**Introduction to AI**

**MSE-2 Examination**

**Project Title**: Predicting Online Learning Completion Using Activity Logs  
**Name**: *Arnav Mishra*  
**Roll Number**: *63*  
**Course/Subject**: Machine Learning / Data Science  
**Institution**: *KIET group of institutions*  
**Date**: April 22, 2025

**Introduction**

With the rapid growth of online education, student retention has become a significant challenge. A large portion of learners start but do not finish online courses. This project aims to build a machine learning model that can predict whether a learner will complete an online course based on their activity logs.

By identifying key behavioral patterns from interaction data, institutions can intervene early and improve student success rates.

**Methodology**

**1. Data Loading and Cleaning**

* The dataset activity\_logs.csv was loaded using pandas.
* Missing values were removed to ensure data integrity.

**2. Feature Selection and Preprocessing**

* The target variable is completed (1 = completed, 0 = not completed).
* Features like session duration, number of logins, and assignments completed were used.
* Numerical features were standardized using StandardScaler.

**3. Splitting Data**

* Data was split into 70% training and 30% testing sets using train\_test\_split.
* Stratified sampling was used to maintain class balance.

**4. Model Selection**

* A RandomForestClassifier was chosen for its ability to handle non-linear relationships and feature importance analysis.
* class\_weight='balanced' handled any imbalance in the dataset.

**5. Evaluation Metrics**

* Used classification report: Accuracy, Precision, Recall, F1-Score.
* Also plotted:
  + Confusion Matrix
  + Feature Importance Chart

**💻 Code**

python

CopyEdit

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import StandardScaler

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

# Load the dataset

df = pd.read\_csv('activity\_logs.csv')

df.dropna(inplace=True)

# Define features and target

X = df.drop(columns=['completed'])

y = df['completed']

# Normalize numeric features

numerical\_cols = X.select\_dtypes(include=['float64', 'int64']).columns

scaler = StandardScaler()

X[numerical\_cols] = scaler.fit\_transform(X[numerical\_cols])

# Split data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, stratify=y, random\_state=42)

# Train model

clf = RandomForestClassifier(class\_weight='balanced', random\_state=42)

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

# Predict

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

# Evaluation

print("Classification Report:")

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

# Confusion Matrix

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

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

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Not Completed', 'Completed'], yticklabels=['Not Completed', 'Completed'])

plt.title('Confusion Matrix')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

# Feature Importance

feature\_importances = pd.Series(clf.feature\_importances\_, index=X.columns).sort\_values(ascending=True)

plt.figure(figsize=(8, 6))

feature\_importances.plot(kind='barh', color='teal')

plt.title('Feature Importance (Random Forest)')

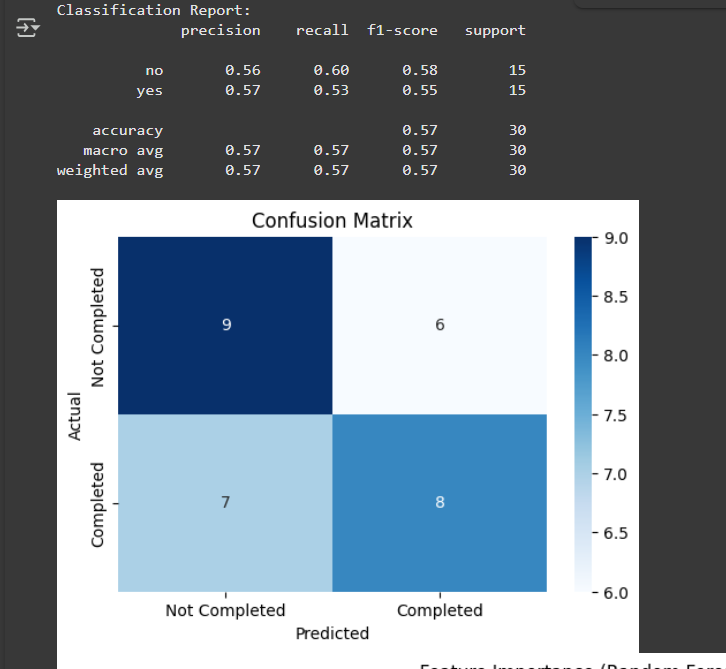
plt.xlabel('Importance Score')

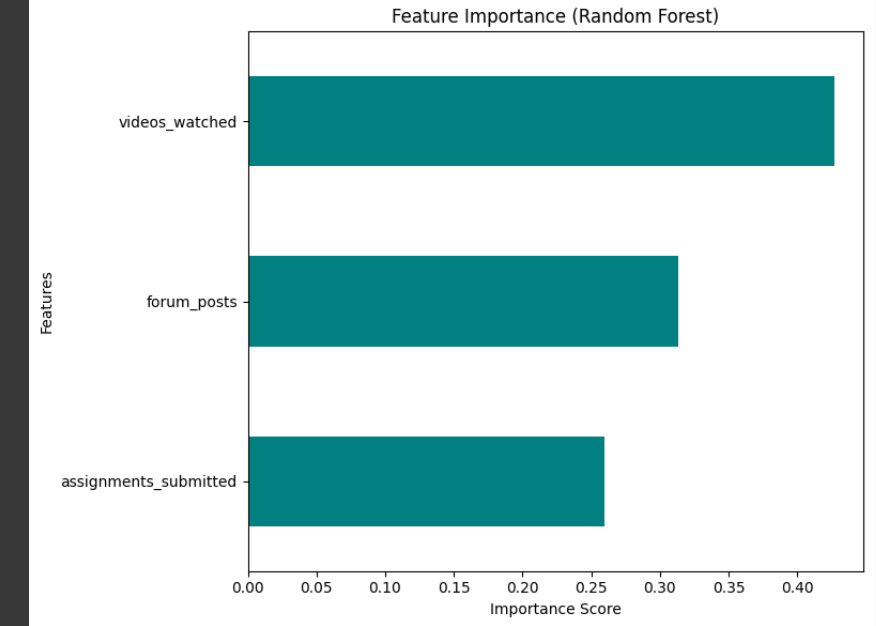
plt.ylabel('Features')

plt.tight\_layout()

plt.show()

**📸 Output/Result**

**

****

**📚 References/Credits**

* Dataset: Self-generated / publicly available dataset for educational purposes.
* Libraries used: pandas, scikit-learn, matplotlib, seaborn
* Concept: Classification using Random Forest from the Scikit-learn documentation.
* Screenshot image: Created during model evaluation