

# **TITLE OF PROJECT REPORT**

Predict Online Learning Completion

PROJECT REPORT

Submitted by:

Radhika

CSEAI-C

202401100300190

KIET GROUP OF INSTITUTIONS

# Introduction:

Online learning has grown rapidly, offering flexible access to education through platforms like MOOCs. However, many learners struggle to complete courses due to low engagement or other challenges. Predicting course completion using activity logs—such as video views, quiz attempts, and login frequency—can help identify at-risk students early. This project aims to build a model that classifies whether a learner will complete a course, enabling better support and improved learning outcomes.

# Methodology:

To build a reliable model for predicting course completion, we followed a structured machine learning pipeline. The key steps involved are:

## 1. Data Collection and Preprocessing:

The dataset was sourced from an online learning platform, consisting of user activity logs. Features included login frequency, time spent on the platform, video interactions, quiz attempts, forum activity, and final course status (completed or not).

## 2. Feature Selection:

Important features were selected based on correlation analysis and domain knowledge. New features were also engineered (e.g., average session time, activity frequency) to improve model performance.

## 3. Model Evaluation:

Models were evaluated using performance metrics such as Accuracy, Precision, Recall, F1-score, and ROC-AUC. Cross-validation was used to ensure generalization and prevent overfitting.

## CODE TYPED:

```
# Import necessary libraries

import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report


# Step 1: Load the dataset from the given file path
file_path = '/content/online_learning.csv' # Ensure this file exists in your environment
df = pd.read_csv(file_path) # Using default comma separator
df.columns = df.columns.str.strip() # Clean up column names


# Step 2: Preview the data
print("First 5 rows of the dataset:")
print(df.head())
print("Columns:", df.columns.tolist()) # Optional: check column names


# Step 3: Convert 'completed' column from 'yes'/'no' to 1/0
label_encoder = LabelEncoder()
df['completed'] = label_encoder.fit_transform(df['completed']) # yes-
> 1, no-> 0


# Step 4: Split the dataset into features (X) and target (y)
X = df[['videos_watched', 'assignments_submitted', 'forum_posts']] #
Feature columns
y = df['completed'] # Target column
```

# Step 5: Split the dataset into training and testing sets (80% train, 20% test)

```
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 a Logistic Regression model

```
model = LogisticRegression()
```

```
model.fit(X_train, y_train)
```

# Step 7: Make predictions on the test data

```
y_pred = model.predict(X_test)
```

# Step 8: Evaluate the model performance

```
print("\nModel Evaluation:")
```

```
print("Accuracy:", accuracy_score(y_test, y_pred))
```

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

```
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

# SCREENSHOTS OF OUTPUT:

First 5 rows of the dataset:

	videos_watched	assignments_submitted	forum_posts	completed
cell output actions	11	6	5	yes
1	43	1	11	no
2	37	1	8	no
3	18	4	14	yes
4	6	4	15	yes

Columns: ['videos\_watched', 'assignments\_submitted', 'forum\_posts', 'completed']

Model Evaluation:

Accuracy: 0.4

Confusion Matrix:

[[4 3]  
[9 4]]

Classification Report:

	precision	recall	f1-score	support
0	0.31	0.57	0.40	7
1	0.57	0.31	0.40	13

accuracy

macro avg

weighted avg

0.40

0.44

0.48

20

20

20



# REFERENCES:

videos\_watched

assignments\_submitted  
forum\_posts      completed

11	6	5	yes
43	1	11	no
37	1	8	no
18	4	14	yes
6	4	15	yes
34	6	11	no
47	0	13	no
17	7	12	no
15	2	1	yes
38	3	2	yes
3	8	9	yes
24	3	8	no
31	9	19	no
39	4	7	yes
12	6	9	yes
44	3	18	no
16	0	2	yes
48	8	4	no
36	3	2	no
5	8	15	yes
18	5	16	no
12	0	14	yes
25	2	2	no
13	1	11	no
30	8	8	yes
3	0	11	no

38	8	14	yes
7	8	1	yes
24	7	17	no
12	7	19	yes
31	0	11	yes
37	7	3	yes
37	9	15	yes
27	2	15	yes
29	6	0	yes
2	0	1	no
27	0	2	yes
2	9	19	no
44	4	16	no
33	4	19	yes
15	7	0	no
21	6	19	no
31	7	5	yes
37	5	14	yes
7	5	13	yes
7	1	19	no
5	5	14	no
13	4	13	yes
31	2	4	yes
34	5	5	no
9	7	16	no
2	8	10	no
11	0	15	no
48	2	11	yes
30	2	14	no



28	7	3	no
4	0	17	no
24	4	2	no
40	2	7	yes
30	1	2	yes
20	7	7	no
19	2	17	no
38	6	9	yes
21	0	19	yes
3	5	16	no
12	2	7	no
44	7	10	yes
41	5	4	yes
49	1	12	yes
48	5	4	yes
19	6	12	yes
12	4	5	yes
16	2	4	no
35	3	8	yes
32	3	15	yes
0	5	1	yes
14	9	0	yes
12	0	7	no
16	3	18	no
23	9	10	yes
7	3	11	no
49	9	11	no
26	0	6	no
44	8	12	no

22	3	7	no
45	6	8	no
7	2	6	yes
7	9	14	yes
25	2	1	yes
46	7	19	yes
12	7	11	yes
38	0	0	no
36	9	7	yes
48	1	4	yes
27	7	14	no
24	0	0	no
15	7	0	yes
1	7	9	no
17	1	9	no
16	3	14	no

And using google colab