```
# ============================================================
# RANDOM FOREST CLASSIFICATION – CUSTOMER PURCHASE PREDICTION
# ============================================================

# ------------------------------------------------------------
# Experiment Objective
# ------------------------------------------------------------
# The objective of this experiment is to apply the Random Forest
# Classification algorithm to predict whether a customer will
# purchase a product based on attributes such as age, salary,
# previous purchases, and browsing time.

# Random Forest is an ensemble learning algorithm that combines
# multiple decision trees to improve prediction accuracy and
# reduce overfitting.

# ------------------------------------------------------------
# Features Used
# ------------------------------------------------------------
# 1. Age – Age of the customer
# 2. Salary – Annual salary of the customer
# 3. Previous_Purchases – Number of previous purchases made
# 4. Browsing_Time – Time spent browsing the product

# Target Variable:
# Purchase (Yes / No)

# ------------------------------------------------------------
# STEP 1: Import Required Libraries
# ------------------------------------------------------------

import pandas as pd
import numpy as np

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
# ------------------------------------------------------------
# STEP 2: Create Sample Customer Dataset
# ------------------------------------------------------------
# A sample dataset is created to simulate customer purchase data.

data = {
    "Age": [22,25,35,45,28,32,40,50,23,37],
    "Salary": [20000,30000,60000,80000,35000,50000,70000,90000,25000,65000],
    "Previous_Purchases": [1,2,3,5,2,3,4,6,1,4],
    "Browsing_Time": [15,20,30,40,18,25,35,45,12,33],
    "Purchase": [
        "No","No","Yes","Yes","No",
        "Yes","Yes","Yes","No","Yes"
    ]
}

df = pd.DataFrame(data)

print("Customer Dataset")
print(df)
```

```
Customer Dataset
   Age  Salary  Previous_Purchases  Browsing_Time Purchase
0   22   20000                   1             15       No
1   25   30000                   2             20       No
2   35   60000                   3             30      Yes
3   45   80000                   5             40      Yes
4   28   35000                   2             18       No
5   32   50000                   3             25      Yes
6   40   70000                   4             35      Yes
7   50   90000                   6             45      Yes
8   23   25000                   1             12       No
9   37   65000                   4             33      Yes
```

```
# ------------------------------------------------------------
# STEP 3: Separate Features and Target Variable
# ------------------------------------------------------------
# X represents the input features
# y represents the target variable to be predicted
```

```
X = df[["Age","Salary","Previous_Purchases","Browsing_Time"]]
y = df["Purchase"]
```

```
# ------------------------------------------------------------
# STEP 4: Split Dataset into Training and Testing Sets
# ------------------------------------------------------------
# 80% of the data is used for training
# 20% of the data is used for testing
# Stratified sampling maintains class balance

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

```
# ------------------------------------------------------------
# STEP 5: Train Random Forest Classifier
# ------------------------------------------------------------
# Random Forest builds multiple decision trees and combines
# their predictions to produce a final output.

model = RandomForestClassifier(
    n_estimators=100,   # number of trees in the forest
    max_depth=5,        # maximum depth of each tree
    random_state=42
)

model.fit(X_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier(max_depth=5, random_state=42)
```

```
# ------------------------------------------------------------
# STEP 6: Predict Test Data
# ------------------------------------------------------------
# The trained model predicts customer purchase decisions
# for the testing dataset.

y_pred = model.predict(X_test)
```

```
# ------------------------------------------------------------
# STEP 7: Model Evaluation – Accuracy
# ------------------------------------------------------------
# Accuracy measures the percentage of correctly predicted results.

accuracy = accuracy_score(y_test, y_pred)

print("\nModel Evaluation")
print("Accuracy:", round(accuracy,3))
```

```
Model Evaluation
Accuracy: 1.0
```

```
# ------------------------------------------------------------
# STEP 8: Confusion Matrix
# ------------------------------------------------------------
# The confusion matrix shows the classification performance.

print("\nConfusion Matrix")
print(confusion_matrix(y_test, y_pred))
```

```
Confusion Matrix
[[1 0]
 [0 1]]
```

```
# ------------------------------------------------------------
# STEP 9: Classification Report
# ------------------------------------------------------------
# The classification report provides precision, recall,
# F1-score, and support values.
print("\nClassification Report")
print(classification_report(y_test, y_pred))
```

```
Classification Report
              precision    recall  f1-score   support

          No       1.00      1.00      1.00         1
         Yes       1.00      1.00      1.00         1

    accuracy                           1.00         2
   macro avg       1.00      1.00      1.00         2
weighted avg       1.00      1.00      1.00         2
```

```
# ------------------------------------------------------------
# STEP 10: Feature Importance
# ------------------------------------------------------------
# Random Forest can determine the importance of each feature
# used in prediction.

print("\nFeature Importance")

for feature, importance in zip(X.columns, model.feature_importances_):
    print(feature, ":", round(importance,3))
```

```
Feature Importance
Age : 0.25
Salary : 0.29
Previous_Purchases : 0.24
Browsing_Time : 0.22
```

```
# ------------------------------------------------------------
# STEP 11: Predict for a New Customer
# ------------------------------------------------------------
# Example new customer data is provided to test prediction.

new_customer = pd.DataFrame({
    "Age": [30],
    "Salary": [55000],
    "Previous_Purchases": [3],
    "Browsing_Time": [28]
})

prediction = model.predict(new_customer)

print("\nPrediction for New Customer:")
print("Will Purchase:", prediction[0])
```

```
Prediction for New Customer:
Will Purchase: Yes
```

```
# ------------------------------------------------------------
# STEP 12: Final Message
# ------------------------------------------------------------

print("\nRandom Forest Classification Demonstration Completed Successfully.")
```

```
Random Forest Classification Demonstration Completed Successfully.
```