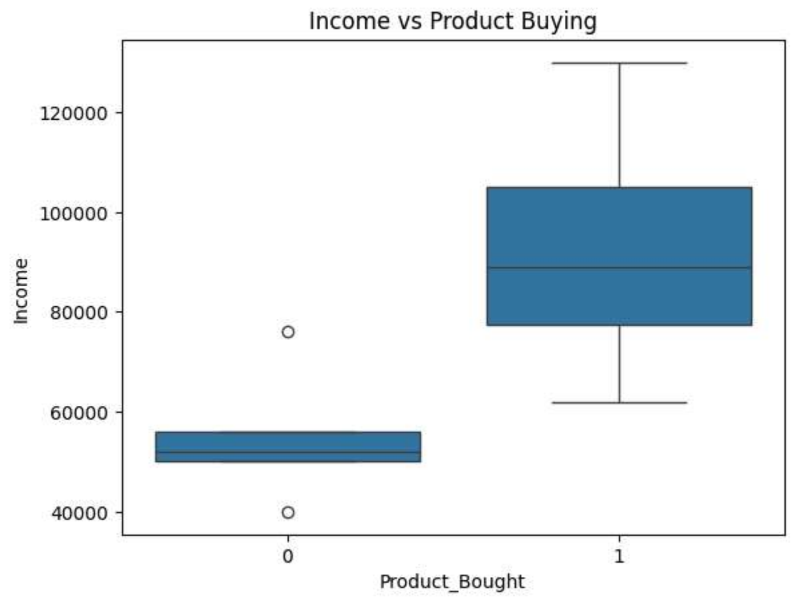
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
```

```
# Sample data
data = {
    'Age': [25, 45, 35, 50, 23, 42, 36, 29, 30, 41, 55, 60, 33, 28, 40, 38],
    'Income': [50000, 100000, 75000, 120000, 40000, 95000, 76000, 56000,
               62000, 85000, 110000, 130000, 70000, 52000, 89000, 80000],
    'Gender': [1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1],
    'Product_Bought': [0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1]
}
df = pd.DataFrame(data)
```

```
# Visualize product buying behavior
sns.pairplot(df, hue='Product_Bought')
plt.suptitle("Pairplot of Product Buying Data", y=1.02)
plt.show()
```



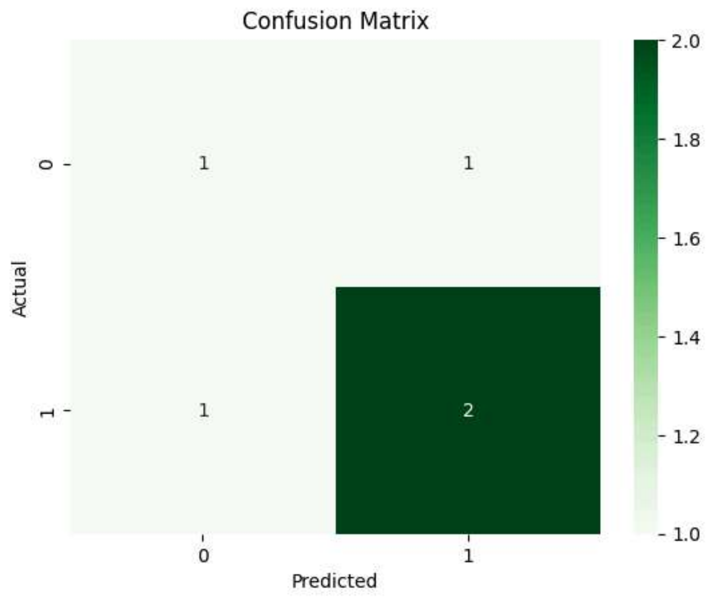
```
sns.boxplot(x='Product_Bought', y='Income', data=df)
plt.title("Income vs Product Buying")
plt.show()
```



```
# Features and target
X = df.drop('Product_Bought', axis=1)
y = df['Product_Bought']


# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)



from sklearn.preprocessing import StandardScaler
# Scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)


# Train model
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# Evaluate
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Classification Report:
               precision    recall  f1-score   support

           0       0.50      0.50      0.50         2
           1       0.67      0.67      0.67         3

    accuracy                           0.60         5
   macro avg       0.58      0.58      0.58         5
weighted avg       0.60      0.60      0.60         5

/usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
# Confusion Matrix
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, cmap='Greens')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```



```
Start coding or generate with AI.
```