Red Neuronal para Predicción de Abandono Estudiantil

## Objetivo del Proyecto

Diseñar, entrenar y probar una red neuronal artificial (RNA) que prediga si un estudiante universitario continuará, abandonará o se graduará, con base en datos reales.

## Tecnologías Utilizadas

- Python 3.11  
- TensorFlow / Keras  
- Pandas, NumPy, Scikit-learn  
- Jupyter o Visual Studio Code (opcional)

## Estructura del Proyecto

RED-NEURONAL/  
├── data/ # Dataset base  
│ └── data.csv  
├── models/ # Modelo entrenado  
│ └── dropout\_model.h5  
├── src/ # Código fuente organizado  
│ ├── model.py  
│ ├── preprocessing.py  
│ └── main.py  
├── predict.py # Script de predicción independiente  
├── requirements.txt # Dependencias del proyecto  
├── venv/ # Entorno virtual (generado localmente)  
└── pyvenv.cfg

**DATASET:**

**URL:** <https://archive.ics.uci.edu/dataset/697/predict%2Bstudents%2Bdropout%2Band%2Bacademic%2Bsuccess?utm_source=chatgpt.com>

## Fuente del Conjunto de Datos

## El conjunto de datos utilizado fue obtenido de la plataforma pública Kaggle y forma parte de un proyecto que busca aplicar modelos de aprendizaje automático para predecir el abandono estudiantil en la educación superior.

## Este dataset fue recolectado y publicado por Alexandre Ribeiro, y está respaldado por el programa SATDAP - Capacitação da Administração Pública, subvención POCI-05-5762-FSE-000191 (Portugal).

El dataset contiene un total de **42 columnas (atributos)**, que incluyen aspectos demográficos, socioeconómicos, académicos y de desempeño con mas de 4424 estudiantes.

|  |  |  |  |
| --- | --- | --- | --- |
| Marital status | Mother's qualification | Curricular units 1st sem (evaluations) | Curricular units 2nd sem (grade) |
| Application mode | Father's qualification | Curricular units 1st sem (approved) | Curricular units 2nd sem (without evaluations) |
| Application order | Mother's occupation | Curricular units 1st sem (grade) | Unemployment rate |
| Course | Father's occupation | Curricular units 1st sem (without evaluations) | Inflation rate |
| Daytime/evening attendance | Admission grade | Curricular units 2nd sem (credited) | GDP |
| Previous qualification | Displaced | Curricular units 2nd sem (enrolled) | Target (abandono, continúa, graduado) |
| Previous qualification (grade) | Educational special needs | Curricular units 2nd sem (evaluations) |  |
| Nacionality | Debtor | Curricular units 2nd sem (approved) |  |
| Gender | Tuition fees up to date |  |  |
| Scholarship holder | Age at enrollment |  |  |
| International |  |  |  |

## Pasos para Levantar el Proyecto

## 1. Instalar Python 3.11

Verifica tu versión:  
python --version

Si tienes Python 3.11.x ya estás listo.

## 2. Crear un entorno virtual

Desde tu carpeta del proyecto red-neuronal, ejecuta:  
python -m venv venv

## 3. Activar el entorno virtual

Windows CMD:  
venv\Scripts\activate

PowerShell:  
.\venv\Scripts\Activate.ps1

## 4. Instalar dependencias necesarias

pip install --upgrade pip  
pip install tensorflow pandas numpy scikit-learn matplotlib

## 5. Colocar el modelo entrenado

Asegúrate de tener el archivo dropout\_model.h5 dentro de la carpeta models/.

## 6. Probar el modelo con una entrada real

Usa el script predict.py con el siguiente contenido:

## Código: predict.py

|  |
| --- |
| import numpy as np  import pandas as pd  from sklearn.preprocessing import LabelEncoder, StandardScaler  from tensorflow.keras.models import load\_model  # Cargar modelo  model = load\_model('models/dropout\_model.h5')  # Cargar datos originales  df = pd.read\_csv('data/data.csv', sep=';')  # Igual que en el entrenamiento  drop\_cols = ['Course', 'Nacionality', "Daytime/evening attendance\t"]  df.drop(columns=drop\_cols, inplace=True)  # Codificar variables categóricas  le = LabelEncoder()  for col in df.select\_dtypes(include='object').columns:      df[col] = le.fit\_transform(df[col])  # Separar features y target  X = df.drop('Target', axis=1)  y = df['Target']  # Escalar  scaler = StandardScaler()  X\_scaled = scaler.fit\_transform(X)  # ✅ Elegir una fila de prueba (por ejemplo, la número 10)  fila = X\_scaled[9].reshape(1, -1)  # Predecir  prediccion = model.predict(fila)  clase = np.argmax(prediccion)  clases = {0: 'Continúa', 1: 'Abandona', 2: 'Se gradúa'}  # Mostrar resultado  print(f"Resultado: {clases[clase]} (probabilidades: {prediccion[0]})") |

## 7. Ejecutar el script

Con el entorno activado:  
python predict.py

Ejemplo de salida:  
Resultado: Abandona (probabilidades: [0.02 0.93 0.05])

## Aplicaciones en Ingeniería de Sistemas

- Sistemas de alerta temprana  
- Dashboard de monitoreo  
- Optimización académica  
- Proyectos de Machine Learning aplicados

**8. RESULTADOS**  
  
Durante la ejecución del archivo main.py, se cargaron y preprocesaron los datos correctamente, luego se entrenó una red neuronal con arquitectura básica y técnica de regularización **Dropout.** El modelo fue entrenado sobre un conjunto de datos obtenidas de Este conjunto de datos cuenta con el apoyo del programa SATDAP - Capacitação da Administração Pública bajo la subvención POCI-05-5762-FSE-000191, Portugal. que tiene características de estudiantes universitarios (como edad, promedio académico, cursos desaprobados, etc.).

Tras el entrenamiento, se obtuvo una precisión del 74% sobre el conjunto de prueba, lo que indica un buen rendimiento general del modelo.

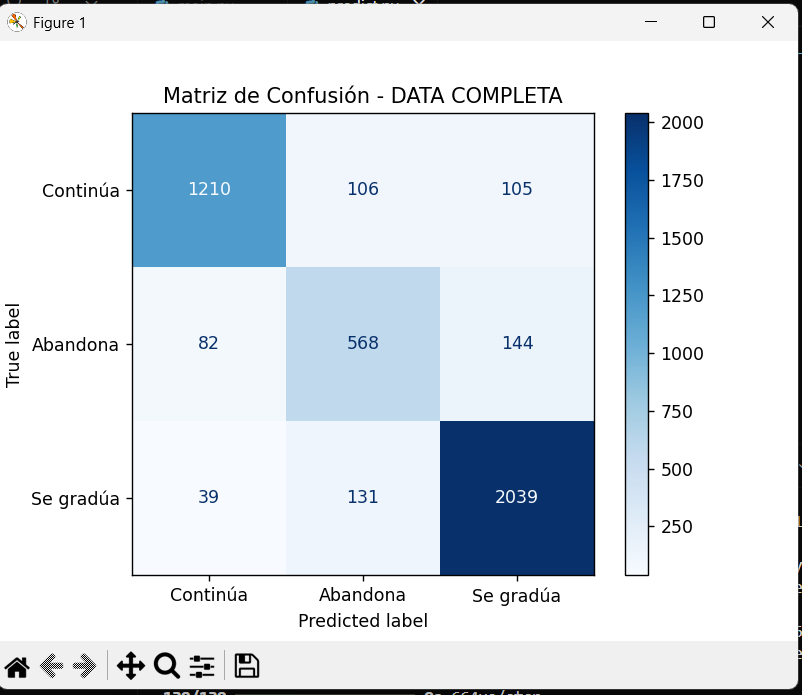
Posteriormente, se utilizó el script predict.py para realizar una predicción sobre la fila número 10 del conjunto de datos. El modelo cargado desde models/dropout\_model.h5 predijo el siguiente resultado:

**Resultado:** Continúa

**Probabilidades**: [0.9678, 0.0322, 0.00001]

Esto significa que el estudiante evaluado tiene una alta probabilidad (96.78%) de continuar sus estudios, según el análisis de la red neuronal.

Este resultado respalda la utilidad del modelo para predecir el riesgo de abandono académico, lo cual puede ser aprovechado por universidades o departamentos académicos como herramienta preventiva.  
  
**Matriz de confusión**

  
  
**Interpretacion:**

Filas: Clase real (lo que realmente ocurrió con el estudiante).

Columnas: Clase predicha por el modelo (lo que la red neuronal predijo).

Diagonal principal: Predicciones correctas.

Fuera de la diagonal: Errores de predicción.  
  
**Primera fila (Continúa):**

1210: El modelo predijo correctamente que estos estudiantes continúan.

106: El modelo predijo que abandonarían, pero en realidad continúan.

105: El modelo predijo que se gradúan, pero en realidad continúan.

**Segunda fila (Abandona):**

82: El modelo predijo que continúan, pero en realidad abandonan.

568: Predicción correcta de abandono.

144: El modelo predijo que se gradúan, pero en realidad abandonan.

**Tercera fila (Se gradúa):**

39: El modelo predijo que continúan, pero en realidad se gradúan.

131: El modelo predijo que abandonan, pero en realidad se gradúan.

2039: Predicción correcta de graduación.

**¿Qué significa esto?**

Valores altos en la diagonal (1210, 568, 2039): El modelo acierta en la mayoría de los casos.

Valores fuera de la diagonal: Son los errores. Por ejemplo, 144 estudiantes que realmente abandonan fueron clasificados como si se graduaran.

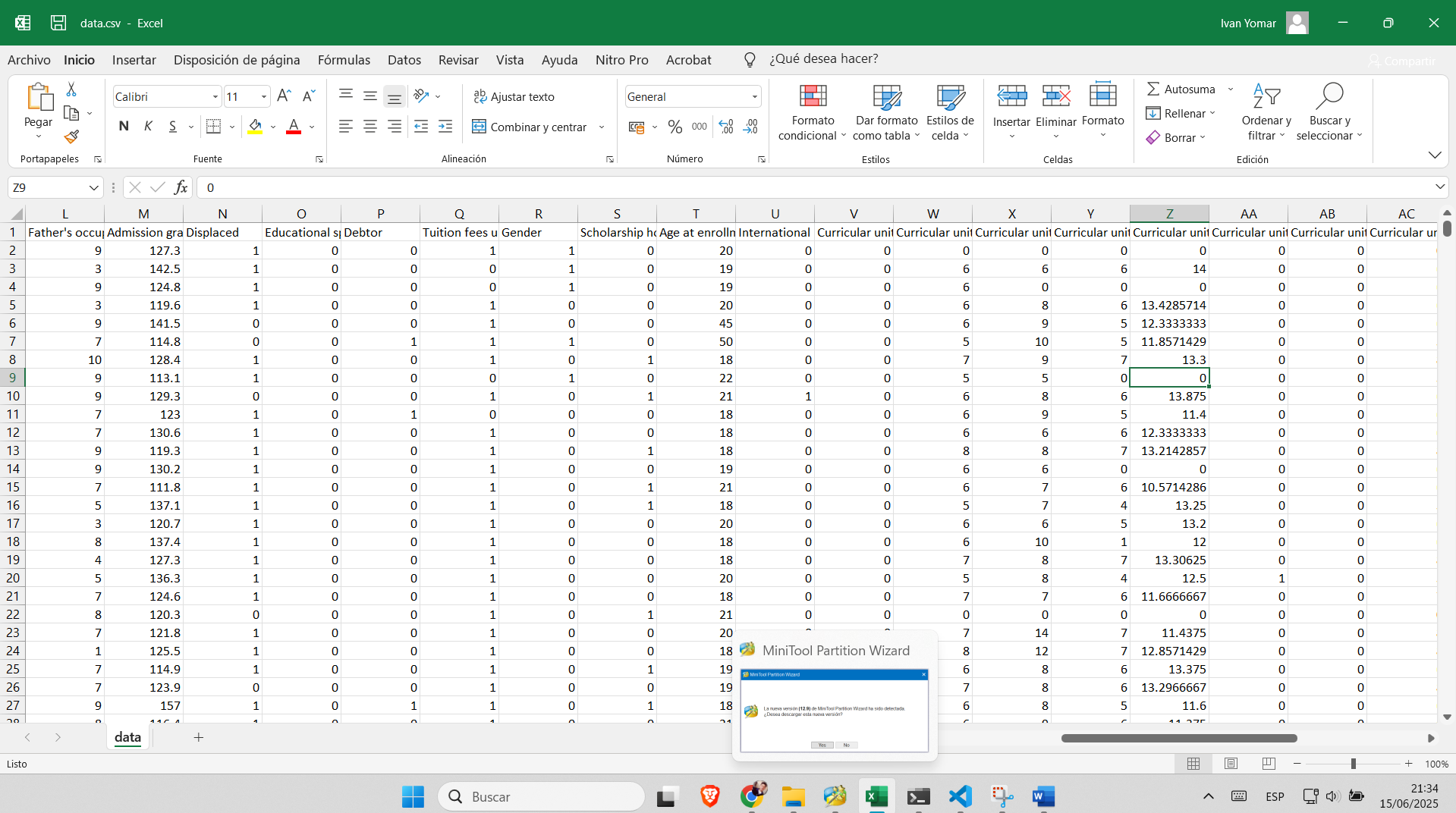
## 9. Conclusión

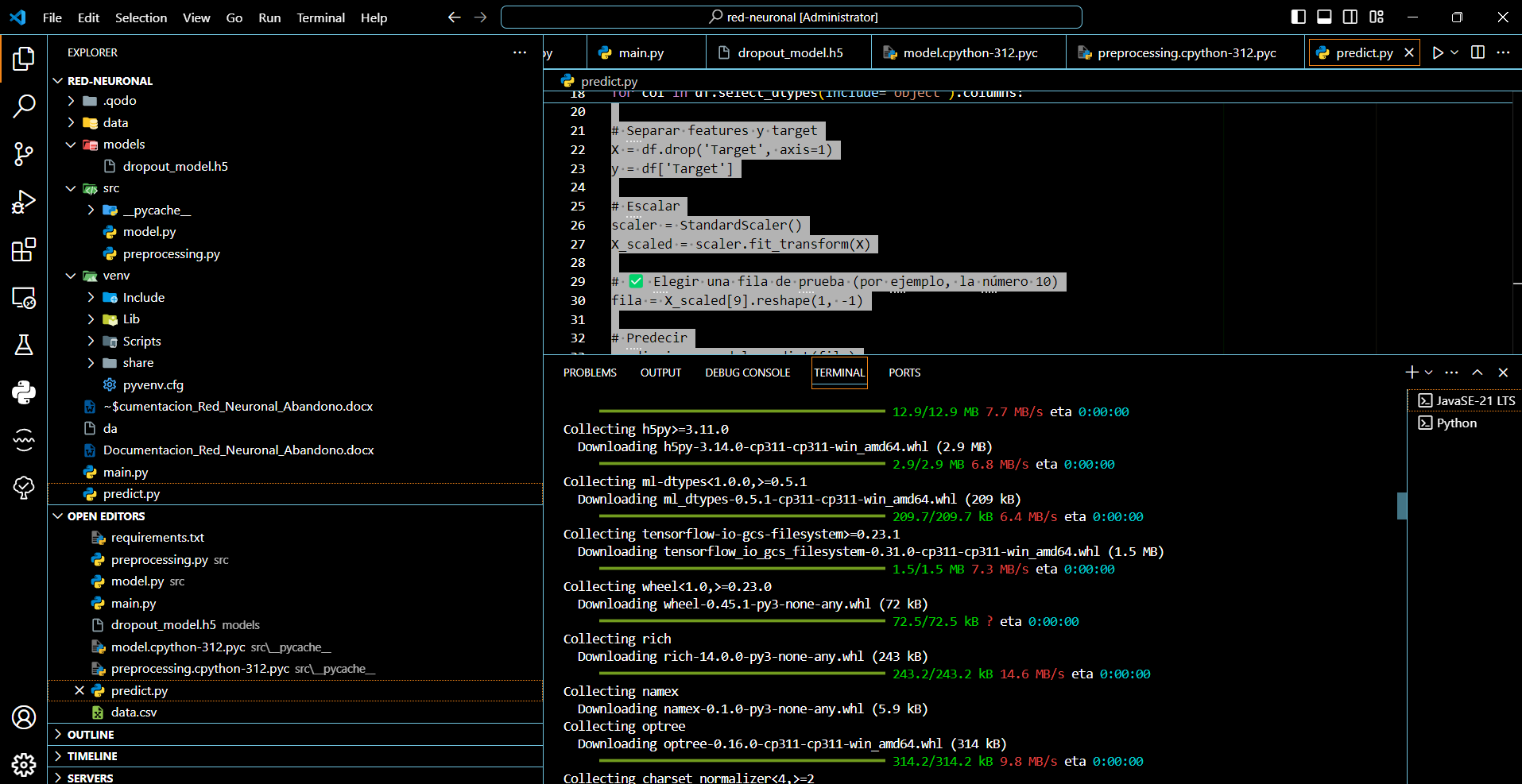
Este proyecto muestra cómo aplicar redes neuronales para resolver un problema social y educativo. Al predecir el abandono estudiantil, los sistemas informáticos pueden integrarse en plataformas web de gestión académica para emitir alertas tempranas. Así, un ingeniero de sistemas puede contribuir al bienestar educativo mediante IA, creando soluciones inteligentes basadas en datos reales.

**10. REFERENCIAS**  
  
Realinho, V., Vieira Martins, M., Machado, J., & Baptista, L. (2021). Predict Students' Dropout and Academic Success [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5MC89.

## 11. ANEXOS

DATA



ENTRENANDO LA RED NEURONAL CON LA DATA   
  
  
PRUEBAS  
