Documentación del Proyecto: 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 o simulados.

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

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

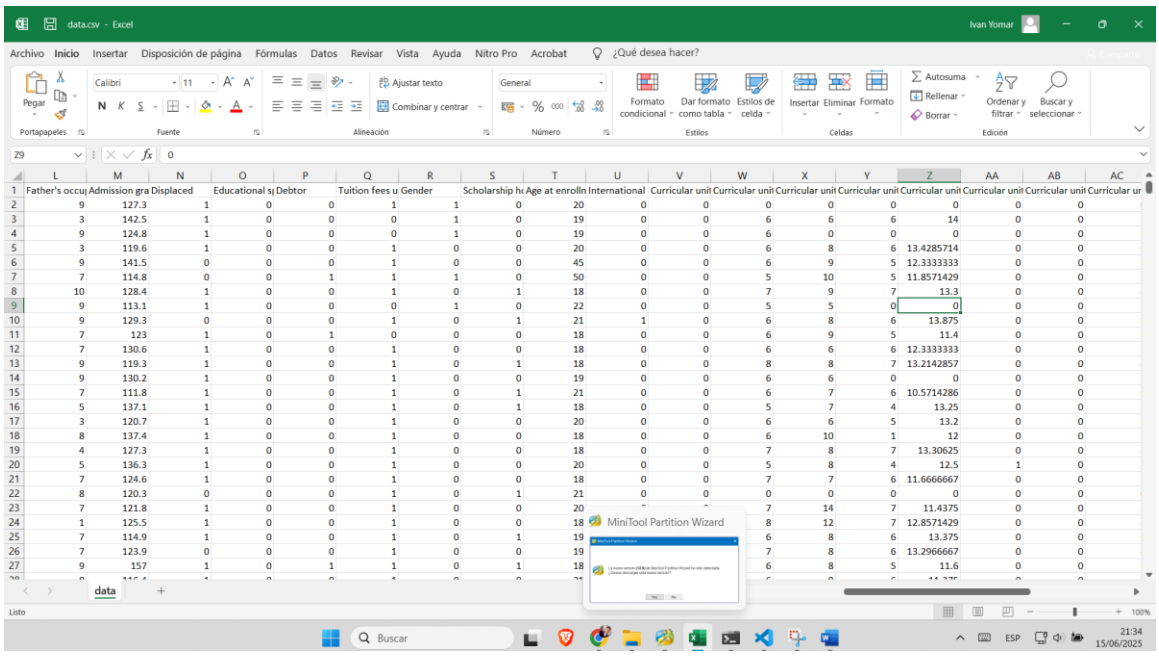
8. Conclusion

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

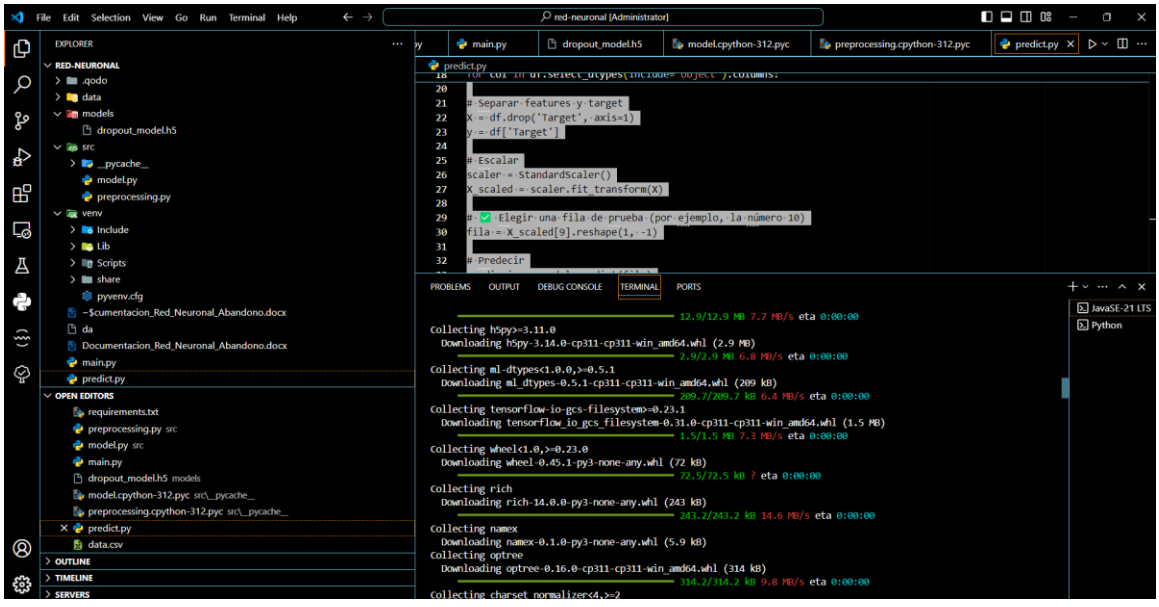
basadas en datos reales.

8. ANEXOS

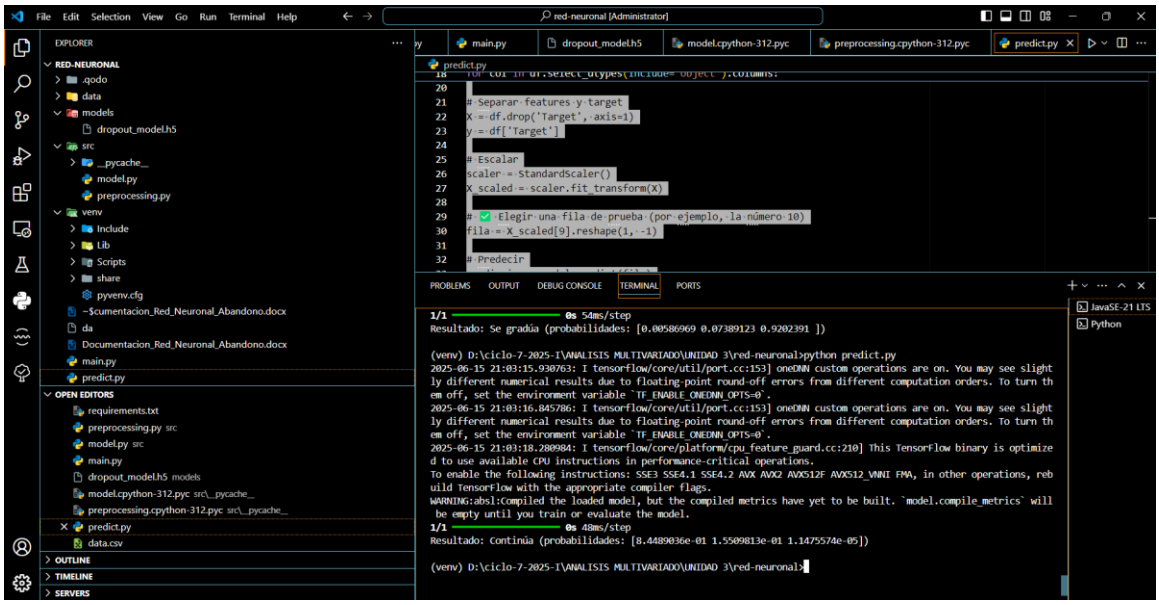
DATA

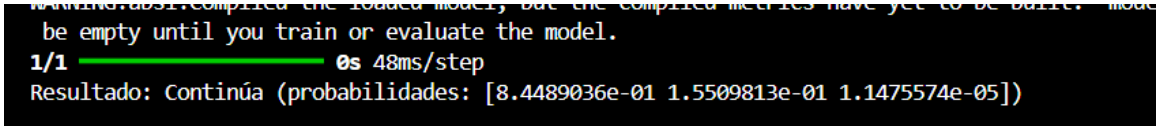


ENTRENANDO LA RED NEURONAL CON LA DATA



PRUEBAS


File Edit Selection View Go Run Terminal Help
red-neuronal [Administrator]
EXPLORER
RED-NEURONAL
.qodo
data
models
dropout_model.h5
src
__pycache__
model.py
preprocessing.py
venv
Include
Lib
Scripts
share
pyvenv.cfg
~$cumentacion_Red_Neuronal_Abandono.docx
da
Documentacion_Red_Neuronal_Abandono.docx
main.py
predict.py
OPEN EDITORS
requirements.txt
preprocessing.py src
model.py src
main.py
dropout_model.h5 models
model.cpython-312.pyc src__pycache__
preprocessing.cpython-312.pyc src__pycache__
predict.py
data.csv
OUTLINE
TIMELINE
SERVERS
main.py
dropout_model.h5
model.cpython-312.pyc
preprocessing.cpython-312.pyc
predict.py
predict.py
# 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
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
JavaSE-21 LTS
Python
1/1 0s 54ms/step
Resultado: Se gradúa (probabilidades: [0.00586969 0.07389123 0.9202391])
(venv) D:\ciclo-7-2025-I\ANALISIS MULTIVARIADO\UNIDAD 3\red-neuronal>python predict.py
2025-06-15 21:03:15.930763: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-06-15 21:03:16.845786: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-06-15 21:03:18.280984: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.
1/1 0s 48ms/step
Resultado: Continúa (probabilidades: [8.4489036e-01 1.5509813e-01 1.1475574e-05])
(venv) D:\ciclo-7-2025-I\ANALISIS MULTIVARIADO\UNIDAD 3\red-neuronal>


be empty until you train or evaluate the model.
1/1 0s 48ms/step
Resultado: Continúa (probabilidades: [8.4489036e-01 1.5509813e-01 1.1475574e-05])