

TITLE OF PROJECT REPORT  
**ONLINE LEARNING**

A PROJECT REPORT

*Submitted by*

**PRITHVI NAWADIYA**

**202401100300183**

*in partial fulfilment for the award of the degree*  
*of*

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2025-26

in

**CSE(AI)**

**INTRODUCTION**

Online learning platforms need ways to predict learner engagement and course completion to improve retention and learning outcomes. This project aims to predict course completion using behavioural features such as videos watched, assignments submitted, and forum activity.

By training a classification model, we hope to identify patterns in learner activity that indicate whether they are likely to complete a course or not.

**Methodology**

1. **Dataset**:  
The dataset contains learner behaviour logs, including:

* Videos watched
* Assignments submitted
* Forum posts
* completed (target variable: yes/no)

2. **Preprocessing**:

* Categorical target values ('yes'/'no') were converted to binary (1/0).
* Data was split into training and test sets (80/20).

3. **Model Selection**:

* A **Logistic Regression** model was used due to the small dataset and binary classification problem.
* Predictions were made on the test set.

4. **Evaluation Metrics**:

* A **confusion matrix** was plotted as a heatmap.
* Metrics computed: **Accuracy**, **Precision**, **Recall**.
* **ROC AUC** score and curve were also included for additional evaluation.

**CODE**

# Step 1: Import required libraries

import pandas as pd  # For data handling

import numpy as np  # For numerical operations

import matplotlib.pyplot as plt  # For plotting

import seaborn as sns  # For heatmap

from sklearn.model\_selection import train\_test\_split  # To split data

from sklearn.ensemble import RandomForestClassifier  # ML model

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

# Step 2: Load the dataset

df = pd.read\_csv('/content/online\_learning.csv')  # Load your dataset from file

# Step 3: Data preprocessing

# Convert 'yes'/'no' in 'completed' column to 1/0 for binary classification

df['completed'] = df['completed'].map({'yes': 1, 'no': 0})

# Step 4: Define features and target variable

# Features used for prediction

X = df[['videos\_watched', 'assignments\_submitted', 'forum\_posts']]

# Target column we want to predict

y = df['completed']

# Step 5: Split the dataset into training and testing sets

# 80% training, 20% testing

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

# Step 6: Initialize and train the model

model = RandomForestClassifier(random\_state=42)

model.fit(X\_train, y\_train)  # Fit the model to training data

# Step 7: Predict on the test set

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

# Step 8: Generate confusion matrix

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

# Step 9: Visualize the confusion matrix using a heatmap

plt.figure(figsize=(6, 4))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',

            xticklabels=['Not Completed', 'Completed'],

            yticklabels=['Not Completed', 'Completed'])

plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.title('Confusion Matrix')

plt.show()

# Step 10: Calculate and print evaluation metrics

accuracy = accuracy\_score(y\_test, y\_pred)  # Correct predictions / Total predictions

precision = precision\_score(y\_test, y\_pred)  # True Positives / (True Positives + False Positives)

recall = recall\_score(y\_test, y\_pred)  # True Positives / (True Positives + False Negatives)

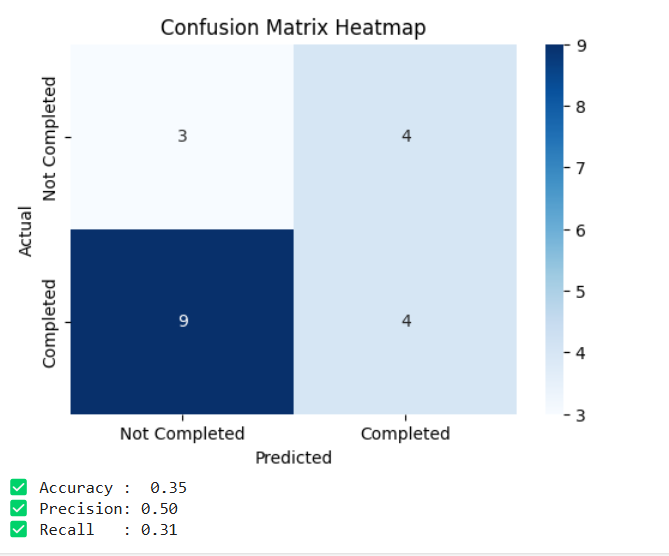
# Step 11: Display evaluation results

print(f"✅ Accuracy :  {accuracy:.2f}")

print(f"✅ Precision: {precision:.2f}")

print(f"✅ Recall   : {recall:.2f}")

**OUTPUT**



**References**

* Dataset: Provided by course instructor / academic dataset on online learning logs.
* Tools Used: Python, Google Colab, scikit-learn, pandas, seaborn, matplotlib
* Image: Confusion matrix screenshot generated from model output.