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
0              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                           0.40        20
   macro avg       0.44      0.44      0.40        20
weighted avg       0.48      0.40      0.40        20
```

cell output actions

REFERENCES:

1. **Kloft, M., Stiehler, F., Zheng, Z., & Pinkwart, N. (2014).**
   Predicting MOOC dropout over weeks using machine learning methods.
   In *Proceedings of the EMNLP Workshop on Modeling Large Scale Social Interaction in Massively Open Online Courses*.
   ACL Anthology Link

2. **Zhou, M., & Chiu, M. M. (2016).**
   Personalized learning in MOOCs: A fuzzy item response theory approach.
   In *The Handbook of Learning Analytics*, edited by Lang et al.
   Society for Learning Analytics Research (SoLAR).
   https://solaresearch.org/hla-2022/ (Latest edition online)