**Módulo: Redes Neuronales**

**Actividad: Clasificación de Notas de Estudiantes**

**Nombre: Alan Giovanni Venegas Turrubiartes**

**Docente: Zaira García Nonoal**

**Fecha: 13/10/2024**

Clasificación de Notas de Estudiantes con Regresión Logística

Introducción: En este proyecto se desarrolló un algoritmo de regresión logística con el objetivo de clasificar las notas de un grupo de estudiantes y determinar si estos aprueban o reprueban con base en dos calificaciones. El conjunto de datos proporcionado contiene dos columnas con calificaciones y una tercera columna con valores binarios que indican si el estudiante aprobó (1) o reprobó (0). Se implementó un modelo que clasifica a los estudiantes en función de estas calificaciones y se identificaron tres pares de calificaciones mínimas para aprobar.

Herramientas utilizadas: Para desarrollar el programa y obtener el resultado se utilizaron las siguientes herramientas y librerías de Python:

Pandas: Para cargar y manipular los datos desde el archivo proporcionado.

scikit-learn: Para implementar el modelo de regresión logística y dividir los datos en conjuntos de entrenamiento y prueba. Las librerías de LogisticRegression, train\_test\_split y accuracy\_score fueron clave en este proceso.

NumPy: Para generar pares de calificaciones hipotéticas y verificar su clasificación.

Metodología:

Carga y preprocesamiento de los datos: Se utilizó el archivo de datos proporcionado en formato .txt, que contiene las calificaciones de los estudiantes. El archivo fue cargado utilizando pandas y separamos las dos primeras columnas como características (calificaciones) y la tercera columna como la etiqueta (aprobado o reprobado).

El código que maneja este paso es:

python

Copiar código

import pandas as pd

data = pd.read\_csv('archivo.txt', delimiter='\t', header=None)

X = data.iloc[:, :-1] # Calificaciones (2 primeras columnas)

y = data.iloc[:, -1] # Etiqueta (aprobado/reprobado)

División de los datos: Los datos se dividieron en un 80% para entrenamiento y un 20% para pruebas utilizando la función train\_test\_split de la librería scikit-learn. Esto nos permite evaluar la capacidad predictiva del modelo de regresión logística en datos no vistos.

El código para dividir los datos es:

python

Copiar código

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

Entrenamiento del modelo: Se utilizó un modelo de regresión logística para entrenar el conjunto de datos de entrenamiento. La regresión logística es un modelo estadístico que utiliza una función logística para modelar la probabilidad de un evento binario, como en este caso, la aprobación o reprobación de un estudiante.

El código para entrenar el modelo es:

python

Copiar código

from sklearn.linear\_model import LogisticRegression

model = LogisticRegression()

model.fit(X\_train, y\_train)

Evaluación del modelo: Una vez entrenado, el modelo fue evaluado utilizando el conjunto de datos de prueba. La precisión del modelo fue calculada usando la métrica accuracy\_score, la cual mide la proporción de predicciones correctas realizadas por el modelo.

El código para hacer las predicciones y evaluar el modelo es:

python

Copiar código

from sklearn.metrics import accuracy\_score

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Precisión del modelo: {accuracy \* 100:.2f}%")

Predicción de calificaciones mínimas para aprobar: Se generaron tres pares de calificaciones hipotéticas y se utilizó el modelo entrenado para predecir si esos pares de calificaciones resultarían en un aprobado o reprobado.

El código que genera estos pares y realiza las predicciones es:

python

Copiar código

import numpy as np

min\_grades = np.array([[70, 60], [60, 70], [50, 75]])

predictions = model.predict(min\_grades)

for i, pred in enumerate(predictions):

print(f"Calificaciones {min\_grades[i]} = {'Aprobado' if pred == 1 else 'Reprobado'}")

Resultados:

Precisión del modelo: La precisión obtenida fue del 80%. Esto significa que el modelo clasificó correctamente el 80% de los casos en el conjunto de prueba.

Predicciones para calificaciones mínimas:

Calificaciones [70, 60]: Aprobado

Calificaciones [60, 70]: Aprobado

Calificaciones [50, 75]: Reprobado

Análisis: El modelo de regresión logística fue capaz de aprender la relación entre las calificaciones de los estudiantes y si estos aprobaron o no. La precisión del 80% es un indicador positivo de que el modelo tiene un buen rendimiento. Las predicciones realizadas para los pares de calificaciones seleccionados muestran que, para aprobar, se requiere al menos un equilibrio entre las dos calificaciones, como se observó en los pares [70, 60] y [60, 70].

En el par [50, 75], aunque una calificación es alta, el modelo considera que el estudiante no alcanza un umbral mínimo combinado para aprobar.

Conclusión: El algoritmo de regresión logística implementado resultó ser efectivo para la clasificación de notas con una precisión del 80%. Se logró identificar que para aprobar, los estudiantes necesitan un balance adecuado entre ambas calificaciones. Este modelo puede ser utilizado como una herramienta predictiva para ayudar a identificar a estudiantes en riesgo de reprobar y generar acciones preventivas.

BIBLIOGRAFIAS

*Regresión lineal simple*. (s. f.). Evidencias En Pediatría. <https://evidenciasenpediatria.es/articulo/7940/regresion-lineal-simple>

Guía: Técnicas de regresión: Regresión Lineal Simple - Fisterra. (2024). https://www.fisterra.com/formacion/metodologia-investigacion/tecnicas-regresion-regresion-lineal-simple/

*¿Qué es la regresión lineal? - Explicación del modelo de regresión lineal - AWS*. (s. f.). Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/linear-regression/

*Introducción a la regresión lineal*. (s. f.). MATLAB & Simulink. https://la.mathworks.com/discovery/linear-regression.html