**Problem Statement**:

Predict whether a product will be returned based on purchase amount, review score, and delivery time.

**Name**: ARPIT  
 **Roll No**: 202401100300065

**Introduction**

The aim is to classify whether a product will be returned or not using three features:

* Purchase amount
* Review score
* Days taken for delivery

This is a binary classification task and helps businesses reduce cost by predicting return tendencies early.

**Methodology**

* Data is read from a CSV file.
* Label Encoding converts yes/no in the returned column to 1/0.
* Data is split into 80% training and 20% testing sets.
* A **Random Forest Classifier** is used for prediction.
* Model is evaluated using:
  + **Accuracy**
  + **Precision**
  + **Recall**
  + **Confusion Matrix** (shown as a heatmap)

**Code**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import confusion\_matrix, accuracy\_score, precision\_score, recall\_score

from sklearn.preprocessing import LabelEncoder

import seaborn as sns

import matplotlib.pyplot as plt

df = pd.read\_csv('product\_return.csv')

df['returned'] = LabelEncoder().fit\_transform(df['returned'])

X = df.drop('returned', axis=1)

y = df['returned']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

clf = RandomForestClassifier(random\_state=42)

clf.fit(X\_train, y\_train)

y\_pred = clf.predict(X\_test)

print("Accuracy:", round(accuracy\_score(y\_test, y\_pred), 2))

print("Precision:", round(precision\_score(y\_test, y\_pred), 2))

print("Recall:", round(recall\_score(y\_test, y\_pred), 2))

cm = confusion\_matrix(y\_test, y\_pred)

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['No', 'Yes'], yticklabels=['No', 'Yes'])

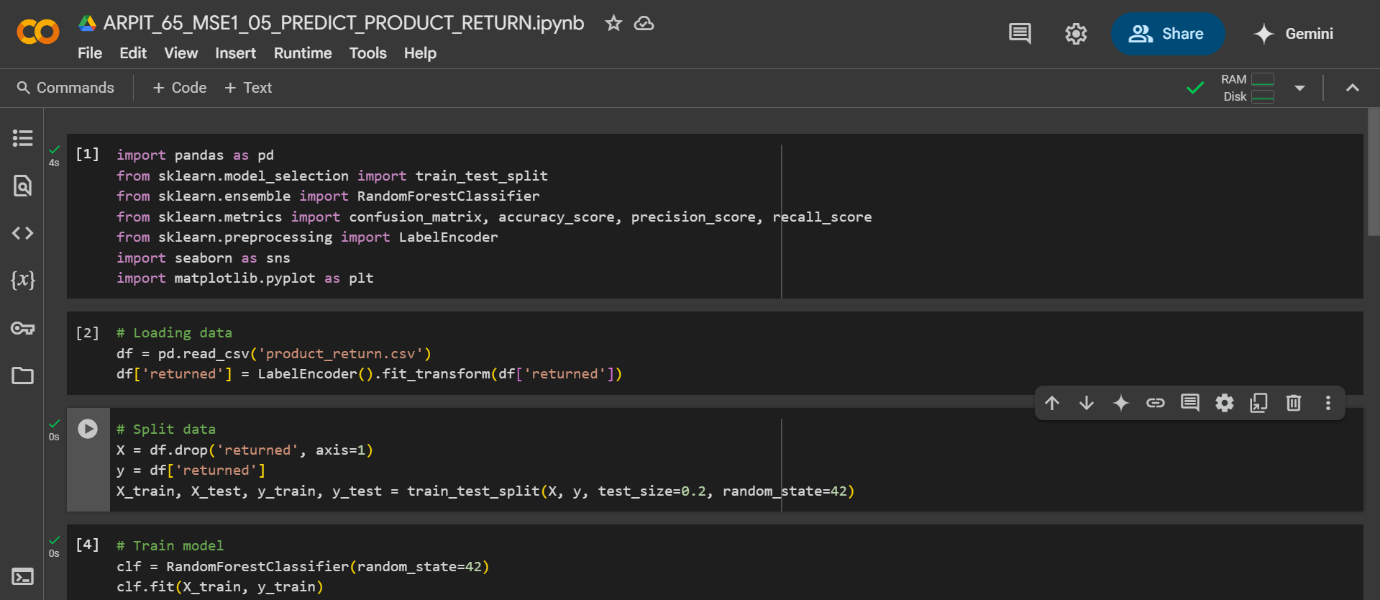
plt.xlabel('Predicted')

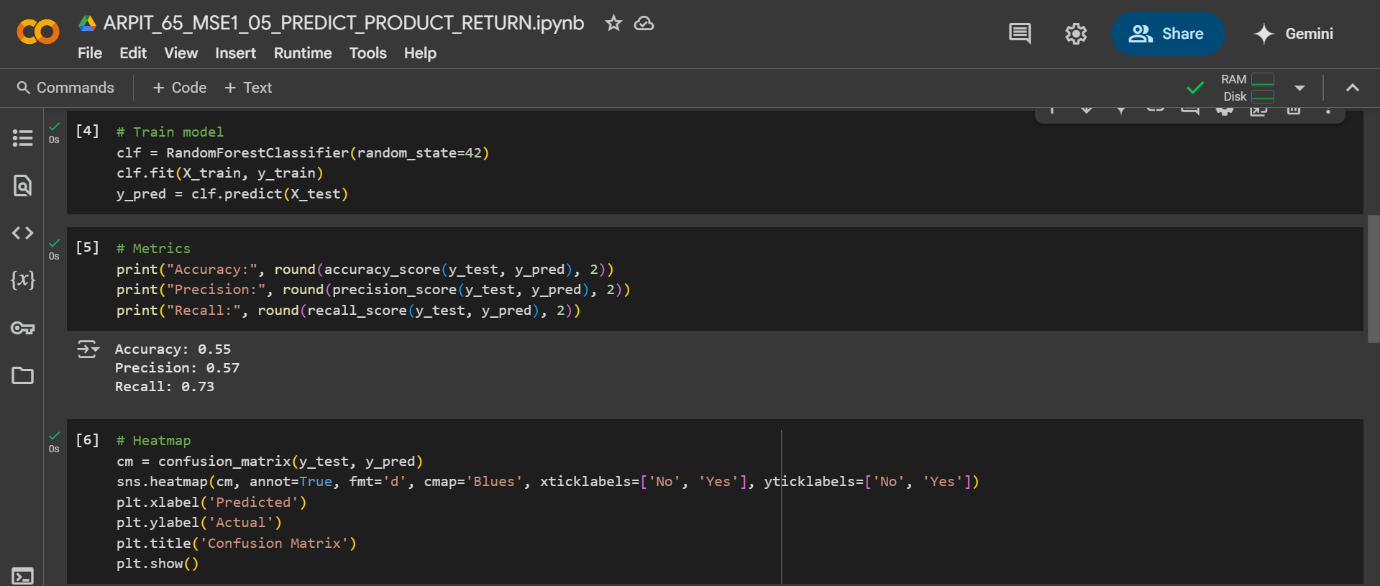
plt.ylabel('Actual')

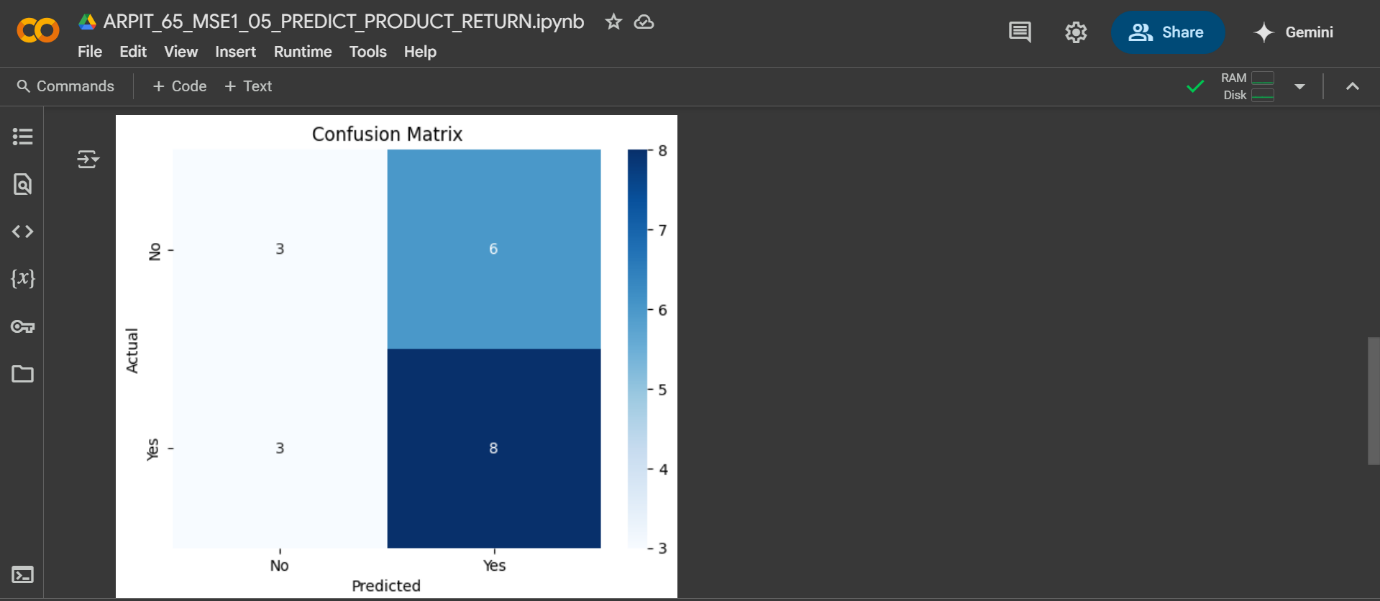
plt.title('Confusion Matrix')

plt.show()

**Output/Result**

****

****

****

**References/Credits**

* Dataset: Provided by instructor
* Libraries: pandas, sklearn, matplotlib, seaborn
* Classifier: Random Forest (from sklearn.ensemble)