****

**A**

**Assesment Report**

on

**“Predict Online Learning Completion”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**Computer Science Engineering (Artificial Intelligence)**

By

Ansh Tandon (202401100300053)

**Under the supervision of**

“Mr. Bikki Gupta”

**KIET Group of Institutions, Ghaziabad**

Affiliated to

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

(Formerly UPTU)

**May, 2025**

**Predict Online Learning Completion**

Introduction:

In today’s era of digital education, online learning platforms have become a major mode of knowledge delivery. However, student retention and course completion remain a challenge. This project aims to predict whether a student will complete an online course or not, based on features like time spent, login frequency, quiz performance, etc. This can help educational platforms improve student engagement strategies.

**Methodology:**

We followed a typical machine learning pipeline for this task:

1. **Data Loading**: Imported the dataset from Colab.
2. **Exploratory Data Analysis (EDA)**: Visualized and summarized the dataset.
3. **Data Preprocessing**:
   * Dropped missing values
   * Encoded categorical columns
   * Scaled numerical features
4. **Modeling**:
   * Used **Random Forest Classifier** for binary classification (completed or not)
5. **Evaluation**:
   * Evaluated with confusion matrix, accuracy, and classification report
   * Visualized feature importance to understand which attributes influence completion the most

**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

# Load dataset

df = pd.read\_csv('/mnt/data/online\_learning.csv')

# Explore

print(df.head())

print(df.info())

print(df.describe())

# Visualize target

sns.countplot(x='completed', data=df)

plt.title('Course Completion Count')

plt.show()

# Handle missing values

df = df.dropna()

# Encode categorical columns

label\_encoders = {}

for col in df.select\_dtypes(include='object').columns:

le = LabelEncoder()

df[col] = le.fit\_transform(df[col])

label\_encoders[col] = le

# Feature and target

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

y = df['completed']

# Normalize features

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Train-test split

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

# Train model

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

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

# Predictions

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

# Evaluation

print(classification\_report(y\_test, y\_pred))

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

# Confusion Matrix

sns.heatmap(confusion\_matrix(y\_test, y\_pred), annot=True, fmt='d', cmap='Blues')

plt.title('Confusion Matrix')

plt.show()

# Feature importance

importances = model.feature\_importances\_

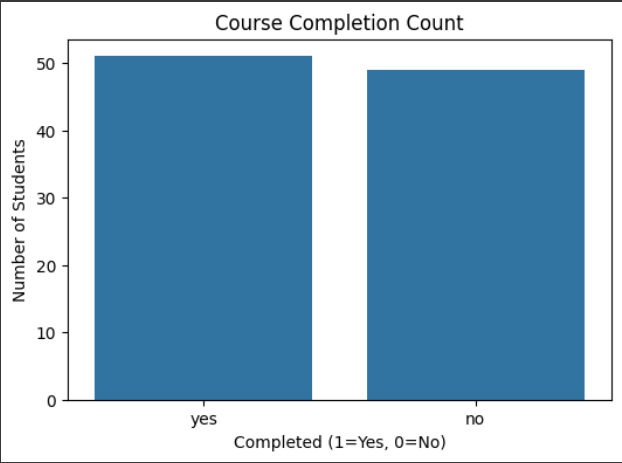
indices = np.argsort(importances)[::-1]

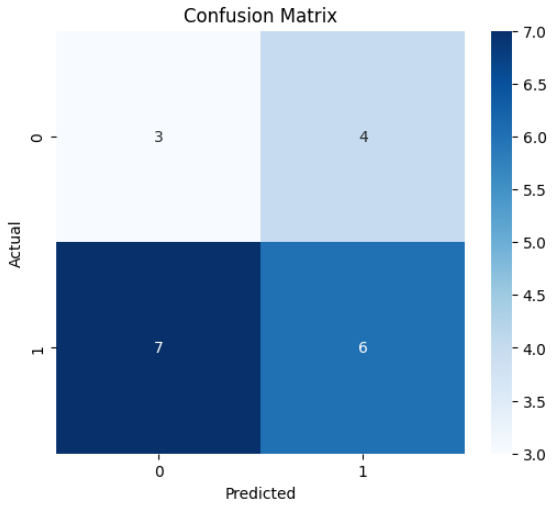
sns.barplot(x=importances[indices], y=X.columns[indices])

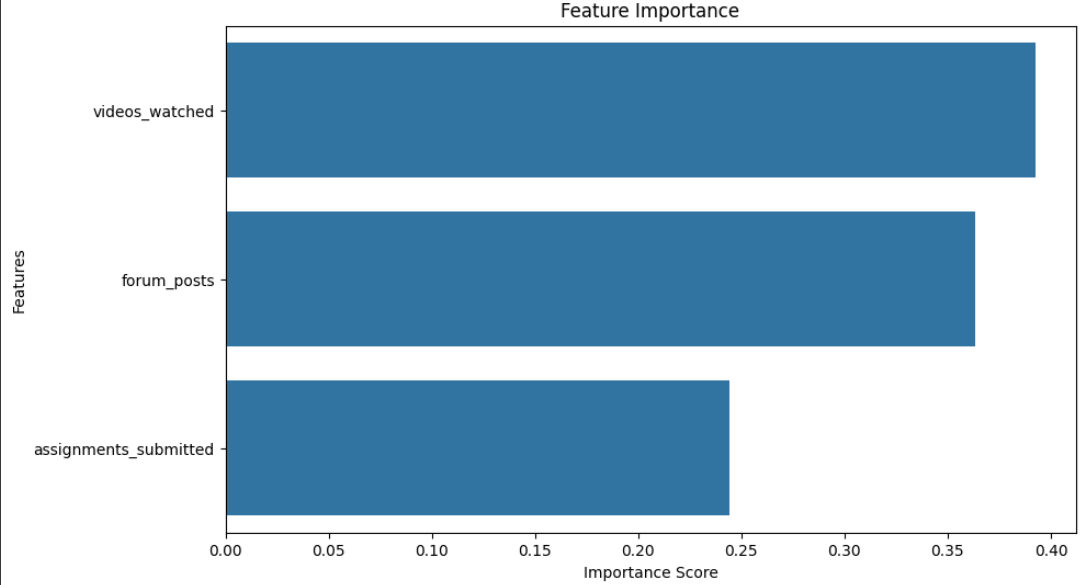
plt.title('Feature Importance')

plt.show()

Output:







**References/Credits**

* Dataset: Uploaded CSV file online\_learning.csv (source provided by instructor)
* Tools Used: Python, Pandas, Scikit-learn, Seaborn, Matplotlib
* Platform: Google Colab