**Early Prediction of student's Performance   
in Higher Education: A Case Study**

Johan Sebastián Valencia Gil

1. **Introducción.**

Este estudio aborda la problemática del fracaso académico en la educación superior mediante la aplicación de técnicas de aprendizaje automático, con el objetivo de anticipar el riesgo de fracaso y facilitar intervenciones tempranas de apoyo, utilizamos datos al momento de la inscripción, incluyendo información académica, demográfica y socioeconómica. El conjunto de datos desafía la construcción de modelos de clasificación debido a un fuerte desequilibrio en las clases. Se exploraron diferentes modelos de redes neuronales utilizando la librería Tensorflow.

**2. Trabajo previo**

***2.1 Instalación y configuración de ambiente***

Se procedió con la instalación más reciente de Python (3.11.6), seguidamente la instalación y configuración de diversas bibliotecas. Tensorflow para la construcción de redes neuronales, scikit-learn para algoritmos de aprendizaje (convencionales) y pandas para la manipulación eficiente de datos. Con esta integración y configuración de librerías se establece un entorno de desarrollo, facilitando la implementación y evaluación de modelos predictivos.

***2.2 Dataset (normalización y categorización de datos)***

En la primera fase del estudio, nos dedicamos al procesamiento y preparación rigurosa de los datos. Inicialmente se llevó a cabo un análisis exhaustivo del conjunto de datos proveniente de la institución educativa, identificando variables categóricas y variables numéricas. Para las variables numéricas aplicamos técnicas de normalización para asegurar la coherencia y escalabilidad de los datos. Para las variables categóricas (“Dropout”, “Enrolled” y “Graduate”) se implementó una categorización de clases. De esta manera transformamos el problema en una tarea de clasificación y así, tenemos las bases para la construcción y evaluación de modelos predictivos.

**3. Implementación: Modelo de redes neuronales.**

***3.1 Arquitectura***

Se utilizó una arquitectura de red de 3 capas, una de entrada, una de salida y una capa oculta.

* Capa de entrada: 36 nodos para entrada de datos.
* Capa oculta intermedia: 72 nodos.
* Capa de salida: 3 nodos para clasificación.

***3.2 Funciones de activación***

En nuestros datos, se encuentran muchos datos negativos, por lo cual es esencial utilizar funciones de activación que aparte de que permitan el flujo de estos datos, la red neuronal aprenda de los mismos evitando así el problema de las “neuronas muertas”

* Leacky ReLU: Introduce una pequeña pendiente para los valores negativos (para nuestra implementación es 0.005), así ayuda a mitigar neuronas muertas.
* ELU: Tiene una salida suave para valores negativos y una salida exponencial para valores positivos. Gracias a esto nos conduce a un entrenamiento más rápido del modelo.
* Softmax: Se utiliza en la capa de salida de la red neuronal para la clasificación.

***3.3 Hiperparámetros***

* Optimizador: Adam. Es un optimizador predefinido por la librería Tensorflow, podemos modificar sus diferentes parámetros. En nuestro caso solo se realizó el ajuste de su taza de aprendizaje.
* Métricas de evaluación: Se utiliza la precisión como métrica principal para evaluar el rendimiento del modelo.
* Función de pérdida: Entropía cruzada categórica escasa. Se usa en problemas de clasificación múltiple.
* Épocas: Para este estudio se trabajó con un rango de entre 10 y 20.
* Tamaño de lote: Para este estudio se utilizó un tamaño de lote entre 32 y 64. Este parámetro especifica cuantas muestras utiliza antes de realizar una actualización de peso. Si se aumenta o se disminuye de este rango, la precisión se verá muy afectada.

**4. Conclusiones.**

Este estudio se centró en la aplicación de técnicas de aprendizaje automático a través de la construcción y evaluación de un modelo de redes neuronales, se lograron avances significativos, obteniendo una precisión del 72% con la implementación del modelo de red anteriormente expuestos.

  
  
A pesar de los avances logrados, se reconoce como mayor desafío para el entrenamiento de los modelos, el desequilibrio de clases.

**5. Referencias.**

1. <https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success>