Transformando la educación a través de la inteligencia artificial: un caso práctico

Andres Felipe Lopez Lozano

Alfonso Ibañez

Inteligencia Artificial

2025

**Solución al proyecto de aplicación**

1. Descripción: Introducción al problema

El sector educativo está atravesando una transformación significativa impulsada por los rápidos avances en inteligencia artificial (IA). Las metodologías de aprendizaje tradicionales a menudo no abordan las diversas necesidades de los estudiantes, lo que da como resultado una falta de experiencias de aprendizaje personalizadas. La IA presenta una oportunidad para mejorar los resultados de aprendizaje al brindar tutoría personalizada, mejorar la accesibilidad para estudiantes con discapacidades y optimizar los procesos administrativos.

Este proyecto propone un sistema de tutoría inteligente (ITS) impulsado por IA que personaliza el aprendizaje en función del progreso de los estudiantes, identifica las brechas de conocimiento y ofrece retroalimentación en tiempo real. Al integrar soluciones impulsadas por IA, las instituciones educativas pueden mejorar significativamente la participación, la retención y el rendimiento académico general.

2. Datos necesarios: fuentes de información para el caso de uso

Para implementar de manera eficaz un sistema de tutoría basado en IA, se necesitarán las siguientes fuentes de datos:

**Datos de rendimiento de los estudiantes:** registros académicos históricos, puntajes de exámenes y tasas de finalización de cursos.

Creamos una tabla para representar nuestros casos de uso reales y para representar nuestros modelos algorítmicos.

```sql

CREATE TABLE Rendimiento\_estudiante (

student\_id INT PRIMARY KEY,

name VARCHAR(50),

test\_scores FLOAT,

course\_completion\_rate FLOAT

);

INSERT INTO Rendimiento\_estudiante (student\_id, name, test\_scores, course\_completion\_rate)

VALUES

(101, 'Alice Smith', 85.5, 0.92),

(102, 'Bob Johnson', 78.0, 0.85);

```

|  |  |  |  |
| --- | --- | --- | --- |
| student\_id | name | test\_scores | course\_completion\_rate |
| 101 | Alice Smith | 85.5 | 0.92 |
| 102 | Bob Johnson | 78.0 | 0.85 |

**Datos de comportamiento de aprendizaje:** registros de interacción de los sistemas de gestión del aprendizaje (LMS), incluidos los intentos de realizar exámenes, la duración de visualización de videos y los niveles de participación.

CREATE TABLE Comportamiento\_aprendizaje (

student\_id INT,

quiz\_attempts INT,

video\_watch\_duration INT, -- in minutes

engagement\_level FLOAT

);

INSERT INTO Comportamiento\_aprendizaje (student\_id, quiz\_attempts, video\_watch\_duration, engagement\_level)

VALUES

(101, 5, 120, 0.89),

(102, 3, 90, 0.76);

|  |  |  |  |
| --- | --- | --- | --- |
| student\_id | quiz\_attempts | video\_watch\_duration | engagement\_level |
| 101 | 5 | 120 | 0.89 |
| 102 | 3 | 90 | 0.76 |

**Información demográfica:** edad, ubicación, preferencias de aprendizaje y dominio del idioma.

```sql

CREATE TABLE Informacion\_demografica (

student\_id INT PRIMARY KEY,

age INT,

location VARCHAR(50),

learning\_preference VARCHAR(50),

language VARCHAR(20)

);

INSERT INTO Informacion\_demografica (student\_id, age, location, learning\_preference, language)

VALUES

(101, 21, 'New York', 'Visual', 'English'),

(102, 25, 'Los Angeles', 'Auditory', 'Spanish');

```

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| student\_id | age | location | learning\_preference | language |
| 101 | 21 | New York | Visual | English |
| 102 | 25 | Los Angeles | Auditory | Spanish |

**Material del curso:** libros de texto digitales, videos de conferencias, tareas y evaluaciones anteriores.

CREATE TABLE Informacion\_curso (

course\_id INT PRIMARY KEY,

course\_name VARCHAR(100),

material\_type VARCHAR(50),

content\_link VARCHAR(255)

);

INSERT INTO Informacion\_curso (course\_id, course\_name, material\_type, content\_link) VALUES

(301, 'AI Fundamentals', 'Video', 'https://edu-platform.com/ai\_video'),

(302, 'Data Science', 'Textbook', 'https://edu-platform.com/ds\_textbook');

|  |  |  |  |
| --- | --- | --- | --- |
| course\_id | course\_name | material\_type | content\_link |
| 301 | AI Fundamentals | Video | https://edu-platform.com/ai\_video |
| 302 | Data Science | Textbook | https://edu-platform.com/ds\_textbook |

**Datos de retroalimentación:** evaluaciones de estudiantes y docentes, análisis de sentimientos de foros de discusión y puntajes de revisión.

CREATE TABLE Retroalimentacion (

student\_id INT,

instructor\_id INT,

feedback\_text TEXT,

rating FLOAT

);

INSERT INTO Feedback (student\_id, instructor\_id, feedback\_text, rating)

VALUES

(101, 501, 'The course was engaging and structured.', 4.8),

(102, 502, 'Includes real-world case studies.', 3.9);

|  |  |  |  |
| --- | --- | --- | --- |
| student\_id | instructor\_id | feedback\_text | rating |
| 101 | 501 | El curso fue interesante y estructurado. | 4.8 |
| 102 | 502 | Incluye estudios de casos del mundo real. | 3.9 |

3. Enfoque analítico: tipo de aprendizaje y algoritmo utilizado

El sistema de tutoría de IA utilizará un enfoque de aprendizaje supervisado para recomendaciones personalizadas y un enfoque de aprendizaje no supervisado para agrupar a los estudiantes en función de sus estilos de aprendizaje. Se implementarán los siguientes algoritmos:

1. Aprendizaje de refuerzo (RL): para ajustar dinámicamente los planes de lecciones en función de la participación y las respuestas de los estudiantes.
2. Modelo matemático

El aprendizaje por refuerzo (RL) se basa en los procesos de decisión de Markov (MDP), en los que un agente interactúa con un entorno para maximizar las recompensas acumuladas a lo largo del tiempo.

Un MDP consta de:

Estados (S): el nivel de participación actual del alumno y los intentos de resolver el cuestionario.

Acciones (A): las posibles respuestas del tutor de IA (avanzar al siguiente nivel, reforzar conceptos o proporcionar ejercicios adicionales).

Recompensas (R): el beneficio asociado con una acción (p. ej., mejores puntajes en el cuestionario, mayor participación).

Política (π): una correlación de los estados con las acciones que maximiza las recompensas esperadas.

La ecuación de Bellman es fundamental para el aprendizaje por refuerzo:

V(s) = max [R(s, a) + γ ∑s' P(s' | s, a) V(s')]

donde:

𝑉 ( 𝑠 ) V(s) es el valor del estado 𝑠 s,

𝑅 ( 𝑠 , 𝑎 ) R(s,a) es la recompensa inmediata de tomar la acción 𝑎 a,

𝑃 ( 𝑠 ′ ∣ 𝑠 , 𝑎 ) P(s ′ ∣s,a) es la probabilidad de alcanzar el estado 𝑠 ′ s ′ desde 𝑠 s usando la acción 𝑎 a,

𝛾 es el factor de descuento para recompensas futuras.

B. Aristas identificadas y ajustes

Arista 1: Falta de consideración de recompensas a largo plazo

Problema: La implementación actual de RL evalúa solo el nivel de participación inmediata y los intentos de prueba, sin considerar la retención de aprendizaje a largo plazo.

Solución: Introducir un sistema de recompensa acumulativa con descuento:

* Realizar un seguimiento de la participación en varias sesiones.
* Modificar la función de recompensa para considerar la retención de conocimiento a largo plazo:

𝑅𝑡 + 1 = 𝛼 ⋅ Puntuación inmediata + (1 − 𝛼) ⋅ Puntuación de retención

Donde α es un factor de ponderación ajustable.

Arista 2: Necesidad de una tasa de aprendizaje adaptativa

Problema: Los umbrales fijos para los niveles de participación pueden no generalizarse bien entre estudiantes diversos.

Solución: Implementar una tasa de aprendizaje dinámica utilizando una política ε-greedy:

* Con probabilidad ε, explorar una nueva estrategia de aprendizaje.
* Con probabilidad 1-ε, explotar la estrategia de aprendizaje más conocida.

1. Código de Implementación:

import pandas as pd

import numpy as np

**# Cargar datos de aprendizaje**

learning\_behavior\_df = pd.read\_csv("https://drive.google.com/uc?id=YOUR\_CSV\_FILE\_ID")

**# RLM con recompensas con descuento**

def reward\_function(engagement\_level, quiz\_attempts, retention\_score, alpha=0.7):

**""" Calcular una recompensa combinando factores de retención inmediatos y a largo plazo."""**

return alpha \* (engagement\_level \* quiz\_attempts) + (1 - alpha) \* retention\_score

**# Ajuste de la ruta de aprendizaje adaptativa**

def adjust\_learning\_path(engagement\_level, quiz\_attempts, retention\_score, epsilon=0.1):

**""" Ajuste dinámicamente la ruta de aprendizaje según los principios del aprendizaje relizado."""**

reward = reward\_function(engagement\_level, quiz\_attempts, retention\_score)

if np.random.rand() < epsilon: **# Estrategia de exploración**

return np.random.choice(["Avanza al siguiente nivel", "Reforzar conceptos", "Ejercicios extra"])

if reward > 7:

return **"Avanza al siguiente nivel"**

elif reward > 4:

return **"Reforzar conceptos previos"**

else:

return **"Proporcionar ejercicios adicionales"**

**# Aplicar el modelo de aprendizaje por refuerzo**

learning\_behavior\_df["learning\_path"] = learning\_behavior\_df.apply(

lambda row: adjust\_learning\_path(row["engagement\_level"], row["quiz\_attempts"], row.get("retention\_score", 0)), axis=1

)

**# Mostrar resultados**

import ace\_tools as tools

tools.display\_dataframe\_to\_user(name=" **Rutas de aprendizaje mejoradas** ", dataframe=learning\_behavior\_df)

El modelo de aprendizaje automático ajustado mejora la personalización y la adaptabilidad al:

* Incorporar recompensas de retención a largo plazo en lugar de solo una participación inmediata.
* Introducir una tasa de aprendizaje adaptativa (estrategia ε-greedy) para equilibrar la exploración frente a la explotación.

1. Procesamiento del lenguaje natural (NLP): para analizar las consultas de los estudiantes y generar respuestas relevantes.
2. Modelo matemático

El procesamiento del lenguaje natural (PLN) para la tutoría de IA implica aprovechar los modelos de lenguaje basados ​​en Transformers, como BERT (representaciones de codificador bidireccional de Transformers), para comprender las consultas de los estudiantes y generar respuestas adecuadas.

Los Transformers o transformadores utilizan mecanismos de autoatención para procesar el texto de manera eficiente. La formulación matemática clave detrás de Transformers incluye:

1. Mecanismo de autoatención:

Attention(Q,K,V)=softmax(dk​​QKT​)V

donde:

* Q (Consulta), K (Clave), y V (Valor) son representaciones de palabras en el texto de entrada.
* dk es la dimensión de los vectores clave.
* Softmax asegura que las puntuaciones de atención sumen 1, indicando la importancia relativa.

1. Incrustaciones de tokens: las palabras se convierten en vectores numéricos mediante la tokenización de WordPiece, lo que permite que el modelo gestione palabras fuera del vocabulario de manera efectiva.
2. Modelo de lenguaje enmascarado (MLM): Durante el entrenamiento, algunas palabras se enmascaran y se predicen utilizando las palabras circundantes.

P(wt​∣wt−1​,wt+1​)=∑eyj​/eyt​​

donde yt representa la distribución de probabilidad sobre el vocabulario para la palabra faltante.

1. Aristas identificadas y ajustes

Arista 1: Manejo de la ambigüedad de contexto en las consultas de los estudiantes

Problema: Los modelos de Transformer a veces pueden malinterpretar las consultas ambiguas de los estudiantes.

Solución: Implementar la expansión de contexto, lo que permite que el modelo recupere contenido relacionado adicional de una base de conocimiento.

Arista 2: Precisión de la respuesta y mitigación de sesgos

Problema: Los modelos entrenados previamente pueden generar respuestas sesgadas o factualmente incorrectas.

Solución: Ajustar el modelo en conjuntos de datos específicos de la educación e implementar el filtrado de sesgos mediante la calificación de las respuestas por neutralidad.

1. Código de implementación

from transformers import pipeline

**# Cargue el modelo de PNL entrenado previamente para responder preguntas**

nlp\_model = pipeline("question-answering")

**# Base de conocimientos para proporcionar un contexto ampliado**

knowledge\_base = {

"Redes neuronales": "Las redes neuronales son modelos computacionales inspirados en el cerebro humano, que se utilizan para reconocer patrones en los datos".

"Consisten en múltiples capas de neuronas que procesan la información de manera secuencial".

"Descenso de gradiente": "El descenso de gradiente es un algoritmo de optimización utilizado en el aprendizaje automático para minimizar una función de pérdida".

"Actualiza iterativamente los parámetros del modelo para alcanzar la solución óptima".

}

**# Función para procesar consultas de estudiantes**

def get\_ai\_tutor\_response(question):

"""Recupere el contexto relevante y genere una respuesta de IA."""

relevant\_context = None

**# Busque temas relacionados en la base de conocimientos**

for key, value in knowledge\_base.items():

if key.lower() in question.lower():

relevant\_context = value

break

**# Respuesta predeterminada si no se encuentra un contexto relevante**

if not relevant\_context:

return "Necesito más detalles para poder dar una respuesta precisa. ¿Puedes aclararme tu pregunta?"

**# Respuesta generada por IA utilizando el modelo de PNL**

answer = nlp\_model(question=question, context=relevant\_context)

**# Se verifica el nivel de confianza y ajuste la respuesta**

if answer["score"] < 0.5:

return "No estoy completamente seguro. Quizás quieras revisar recursos adicionales".

return answer["answer"]

**# Ejemplos de preguntas para estudiantes**

questions = [

"¿Para qué se utilizan las redes neuronales?",

"¿Puedes explicar el descenso de gradiente?",

"Háblame del sesgo de la IA".

]

**# Generar respuestas**

responses = {q: get\_ai\_tutor\_response(q) for q in questions}

**# Mostrar resultados**

import ace\_tools as tools

tools.display\_dataframe\_to\_user(name="AI Tutor Responses", dataframe=pd.DataFrame(responses.items(), columns=["Question", "Response"]))

1. Árboles de decisión y bosques aleatorios: para predecir el rendimiento de los estudiantes y las tasas de deserción.
2. Modelo matemático

Los árboles de decisión y los bosques aleatorios son modelos de aprendizaje supervisado que se utilizan para tareas de clasificación y regresión. En el contexto de la predicción de la deserción escolar, analizan los datos de rendimiento académico y los niveles de participación para determinar si es probable que un estudiante abandone la escuela.

Criterio de división de árboles de decisión Los árboles de decisión utilizan la impureza de Gini o la entropía para determinar la mejor división en cada nodo.

**Impureza de Gini:**

Gini = 1 − ∑ᵢ₌₁ⁿ pᵢ²

Donde pᵢ es la probabilidad de que ocurra una clase.

**Entropía (Ganancia de Información):**

H(S) = -Σ (pi log2 pi)

La división con la mayor ganancia de información es elegida:

IG = H(Parent) − ∑(|Child| / |Parent|) \* H(Child)

1. Aprendizaje por conjuntos de Random Forest Un Random Forest es un conjunto de múltiples árboles de decisión que reduce la varianza y mejora la solidez. La predicción se realiza mediante una votación mayoritaria:

y^​=argkmax​i=1∑n​I(hi​(x)=k)

Donde Hi(x) es la predicción del árbol i, y se elige la clase mayoritaria.

1. Aristas identificadas y ajustes
2. Redes neuronales (LSTM): para mejorar el aprendizaje adaptativo a través del reconocimiento de patrones secuenciales.

**Aplicación Práctica del Conocimiento**

En este espacio a partir de la solución generada, con sus propias palabras explique la utilidad o cómo podría aplicar la situación planteada del proyecto de aplicación en programa académico y/o su vida profesional, de acuerdo conocimientos adquiridos a lo largo de la asignatura.

**Referencias**

En este espacio debe incluir las referencias citadas en el texto. Se debe hacer uso de la Norma APA, el manual o guía está disponible en <http://uniasturias.edu.co/SUMMA/pdf/norma_apa_7_edicion.pdf>.

"**El trabajo no puede superar las dos caras de folio, sin contar, a estos efectos, la portada y las referencias bibliográficas"**