MASTERPROF UMH
UNIVERSITAS Miguel Hernández
MÁSTER UNIVERSITARIO EN FORMACIÓN DEL PROFESORADO
ESO Y BACHILLERATO, FP Y ENSEÑANZAS DE IDIOMAS

PREDICCIÓN DEL RENDIMIENTO ACADÉMICO MEDIANTE TÉCNICAS DE MINERÍA DE DATOS

TRABAJO FIN DE MÁSTER

Estudiante: Enrique Alfonso Carreres
Especialidad: Matemáticas
Tutor/a: David Úbeda González
Co-tutor: Julio Alberto Ramos Martínez
Curso académico: 2023-24

Resumen

Las técnicas de EDM (Educational Data Mining) son una herramienta la cual consiste en la implementación de técnicas y algoritmos de minería de datos en bases de datos educativas, la cual hoy en día ha tenido un crecimiento casi exponencial en cuanto a estudios con el propósito de predecir problemas en educación como el rendimiento académico, el fracaso escolar o el abandono prematuro.

Además, proporcionan una variedad de metodología, lo cual nos indica que, por un lado, existe un sinfín de posibilidades para poder probar y estudiar distintas formas de aplicación; y, por otro lado, la inexistencia de un consenso en esta. Eso sí, por lo que respecta al método, prácticamente todos los estudios realizan las mismas fases: extracción de datos, lo cual pueden ser mediante cuestionarios o mediante programas informáticos en plataformas online; depuración y preparación de los datos, donde se procederá a la transformación y, si son irrelevantes, eliminación de atributos; la implementación de un algoritmo para intentar predecir la variable a estudiar y, por último, se procede a analizar dichos resultados de la predicción.

En este trabajo final de máster se procederá a predecir el rendimiento académico de dos datasets correspondientes a escuelas de Portugal y universidades de Bolivia. Para ello, se han hecho uso de 3 algoritmos (Árbol de decisión, AdaBoost y Redes neuronales) donde se han comparado entre sí. Se han obtenido como resultados precisiones bastantes altas, sin distinción entre algoritmos. En conclusión, los 3 algoritmos usados pueden utilizarse para la predicción del rendimiento académico.

Palabras Clave: Minería de datos Educativos, Big Data, Data Analytics, rendimiento académico.

Abstract

Educational Data Mining (EDM) techniques are a tool that consists of implementing data mining techniques and algorithms in educational databases. Today, there has been an almost exponential growth in studies with the purpose of predicting problems in education such as academic performance, school failure, or early dropout.

In addition, they provide a variety of methodologies, which indicates that, on the one hand, there are endless possibilities to be able to test and study different forms of application; and on the other hand, the lack of a consensus on this. However, as far as the method is concerned, practically all studies carry out the same phases: data extraction, which can be done by means of questionnaires or through computer programs on online platforms; data cleaning and preparation, where the attributes will be transformed and, if irrelevant, eliminated; the implementation of an algorithm to try to predict the variable to be studied and, finally, the prediction results are analyzed.

In this master's thesis, the academic performance of two datasets corresponding to schools in Portugal and universities in Bolivia will be predicted. To do this, 3 algorithms have been used (Decision Tree, AdaBoost and Neural Networks) where they have been compared with each other. Quite high accuracies have been obtained as results, without distinction between algorithms. In conclusion, the 3 algorithms used can be used to predict academic performance.

Key words: Educational Data Mining, Big Data, Data Analytics, academic performance.

Índice

1. Introducción

Uno de los retos que afronta la educación hoy en día es la mejora del rendimiento académico del alumnado, haciendo que este pueda alcanzar su máximo potencial y adquirir las competencias necesarias para afrontar los retos del siglo XXI. No obstante, a los profesores se nos hace un mundo saber cuando estamos ofreciendo una buena labor como docente y saber si el rendimiento que están adquiriendo el alumnado es el correcto y si estamos usando las metodologías adecuadas. Además, el creciente número de alumnos que decide abandonar la escuela prematuramente preocupa bastante al profesorado, el cual hace todo lo posible para que este número no incremente intentando metodologías nuevas y adaptaciones curriculares severas.

Normalmente, las técnicas de evaluación del rendimiento académico se basan en trabajos según la situación de aprendizaje en la que se esté trabajando, evaluando además el proyecto final de la situación de aprendizaje. A partir de este, una correcta evaluación del alumno según su nivel de estrés, nivel de gestión de las emociones ante un examen, entre otros; se puede intentar predecir cuál será su nota final. Sin embargo, al tener un único profesor demasiados alumnos, o incluso el tutor del grupo; se hace prácticamente imposible predecir qué nota puede obtener.

Es por ello que surge la necesidad de intentar predecir el rendimiento académico del alumnado, así como detectar lo antes posible la posible disertación escolar del mismo. Así se podrá actuar con la mayor antelación posible.

Con el auge del big data y del maching learning surgieron una serie de técnicas para tratar grandes cantidades de datos e intentar extraer de aquí información útil para empresas y multinacionales. No obstante, estas técnicas se pueden usar para un sinfín de posibilidades. De aquí surgieron los EDM. Dichas técnicas son usadas como tratamiento de una serie de datos del alumnado y, mediante la minería de datos, se extrae información acerca de notas finales, dificultades académicas, mejores metodologías o incluso predecir el absentismo escolar.

Generalmente, se extraen datos socio-demográficos de los alumnos mediante encuestas, donde se incluyen además datos sobre notas de controles, trabajos, entre otros. Una vez se tienen estos datos, se incluye una variable más, la cual se trata de las notas finales. Dicha variable es la que se intenta predecir mediante técnicas de DM (*data mining*). Entre las más usadas, encontramos algoritmos de clasificación como árboles de decisión, redes neuronales o redes bayesianas; todas estas con una base matemática compleja.

A partir de la variable predicha y la real, se comparan, obteniendo unas métricas para comparar los distintos algoritmos y saber cuál de estos se obtiene una correcta predicción.

En el presente trabajo final del Máster se hablará sobre estas técnicas, los métodos de obtención de información del alumnado, los algoritmos usados y su breve explicación, así como las métricas extraídas a partir de los primeros resultados. Además, a partir de unas bases de datos publicadas, se procederá a extraer nuestros propios resultados, comparando diferentes algoritmos (a saber, árbol de decisión, AdaBoost y red neuronal) y valorando cuál es mejor para la tarea de la predicción del rendimiento académico.

2. Objetivos

Como objetivo principal del presente trabajo será ilustrar a futuros docentes una técnica de ayuda al profesorado, el cual servirá para conocer aún más al alumnado, así como conocer si se está haciendo una correcta labor docente o se necesita cambiar de metodología; además de ayudar a psicopedagogos a identificar el fracaso y sus motivos antes de que ocurra.

Como objetivos específicos serán lo presentes a continuación:

- Conocer los diferentes algoritmos para la predicción del rendimiento académico.
- Comparación de los diferentes algoritmos y observar la clasificación y su precisión.
- Conocer las distintas formas de obtención de datos de los alumnos, los distintos algoritmos y los distintos resultados de los estudios recopilados.

Con los objetivos, surgen varias hipótesis a comprobar en el presente estudio:

- Se podrá predecir el rendimiento académico, reagrupando la variable a predecir.
- El algoritmo de redes neuronales será el que mejor para la tarea de predicción del rendimiento académico.

3. Marco teórico: predicción del rendimiento académico

Se entiende el rendimiento académico como [23]:

- Actitudes y habilidades cognitivas, que incluyen, entre otros aspectos, la atención, memoria, comprensión verbal, procesamiento de información, motivación, autoconcepto y satisfacción.
- Comportamientos académicos, que abarcan, entre otros, la organización, planificación y asistencia.
- Resultados académicos, que incluyen las puntuaciones en las materias de enseñanza.

Por otro lado, se define el éxito académico como una combinación de múltiples factores, a saber, rendimiento académico, logro de objetivos de aprendizaje, adquisición de habilidades y competencias, satisfacción, persistencia y rendimiento postuniversitario [26]. Dicho modelo queda descrito en la *fig. 1.*

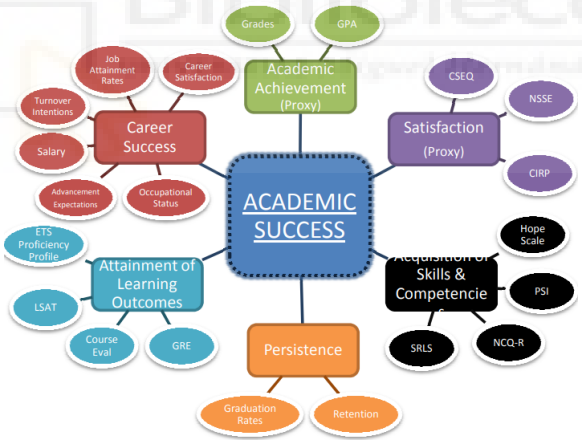


Figura 1: *Modelo del éxito académico, según York et. al., 2015 [26].*

Por tanto, se puede entender la preocupación que tiene el personal docente a intentar predecir el rendimiento académico de sus alumnos para así optimizar al máximo sus clases y ofrecer la máxima calidad a sus alumnos. Es por ello que se hacen uso de técnicas de maching learning para llevar a cabo este trabajo.

Del mismo modo, se define minería de datos a la obtención de información, como patrones, estructuras, asociaciones a partir de una gran cantidad de datos almacenada en bases de datos [24]. No es de extrañar que dichas técnicas se utilicen para detectar anomalías en el rendimiento académico, así como detectar el fracaso escolar [15] y las dificultades académicas [21].

En una revisión bibliográfica acerca de las EDM y las LA (*Learning analytics*), destaca 4 grandes grupos de varias aplicaciones del data mining en educación [1]:

- **Computer-supported learning analytics (CSLA)**: uso del análisis estadístico de los datos para analizar la información de los estudiantes, buscar y aprender colaborativamente el comportamiento en un curso.
- **Computer-supported predictive analytics (CSPA)**: se focaliza en el uso de funciones predictivas o variables continuas para sugerir vías para mejorar el rendimiento académico de los alumnos, además de evaluar la contribución de los materiales de enseñanza para estos.
- **Computer-supported behavioral analytics (CSBA)**: la gran mayoría descubren modelos de comportamientos, acciones y conocimientos de los estudiantes.
- **Computer-supported visualization analytics (CSVA)**: se focaliza en el método de visualizar la información, para vislumbrar información relevante y para contribuir en decisiones de la información.

Donde en dicho estudio concluye que los EDM y LA son comúnmente usadas para dar oportunidades y soluciones a varios problemas de enseñanza. Y es que en otra revisión bibliográfica [14], analizando 133 artículos (82 seleccionados), desde la década de los noventa hasta el 2018, donde hubo un repunte de artículos. En dicho artículo hace hincapié en la procedencia de los datos, los cuales son enseñanzas secundarias y superiores, donde se proporciona enseñanzas tradicionales, e-learning y blended learning.

Es más, si analizamos un poco más los artículos, se puede destacar de las EDM fortalezas, debilidades, oportunidades y amenazas [25][19]. Todas estas están recogidas en la *Tabla I*. Es por ello que se debe tener cuidado con el tratamiento de los datos y analizar críticamente si el método seguido es el correcto o no, además de adquirir todos los permisos necesarios para su correcta puesta en práctica.

Tabla I: *Tabla de Fortalezas, Debilidades, Oportunidades y Amenazas de los métodos de EDM [25][19].*

Fortalezas	**Debilidades**
- Gran volumen de datos disponibles. - Uso de algoritmos poderosos y validados ya existentes. - Múltiples formas de visualizar los datos. - Modelos más precisos de los usuarios para la mejora y personalización de los sistemas. - Encontrar momentos críticos y patrones de aprendizaje. - Obtener una visión de las estrategias de aprendizaje y sus resultados.	- Errores en la interpretación de los resultados debido a factores humanos. - Fuentes de datos heterogéneos. No existe todavía un estándar para los datos. - Los resultados en su mayoría son cuantitativos. Los métodos cualitativos no han brindado resultados significantes. - Sobrecarga de información. Sistemas complejos. - Incertidumbre debido a que solo los docentes o instructores con cierto nivel de habilidades pueden interpretar correctamente los resultados.
Oportunidades	**Amenazas**
- Estandarización de los datos y mejora de compatibilidad entre las diferentes aplicaciones y herramientas. - Aprendizaje multimodal y afectiva. - Capacidad de autoaprendizaje en sistemas inteligentes y autónomos. - Integración de los resultados obtenidos con otros sistemas de toma de decisiones. - Modelo de aceptación, describiendo usabilidad, expectativas, confiabilidad, entre otros.	- Aspectos éticos como privacidad de los datos. - Sobe-análisis (*overfiting*). - Posibilidad de errores en la clasificación de patrones. - Confiabilidad: resultados contradictorios durante la implementación de modelos ya establecidos.

Además, destacar el creciente número de estudios del uso del big data en la educación, donde las palabras clave más usadas para estos estudios son *Big*, *data*, *learning*, *education* y *analytics* [16].

3.1. Fases de trabajo

La forma de trabajar las técnicas de minería de datos educativos se puede dividir en 4 fases, cada una igual de importante. En realidad, existe una fase 0, la cual es el análisis del problema. es decir, el planteamiento inicial de problema y el porqué se va a usar los EDM para la resolución de dicho problema. Esta fase, pese a que es un tanto clara, es de las más importantes, pues de esta derivan las demás.

Obtención de los datos

Una vez analizado el problema que se va a tratar se debe planear que datos vamos a recabar. No existe un estándar de qué datos se deben recoger y cuanta cantidad se debe recoger. Aquí existe una libertad para que el investigador/profesor pueda recabar tanta información como cree necesario. esto se puede considerar una ventaja, pues puedes incluir tantas variables como desees, y una desventaja, como se comentaba en la *Tabla I*, pues no existe un consenso con qué datos recabar y ocasiona una heterogeneidad en los resultados.

Los datos se pueden recabar de muchas formas, ya sea por encuestas y las notas académicas de un curso en un instituto, mediante redes sociales[21] o incluso mediante la plataforma Moodle [4][13], donde esta última tiene bastantes atributos para considerar. Lo más usado es la primera, donde se recaudan información directamente de los alumnos mediante encuestas.

Además, existen aplicaciones para recabar información acerca de la interacción de los alumnos en plataformas digitales. El nombre de la aplicación es *DataShop* y ofrece información como interacción de documentos, clicks-stream data, entre otros [11]. Por otro lado, podemos recabar la información a medida que el curso va avanzando, para así detectar el fracaso escolar y el rendimiento académico en etapas tempranas [3].

Por último, destacar que los datos no tienen por qué ser de una edad en concreto. Pueden recabar información desde institutos [15] hasta grados universitarios [13][9][17].

Comprensión de los datos

Una vez se tienen los datos recabados en una base de datos, se procede a lo que se conoce como *sanear los datos*. Es decir, si se encuentra alguna variable vacía porque no se ha llegado a completar, se descarta. Seguidamente, se procede a cambiar los valores de las variables de texto a números, para una mejor comprensión de los datos y, por ende,

obtener mejores resultados.

Por ejemplo, en la tesis doctoral de Carlos Márquez [15], se nos habla de un rebalanceo de datos, quitando aquellas variables que no son relevantes para la predicción del rendimiento académico, además de transformar las notas a ciertos rangos.

Esta fase tiene una relevancia significativa, pues nos determinará la exactitud de los resultados, así como la calidad de la predicción.

Modelización

Una vez se tienen los datos recogidos y en una base de datos limpia, se procede a la modelización. Aquí existen muchos algoritmos para usar, y muchos artículos donde comparan los resultados de cada uno de los algoritmos.

Ente los más usados tenemos el algoritmo de árbol de decisión C4.5, el cual utiliza la ganancia de información a medida que se ejecuta. Utiliza la técnica del *prunning*, el cual consiste en que el árbol de decisión crece y más adelante se van borrando ramas según la información que aporta. Concretamente, estima el error de cada subárbol, y reemplaza aquellos cuyo error es mayor que una rama con menor error [22].

Otros de los más usados son las redes neuronales, los cuales están diseñados para realizar tareas cognitivas como el aprendizaje y la optimización, basados en investigaciones sobre la naturaleza del cerebro [18]. Se tienen los elementos básicos como el conjunto de entradas, los pesos sinápticos, la regla de propagación, la cual proporciona el valor del potencial postsináptico en función de sus pesos y entradas; la función de activación, la cual proporciona el estado de activación actual de la neurona en función de su estado anterior y el potencial postsináptico; y la función salida [2][6][10].

Las neuronas se pueden agrupar en capas y dentro de estas pueden formar grupos o *clusters*. Se tiene:

- **Capa de entrada**: Neuronas que reciben datos o señales del entorno.
- **Capa oculta**: Conectada a capas de entrada y salida o a otras capas ocultas.
- **Capa de salida**: Proporcionan la respuesta de la red neuronal.

Además, se destaca la fase de entrenamiento de la neurona, ya que dicha fase se usa para que se reajusten los pesos sinápticos [2]. Se destacan dos tipos de aprendizaje: supervisado y no supervisado; pero existen otros tipos.

Otros algoritmos utilizados para esta tarea son el algoritmo de vecinos más cercanos, el cual consiste en ver que atributos son más cercanos a los datos que se intentan predecir; los sistemas multiclasificadores, los cuales combinan varios algoritmos para poder obtener resultados más finos a la hora de predecir [9]; regresión logística binaria, la cual permite estudiar la dependencia funcional entre una variable dependiente categórica y un conjunto de variables independientes; y redes bayesianas, las cuales se fundamentan en la teoría de la probabilidad y combinan la potencia del teorema de Bayes con la expresividad semántica de un grafo [17].

Por último, destacar el programa WEKA, una aplicación la cual ofrece a sus usuarios la posibilidad de analizar una gran cantidad de datos con varios algoritmos a su disposición.

Evaluación de resultados

Una vez se obtienen los resultados de la predicción, se procede al análisis de la predicción y obtención de medidas para una correcta interpretación del proceso de predicción.

Dichas medidas surgen, en su mayoría, de la **matriz de confusión**. Dicha matriz ofrece de un vistazo las predicciones hechas correctamente y las que ha predicho mal, comparando la predicción con el valor real. Como se puede observar en la *Tabla II*, en su diagonal se encuentran las predicciones correctas, mientras que en lo que resta son las predicciones erroneas o falsas. Dicha matriz puede ser 2×2 o también puede ser $n \times n$, según los grupos en los que se organiza la variable a predecir.

Tabla II: *Matriz de confusión genérica.*

	Variables predichas	
Variables reales	True Positive	False Negative
	False Positive	True Negative

Una vez se tiene la matriz de confusión, a partir de esta se pueden obtener medidas como la precisión (ec. 1), la tasa de verdaderos positivos o sensibilidad (ec. 2), la tasa de verdaderos negativos o especificidad (ec. 3) y la media geométrica, la cual proporciona una medida de tendencia central usada en conjuntos de datos desbalanceados (ec. 4) [15].

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$TP_{rate} = \frac{TP}{TP + FN} \tag{2}$$

$$TN_{rate} = \frac{TN}{TN + FP} \tag{3}$$

$$GM = \sqrt{TP_{rate} TN_{rate}} \tag{4}$$

Además, existen otras métricas bastante interesantes como son el área bajo la curva ROC, la cual consiste en el índice conveniente de la exactitud global de la prueba (ec. 5) y el coeficiente de Cohen Kappa, el cual es un coeficiente estadístico que permite medir la concordancia entre los resultados de dos o más variables cualitativas (ec. 6) [17].

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} = \frac{TP_{rate} + TN_{rate}}{2} \tag{5}$$

Donde:

$$FP_{rate} = \frac{FP}{FP + TN} = 1 - TN_{rate}$$

$$\kappa = \frac{P_0 - P_e}{1 - P_e} \tag{6}$$

Donde:

$$P_0 = \frac{TP + TN}{N}; P_e = \frac{ac + bd}{N^2}$$

$$a = TP + FP; b = FN + TN; c = TP + FN; d = FP + TN$$

$$N = TP + TN + FP + TN$$

El valor de AUC variará entre $0,5$ y 1, donde valores cercanos a $0,5$ indicará que la prueba no tiene ningún criterio para discriminar entre atributos y valores cercanos a 1 se tendrá una prueba con la cual el algoritmo es capaz de hacer una predicción correcta.

Por lo que respecta al coeficiente de Cohen kappa, se tienen los valores recogidos en la *Tabla III* [12].

Tabla III: *Tabla de valores del coeficiente de Cohen kappa [12].*

κ	Grado de concordancia
$< 0{,}00$	Sin acuerdo
$> 0{,}00 - 0{,}20$	Insignificante
$0{,}21 - 0{,}40$	Discreto
$0{,}41 - 0{,}60$	Moderado
$0{,}61 - 0{,}80$	Sustancial
$0{,}81 - 1{,}00$	Casi perfecto

4. Metodología

En el presente trabajo final de máster se procederá a analizar dos bases de datos diferentes sobre estudiantes y sus calificaciones. En esta sección se comentará la metodología seguida, variables y programas utilizados; además de qué métricas se han usado para la comparación de algoritmos.

4.1. Datasets

Los *datasets* utilizados para el presente trabajo serán uno con datos de alumnos de Portugal, donde contiene datos correspondientes a dos escuelas Portuguesas, cuyo contenido son atributos como las notas, datos socio demográficos y familiares. Además, están provistas por dos asignaturas en concreto (matemáticas y lengua portuguesa) [5]. Presenta variables con texto y las notas finales están sobre 20. Debido a la cantidad de datos en el dataset, se ha decidido hacer un cambio en los valores de la nota final, haciendo que este sean solo dos valores (aprobado o suspendido) y 5 valores (suspenso, suficiente, bien, notable y excelente). Se tienen todas las variables y sus cambios en la *Tabla VII* del anexo A.

Por otro lado, se tiene el dataset de alumnos de grados de ingeniería de distintas universidades de Bolivia. En dichos datos se encuentra atributos socio demográficos y notas relacionadas con dos exámenes que marcan la vida de un estudiante. Uno de ellos es el test nacional estandarizado, realizado en el último año de instituto, evaluando cinco asignaturas generales (matemáticas, lectura crítica, competencia ciudadana, biología e inglés). El examen se llama Saber 11. En segundo lugar, se tiene el examen del último año de carrera en ingeniería (SABER PRO). Es similar al Saber 11, evaluando 5 asignaturas genéricas (Lectura crítica, razonamiento cuantitativo, competencias civiles, comunicación

escrita e inglés). Con todos estos datos, se procesa además las notas de los proyectos de ingeniería y la puntuación global. Existen varios atributos de texto, por lo que se procederá a cambiar dichos atributos por números, además de eliminar algunos atributos irrelevantes. Todos los atributos y sus cambios se exponen en la *tabla VIII* en el anexo A [7] [8].

4.2. Algoritmos

Una vez se tienen los datos preparados y en formato numérico, se procede a aplicar los algoritmos necesarios para la predicción. Antes de nada, se separan los atributos en 4 matrices, 2 de ellas correspondientes a la matriz de variables independientes (X) y las 2 restantes a la matriz de variable dependiente o variable a predecir (Y). Estos conjuntos de dos matrices son, por un lado, matrices para entrenar al algoritmo para la predicción y por otro, matrices para probar si funciona y predice correctamente el algoritmo.

Posteriormente, se procede a aplicar los algoritmos. En este caso se aplicarán 3 algoritmos distintos. El algoritmo de decisión de arbol C4.5, AdaBoost y Redes neuronales. Los atributos están explicados a continuación:

- **Árbol de decisión**
  - *Criterion = entropy*. Se hace uso de este criterio de ganancia de información.
  - *Max_ depth = 3*. Máximo de profundidad a 3 niveles.
- **AdaBoost**. Dicho algoritmo hace uso de un algoritmo más sencillo y lo ejecuta tantas veces como sea necesario (o cuando alcance el máximo de iteraciones), asignando un pesa a cada variable para saber cual está bien predicha o no.
  - *n_ estimators*. Número máximo de iteraciones.
  - *Learning_ rate*. Cantidad que se actualiza a cada peso después de cada iteración.
- **Redes neuronales**. Para dicho algoritmo se han puesto los valores propuestos en el estudio de (Álvarez, 2019), modificando únicamente el número de capas ocultas [2].

El código se efectuará mediante el lenguaje de programación Python y mediante la librería de Scikit-learn para utilizar los algoritmos [20]. En el anexo B se encontrarán los dos programas utilizados.

4.3. Métricas

Una vez se tiene realizada las predicciones oportunas, se procede a calcular las métricas correspondientes para una correcta interpretación de los resultados.

Para el caso de los datasets de Portugal con G3 binario (suspendido/aprobado) se calculará la precisión, el ratio de valores verdaderos, la media geométrica, el área bajo la curva ROC y el coeficiente de Cohen kappa. La librería sklearn tiene ya incorporadas todas las métricas mencionadas anteriormente, por lo que no es necesario calcular mediante las fórmulas antes mencionadas en el apartado de Evaluación de resultados. En el siguiente caso de 5 valores, se calcularán la precisión, el ratio de valores verdaderos, la media geométrica y el coeficiente de Cohen kappa.

Por lo que respecta al dataset de Bolivia, se obtendrán las matrices de confusión y, a partir de estas, se calculará la precisión, la media geométrica y el coeficiente de Cohen kappa.

Destacar que, en el caso de haber un resultado donde haya un 0 en el ratio de valores verdaderos, este no se tomará en cuenta para el cálculo de la media geométrica, pues dará un valor de 0.

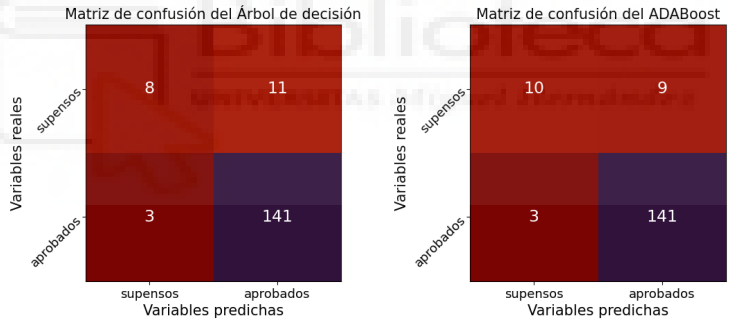
5. Resultados

Siguiendo con el procedimiento antes mencionado, se obtienen los siguientes resultados separados en los datasets para su mayor comprensión.

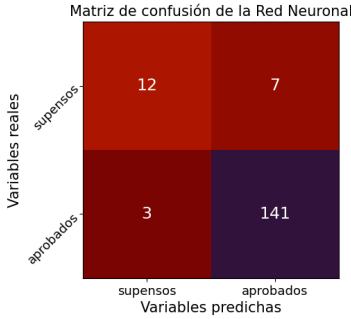
5.1. Dataset de Portugal

Tabla IV: *Resultados para G3 con formato suspenso/aprobado*

	Árbol de Decisión	**AdaBoost**	**Red Neuronal**
Precisión	91,41 %	92,64 %	93,87 %
TA$_{rate}$	97,92 %	97,92 %	97,92 %
TS$_{rate}$	42,11 %	52,63 %	63,16 %
GM	0,6421	0,7179	0,7864
AUC	0,7	0,75	0,81
κ	0,4897	0,5858	0,6722



(a) *Árbol de decisión* **(b)** *AdaBoost*

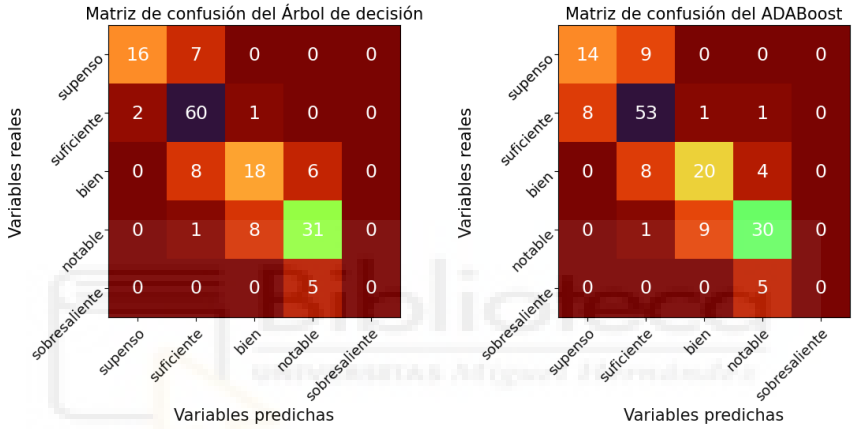


(c) *Redes Neuronales*

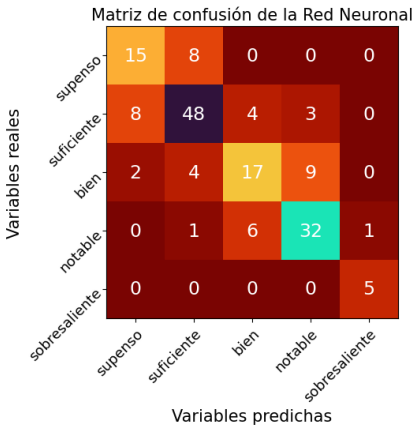
Figura 2: *Matrices de confusión del dataset de Portugal para G3 binario.*

Tabla V: *Resultados G3 con formato de 5 valores*

	Árbol de Decisión	AdaBoost	Red Neuronal
Precisión	76,69 %	71,78 %	71,78 %
T0$_{rate}$	69,57 %	60,87 %	65,22 %
T1$_{rate}$	95,24 %	84,13 %	76,19 %
T2$_{rate}$	56,25 %	62,50 %	53,13 %
T3$_{rate}$	77,50 %	75,00 %	80,00 %
T4$_{rate}$	0,00 %	0,00 %	100 %
GM	0,5374	0,4899	0,4595
κ	0,67	0,61	0,62



(a) *Árbol de decisión* (b) *AdaBoost*



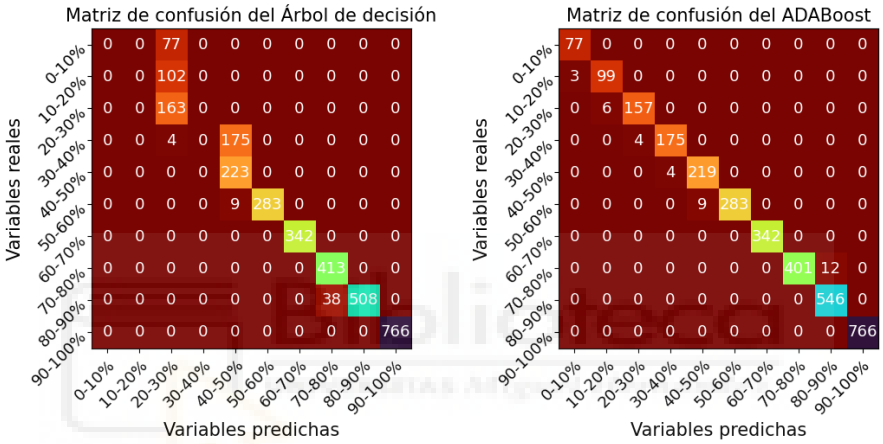
(c) *Redes Neuronales*

Figura 3: *Matrices de confusión del dataset de Portugal para G3 con 5 valores.*

5.2. Dataset de Bolivia

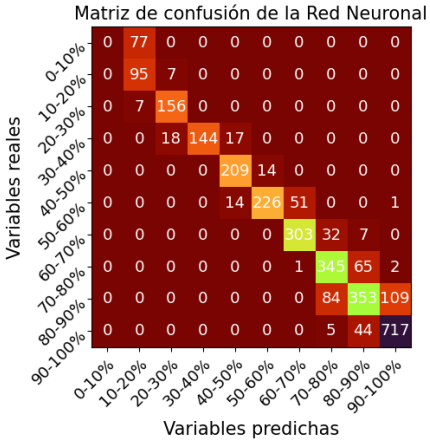
Tabla VI: *Resultados de los diferentes algoritmos con el dataset de Bolivia*

	Árbol de Decisión	AdaBoost	Red Neuronal
Precisión	86,95 %	98,78 %	82,11 %
GM	0,9495	0,9190	0,4827
κ	0,8473	0,9857	0,7907



(a) *Árbol de decisión*

(b) *AdaBoost*



(c) *Redes Neuronales*

Figura 4: *Matrices de confusión del dataset de Bolivia.*

6. Discusión y conclusión

6.1. Discusión de los resultados

Por lo que respecta al dataset de Portugal, a primera vista se puede observar que, debido a la cantidad de campos para clasificar y la cantidad de datos que se tienen, es mejor renombrar la columna a predecir G3 a formato aprobado/suspenso, pues tiene menos valores que predecir y puede perfilar en los resultados.

Si nos centramos un poco más en los resultados binarios (*Tabla IV*), se puede observar que la red neuronal consigue una mayor precisión en cuanto a la predicción de los resultados, pero esto no se diferencia mucho de los resultados de los otros dos algoritmos; siendo una tasa de verdaderos aprobados prácticamente igual. Dicho esto, se observa además en las matrices de confusión (fig. 2), el mismo número de verdaderos aprobados y solo se diferencian unas matrices de otras por la cantidad de verdaderos suspensos. Estos resultados también se pueden observar en los valores del Área bajo la curva ROC y el coeficiente de Cohen Kappa, teniendo como mayor valor el obtenido con la red neuronal. En cuanto a los valores de estos, el valor de AUC de los tres algoritmos nos indica que la exactitud global de la prueba es favorable. En cuanto al valor de κ, se observa un grado de concordancia moderado-sustancial de la clasificación predicha con la clasificación observada. De este modo, se puede decir que los clasificadores presentados son fiables para los datos recogidos.

Si aumentamos la cantidad de variables en la columna G3, se puede observar una disminución de la precisión, pero, aun así, es bastante alta, teniendo con el algoritmo de árbol de decisión C4.5 con mayor precisión, seguido de los otros dos algoritmos muy cerca. Si se observa la *Tabla 3*, se puede reconocer que la tasa de verdaderos sobresalientes tiene un 0 % de acierto en algoritmos como árbol de decisión o AdaBoost, mientras que en red neuronal tiene una tasa de acierto del 100 %. Esto es debido a la cantidad de iteraciones máximas en estos dos últimos pasos. En cuanto al coeficiente κ, tendremos valores entre 0,6 y 0,7, por lo que el grado de concordancia es moderado-sustancial, con lo cual se puede decir que, para los valores de los datos obtenidos son fiables los algoritmos utilizados.

Por lo que respecta al dataset de Bolivia, se puede observar cómo, debido a la gran cantidad de datos que se tienen (12411 estudiantes), obtenemos resultados más precisos que si se aumentara la clasificación de G3 del dataset de Portugal.

Centrándonos en este dataset (*Tabla VI*), se observa como el algoritmo AdaBoost se obtiene la mayor precisión sin llegar al overfitting, teniendo el algoritmo de Red Neuronal

el peor resultado, sobrepasando el 80 %, por lo que es muy buen resultado. Además, si se observa el coeficiente de Cohen κ, vemos como ronda el 0,8, incluso cercanos al 1 como se observa en el algoritmo de AdaBoost. Se pueden observar este hecho en la fig. 4.

Esto puede deberse a la cantidad de variables a considerar, pues al ser un algoritmo sencillo como el AdaBoost (no tan sencillo como el árbol de decisión) tiene mayor capacidad para predecir con mayor precisión. En cambio, una red neuronal, a medida que se aumentan las variables, puede necesitar más iteraciones para poder aumentar la precisión, aumentando así el tiempo de ejecución. Se tendría que valorar si tarda demasiado en ejecutar y la precisión mínima que se requiere.

6.2. Conclusión y líneas a futuro

Como se ha observado a lo largo de este trabajo, los EDM llevan estudiándose bastante tiempo, obteniendo resultados bastante prometedores, diversos y con un sinfín de algoritmos.

Desde las diferentes formas de obtención de datos como con la herramienta Moodle, repositorios de instituciones, notas de los alumnos, cuestionarios a papel u online; pasando por los diferentes ítems y variables que se pueden estudiar (socio demográficas, académicas, personales, ...); hasta los distintos algoritmos que se pueden utilizar para llevar a cabo dicha clasificación. En dicho trabajo se han puesto a prueba 3 algoritmos distintos bastante usados para la clasificación y predicción de datos, teniendo que los tres son muy buenos para llevar a término el trabajo de predecir el rendimiento académico, donde se han obtenido resultados muy parecidos, diferenciándose en poco menos que un 10 %.

Sin embargo, pese a que es una técnica de análisis del rendimiento académico bastante útil, sobre todo para los profesores, la cual puede proporcionar datos no solo del rendimiento académico, sino también de absentismo o dificultades académicas, no se está poniendo en práctica en institutos. Esto puede deberse a la complejidad y a los conocimientos que se requiere para la puesta en práctica de dicha técnica.

Es por ello que, para una línea a futuro de este trabajo, se propone diseñar una aplicación de escritorio la cual tenga una interfaz sencilla, donde se importen una base de datos en formato Excel, csv, etc.; y eligiendo el algoritmo que se desee, pueda predecir el rendimiento académico, absentismo, dificultades académicas, etc. Una de las dificultades que encuentro será al momento de entrenar al algoritmo, pues necesitaría gran cantidad de datos y eso puede llevar al profesor unos años para poder aplicar dicho algoritmo. En vez de importar, se podría proporcionar ciertas variables de serie implementadas en la

aplicación, donde se tendría un repositorio datos ya usados para entrenar el algoritmo y así, el profesor podría usar el algoritmo para predecir.

Otra línea a futuro propuesta sería el diseño de un plan de ejecución y detección temprana de dificultades académicas. Un plan donde tenga varias fases (inicio del curso, primer trimestre (mitad y final), segundo trimestre (mitad y final), tercer trimestre), en las cuales se estudien varias variables como las dificultades en cada materia, variables socio demográficas, nivel de estrés, motivación, etc. Así, con estas variables, se podrán obtener ciertos resultados y predecir el rendimiento del alumno y poder actuar a tiempo. En conclusión, las EDM son unas herramientas muy útiles para el profesorado. Usadas adecuadamente, pueden convertirse en una gran aliada para poder ofrecer al alumnado una buena educación y poner en práctica distintas metodologías activas y analizarlas para observar cuál es la adecuada para nuestro alumnado.

Referencias

[1] Aldowah, H., Al-Samarraie, H. and Fauzy, W.M. (2019). Educational data mining and learning analytucs for 21st century higher education: A review and synthesis. *telematics and informatics*, 37, 13-49. DOI: https://doi.org/10.1016/j.tele.2019.01.007

[2] Álvarez, R. (2019). *Predicción del rendimiento académico en las Matemáticas de la educación secundaria mediante Redes Neuronales*. Universidad Nacional de Educación a Distancia, Departamento de Física Fundamental. [Trabajo Final de Máster. UNED].

[3] Álvarez-Aguilar, J.F. (2020). Minería de datos educativos: una propuesta de innovación en la inspección educativa. *Supervisión 21*, 57. ISSN: 1886-5895. http://usie.es/supervision-21/

[4] Cavaller, D.G., Ortega, C.D., Dueñas, E. and Sosa, H. (2020). *Educación Digital y minería de datos educativos*. [Simposio]. VII Simposio Argentino sobre Tecnología y Sociedad (STS 2020) - JAIIO 49 (Modalidad virtual), La Plata, Argentina. http://sedici.unlp.edu.ar/handle/10915/122052

[5] Cortez, P. and Silva, A. (2008). Using Data Mining to Predict Secondary school Student Performance. In A. Brito and J. Teixeira (Eds.) *Proceeding of 5th Futu-*

re Business Technology Conference (FUBUTEC 2008. pp. 5-12. Porto, Portugal, EUROSIS, ISBN 978-9077381-39-7.

[6] Del Brío, B. M. and Sanz, A. (2006). Redes Neuronales y Sistemas Borrosos. *Ra-Ma*.

[7] Delahoz, E. (2020). Data of academic performance evolution for engineering students. *Mendeley Data*, V1. DOI: 10.17632/83tcx8psxv.1

[8] Delahoz-Dominguez, E., ZUluaga, R. and Fontalvo-Herrera, T. (2020). Dataser of academic perfonrmance evolution for engineering students. *Data in Brief*, 30, 105537. DOI: https://doi.org/10.1016/j.dib.2020.105537

[9] Del Campo-Ávila, J., Ramos-Jiménez, G., Morales-Bueno, R. and Baena-García, M. (2017). *Minería de datos educativos para la predicción personalizada del rendimiento académico.* [Informe práctico, Universidad de Málaga].

[10] Hertz, J., Krogh, a., and Palmer, R. (1991).*Introduction To The theory Of Neural Computation*. Santa Fe Institute Series: Westview Press.

[11] Koedinger, K. R., Cunningham, K., Skogsholm, A., and Leber, B. (2008). An open repository and analysis tools for fine-grained, longitudinal learner data. *Proceedings of the 1st International Conference on Educational Data Mining*. Obtenido de https://www.researchgate.net/publication/221570519_An_Open_Repository_and_analysis_tools_for_fine-grained_longitudinal_learner_data

[12] Landis, J.R., and Koch, G.G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 159-174.

[13] López Zambrano, J. (2021). *Modelos genéricos para la predicción de las notas finales en cursos a partir de la información de interacción de los estudiantes con el sistema Moodle*. Córdoba: [Tesis de doctorado, Universidad de Córdoba].

[14] López-Zambrano, J., Lara, J.A. and Romero, C. (2021). Early Prediction of Student Learning Performance Through Data Mining: A Systematic Review. *Psicothema*, 33(3), 456-465. DOI: 10.7334/psicothema2021.62

[15] Márquez, C. (2015). *Predicción del Fracaso y el abandono escolar mediante técnicas de minería de datos*. Córdoba: [Tesis de doctorado, Universidad de Córdoba].

[16] Matas Terrón, A., Leiva Olivencia, J. J., and Franco Caballero, P. D. (2020). Big Data Irruption in Education. *Pixel-BIT*. DOI:https://doi.org/10.12795/pixelbit.2020.i57.02

[17] Menacho, C.H. (2017). Predicción del rendimiento académico aplicando técnicas de minería de datos. *Anales Científicos*, 78(1), 26-33. DOI: https://dx.doi.org/10.21704/ac.v78i1.811.

[18] Müller, B., Reinhardt, J., and Strickland, M. T. (1995). *Neural networks: an introduction*. Springer.

[19] Papamitsiou, Z.K., and Economides, A.A. (2014). Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *Educational Technology and Society*, 17(4), 49-64.

[20] Pedregosa *et. al.*, Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

[21] Pérez-Suasnavas, A.-L., Hasperué, W., Molina, M. E., and Santamaría C., J. (2022). Predicción de dificultades estudiantiles mediante técnicas de minería de textos. *South Florida Journal of Development*. DOI:10.46932/sfjdv3n5-061

[22] Quinlan, J.R., (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

[23] Rasberry, C. N., Lee, S.M., Robin, L., Laris, B. A., Russell, L. A., Coyle, K. K., and Nihiser, A. J. (2011). The association between school-based physical activity, including physical education, and academic performance: a systematic review of the literature. *Preventive medicine*, 52, S10-S20.

[24] Valero Orea, S., Salvador Vargas, A., and García Alonso, M. (s.f.). Minería de datos: predicción de la deserción escolar mediante el algoritmo de árboles de decisión y el algoritmo de los k vecinos más cercanos. *Academia*.

[25] Yamao, E. (2018). *Predicción del rendimiento académico mediante minería de datos en estudiantes del primer ciclo de la escuela profesional de ingeniería de computación y sistemas*. [Tesis de doctorado, Universidad de San Martín de Porres]. Repositorio Académico USMP.

[26] York, T. T., Gibson, C., and Rankin, S. (2015). Defining and measuring academic success. *Practical Assessment, Research and Evaluation*, 20(5). DOI: https://doi.org/10.7275/hz5x-tx03

Anexos

A. Variables

En esta sección se expondrán las variables de los datasets y los cambios pertinentes a numérico.

Tabla VII: *Variables del dataset de Portugal con sus respectivos cambios.*

Variable	Valor en base de datos	Valor utilizado
School	- GP (Gabriel Pereira) - MS (Mousinho da Silveira)	- 0 - 1
Sex	-F -M	- 0 - 1
Age	Valor numérico	-
Address	- U (Urbano) - R (Rural)	- 0 - 1
Tamaño de la familia (famsize)	- LE3 (menor que 3) - GT3 (más que 3)	- 0 - 1
Cohabitación de los padres (Pstatuts)	- T (juntos) - A (separados)	- 0 - 1
Educación de la madre/padre (Medu/Fedu)	- 0: ninguna - 1: primaria hasta 4º - 2 de 5º a 9º - 3 educación secundaria - 4 educación superior	-
Trabajo de la madre/padre (Mjob/Fjob)	- profesor - sanidad - servicios civiles - En casa - Otro	- 0 - 1 - 2 - 3 - 4
Reason	- Home - Reputation - Course - Other	- 0 - 1 - 2 - 3

Tutor del estudiante (guardian)	- Mother - Father - Other	- 0 - 1 - 2
Traveltime	- 1 (<15min) - 2 (15-30min) - 3 (30min-1h) - 4 (>1h)	-
Studytime	- 1 (<2h) - 2 (2-5h) - 3 (5-10h) - 4 (>10h)	-
Failures	- n con n si está entre 0 y 3 - 4 si son más	-
Clases de refuerzo en escuela (schoolsup) **Refuerzo con la familia (famsup)** **Clases de refuerzo pagadas (paid)** **Actividades extraescolares (activities)** **Nursery** **Educación superior (Higher)** **Internet en casa** **relación romántica**	- Si - No	- 1 - 0
Relación familiar (famrel)	De 1 (muy mal) hasta 5 (excelente)	-
Tiempo libre post-escuela (freetime) **Salidas con los amigos (goout)** **Alcohol entre semana (Dalc)** **Alcohol en fin de semana (Walc)**	De 1 (muy poco) hasta 5 (mucho)	-
Salud (Health)	De 1 (muy mala) hasta 5 (muy buena)	-

Faltas a la escuela	Nº de veces que se ha faltado	-
G1	Nota primer trimestre (0-20)	-
G2	Nota segundo trimestre (0-20)	-
G3	Nota final de curso (0-20)	- 0 (suspenso) - 1 (aprobado)
		- 0 (suspenso) - 1 (suficiente) - 2 (bien) - 3 (notable) - 4 (sobresaliente)

Tabla VIII: *Variables del dataset de Bolivia con sus respectivos cambios*

Variable	Valor en base de datos	Valor utilizado
GENDER	- M - F	- 0 - 1
EDU_FATHER **EDU_MOTHER**	- Ninguno - Complete/incomplete primary -Complete/incomplete professional education - Complete/incomplete secundary - Complete/incomplete technique or technology - Not sure - Postgraduate education	- 0 - 1/-1 - 2/-2 - 3/-3 - 4/-4 - 5 - 6
OCC_FATHER **OCC_MOTHER**	-Auxiliary or Administrative - Entrepreneur - Executive - Home - Independent - Independent profesional - Operator - Small entrepreneur - Technical or professional level employee - Other occupation - Retired	- 1 - 2 - 3 - 4 - 5 - 6 -7 - 8 - 9 -10 -11
SISBEN **(clasificación de nivel económico)**	- Está clasificada en otro level del SISBEN - It is not classified by the SISBEN - Level 1 - Level 2 - Level 3	- -1 - 0 - 1 - 2 - 3
PEOPLE_HOUSE	Del 1 al 12 (12 o más)	-

INTERNET **TV** **COMPUTER** **WASHING_MCH** **MIC_OVEN** **CAR** **DVD** **FRESH** **PHONE** **MOBILE**	- Yes - No	- 1 - 0
REVENUE **(ingresos)**	- Less than 1 LMMW - Between 1 and less than 2 LMMW - Between 2 and less than 3 LMMW - Between 3 and less than 5 LMMW - Between 5 and less than 7 LMMW - Between 7 and less than 10 LMMW - 10 or more LMMW	- 1 - 2 - 3 - 4 - 5 - 6 - 7
JOB	- No - Yes, less than 20 h/week - Yess, 20h or more/week	- 0 - 1 - 2
SCHOOL_NAT	- PRIVATE -PUBLIC	- 0 - 1
SCHOOL_TYPE	- ACADEMIC - TECHNICAL - TECNICAL/ACADEMIC - Not apply	- 1 - 2 - 3 - 0
MAT_S11 **CR_S11** **CC_S11** **BIO_S11** **ENG_S11**	Notas saber 11 (0-100)	-

ACADEMIC_PROGRAM	AERONAUTICAL ENGINEERING - AUTOMOTION ENGINEERING - CATASTRAL ENGINEERING AND GEODESY - CHEMICAL ENGINEERING - CIVIL CONSTRUCTIONS - CIVIL ENGINEERING - CONTROL ENGINEERING - ELECTRIC ENGINEERING - ELECTRIC ENGINEERING AND TELECOMMUNICATIONS - ELECTROMECHANICAL ENGINEERING - ELECTRONIC ENGINEERING - INDUSTRIAL CONTROL AND AUTOMATION ENGINEERING - INDUSTRIAL ENGINEERING - MECHANICAL ENGINEERING - MECHATRONICS ENGINEERING - PRODUCTION ENGINEERING - PRODUCTIVITY AND QUALITY ENGINEERING - TEXTILE ENGINEERING - TOPOGRAPHIC ENGINEERY - TRASPORTATION AND ROAD ENGINEERING	Números del 1 al 21 en ese orden

QR_PRO **CR_PRO** **CC_PRO** **ENG_PRO** **WC_PRO**	Notras correspondientes al examen SABER PRO (0-100)	-
FEP_PRO	Notas proyectos de ingeniería (0-300)	-
SEL **SEL_IHE**	Nivel socioeconómico Nivel socioeconómico de la institución de educación superior	Del 1 al 4
G_SC	Puntuación global 0-300	-
PERCENTILE	1-100	Agrupación de 10 (0-9)

B. Código empleado

```
# -*- coding: utf-8 -*-
"""
Created on Thu May  2 10:25:27 2024

@author: enriq
"""

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import roc_auc_score
from sklearn.metrics import cohen_kappa_score
#
    ================================================================================

# BASES DE DATOS DE PORTUGAL
#
    ================================================================================


base = 'datasets/Portugal/student-por-numeric2.csv'

#
    ================================================================================

# CÓDIGO
#
    ================================================================================

Port = pd.read_csv(base, sep=";")

X = Port.values[:,:32]
# Y = Port["Final"].values #binario
Y = Port["Final2"].values #5 valores
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y)

# textoArbol = open("Portugalarbolbinario", 'w')
# textoADA = open("PortugalADAbinario", 'w')
# textoNN = open("PortugalNNbinario", 'w')
# textoArbol.write("Variable a predecir: aprobado (1), suspenso (0)\n\n")
# textoADA.write("Variable a predecir: aprobado (1), suspenso (0)\n\n")
# textoNN.write("Variable a predecir: aprobado (1), suspenso (0)\n\n")

textoArbol = open("Portugalarbol5valores", 'w')
textoADA = open("PortugalADA5valores", 'w')
textoNN = open("PortugalNN5valores", 'w')
textoArbol.write("Variable a predecir: suspenso (0), suficiente (1), bien
    (2), notable (3), sobresaliente (4)\n\n")
textoADA.write("Variable a predecir: suspenso (0), suficiente (1), bien (2)
    , notable (3), sobresaliente (4)\n\n")
textoNN.write("Variable a predecir: suspenso (0), suficiente (1), bien (2),
     notable (3), sobresaliente (4)\n\n")
#
    ================================================================


#             ÁRBOL DE DECISIÓN
#
    ================================================================


Arbol = DecisionTreeClassifier(criterion='entropy', max_depth=3)
Arbol.fit(X_train, Y_train)
Y_predict_Arbol = Arbol.predict(X_test)
Matriz_Arbol = confusion_matrix(Y_test, Y_predict_Arbol)
accArbol = accuracy_score(Y_test, Y_predict_Arbol)
if len(Matriz_Arbol[0,]) == 2:
    TArate = Matriz_Arbol[1,1]/(Matriz_Arbol[1,1] + Matriz_Arbol[1,0])
    TSrate = Matriz_Arbol[0,0]/(Matriz_Arbol[0,0] + Matriz_Arbol[0,1])
    GM = np.sqrt(TArate*TSrate)
    AUC = roc_auc_score(Y_test, Y_predict_Arbol)
else:
    Trate = np.empty(len(Matriz_Arbol))
    for i in range(len(Trate)):
        N = 0
        for j in range(len(Trate)):
            N += Matriz_Arbol[i,j]
        Trate[i] = Matriz_Arbol[i,i]/N
```

```
    M = 1
    for i in range(len(Trate)):
        if Trate[i] != 0:
            M *= Trate[i]
    GM = np.sqrt(M)
    # AUC = roc_auc_score(Y_test, Y_predict_Arbol, multi_class='ovr')
kappa = cohen_kappa_score(Y_test,Y_predict_Arbol)
print("Matriz de confusión Arbol de decisión\n", Matriz_Arbol)
print("Precisión Arbol: ", accArbol)
textoArbol.write("Matriz de confusión\n\n" + str(Matriz_Arbol) + "\n\n")

if len(Matriz_Arbol[0,]) > 2:
    for i in range(len(Matriz_Arbol[0,])):
        for j in range(len(Matriz_Arbol[0,])):
            if i == j:
                textoArbol.write("T" + str(i) + str(j) + " = " + str(
    Matriz_Arbol[i,j]) + "\n")
            else:
                textoArbol.write("F" + str(i) + str(j) + " = " + str(
    Matriz_Arbol[i,j]) + "\n")
    textoArbol.write("\n")
else:
    textoArbol.write("TS = " + str(Matriz_Arbol[0,0]) + "\nTA = " + str(
    Matriz_Arbol[1,1]) + "\nFS = " + str(Matriz_Arbol[1,0])
                     + "\nFA = " + str(Matriz_Arbol[0,1]) + "\n\n")
textoArbol.write("Precisión = " + str(accArbol) + "\n")

if len(Matriz_Arbol) == 2:
    textoArbol.write("Tasa Verdaderos Aprobados = " + str(TArate) + "\n")
    textoArbol.write("Tasa Verdaderos Suspensos = " + str(TSrate) + "\n")
else:
    for i in range(len(Trate)):
        textoArbol.write("Tasa Verdadero" + str(i) + " = " + str(Trate[i])
    + "\n")

textoArbol.write("Media Geométrica =" + str(GM) + "\n")
# textoArbol.write("Área bajo la curva ROC = " + str(AUC) + "\n")
textoArbol.write("Coeficiente de cohen kappa = " + str(kappa) + "\n")
textoArbol.close()
#
    ================================================================


#           ADABOOST
```

```
#
    ================================================================================


ADA = AdaBoostClassifier(DecisionTreeClassifier(max_depth=3, random_state
    =0), n_estimators=int(10000), learning_rate=0.0001) #consultar
    referencias
ADA.fit(X_train, Y_train)
Y_predict_ADA = ADA.predict(X_test)
Matriz_ADA = confusion_matrix(Y_test, Y_predict_ADA)
accADA = accuracy_score(Y_test, Y_predict_ADA)
if len(Matriz_ADA[0,]) == 2:
    TArate = Matriz_ADA[1,1]/(Matriz_ADA[1,1] + Matriz_ADA[1,0])
    TSrate = Matriz_ADA[0,0]/(Matriz_ADA[0,0] + Matriz_ADA[0,1])
    GM = np.sqrt(TArate*TSrate)
    AUC = roc_auc_score(Y_test, Y_predict_ADA)
else:
    Trate = np.empty(len(Matriz_ADA))
    for i in range(len(Trate)):
        N = 0
        for j in range(len(Trate)):
            N += Matriz_ADA[i,j]
        Trate[i] = Matriz_ADA[i,i]/N
    M = 1
    for i in range(len(Trate)):
        if Trate[i] != 0:
            M *= Trate[i]
    GM = np.sqrt(M)
    # AUC = roc_auc_score(Y_test, Y_predict_ADA, average = "macro",
    multi_class="ovr")
kappa = cohen_kappa_score(Y_test,Y_predict_ADA)
print("Matriz de confusión ADABoost\n", Matriz_ADA)
print("Precisión ADABoost: ", accADA)
textoADA.write("Matriz de confusión\n\n" + str(Matriz_ADA) + "\n\n")
if len(Matriz_ADA[0,]) > 2:
    for i in range(len(Matriz_ADA[0,])):
        for j in range(len(Matriz_ADA[0,])):
            if i == j:
                textoADA.write("T" + str(i) + str(j) + " = " + str(
    Matriz_ADA[i,j]) + "\n")
            else:
                textoADA.write("F" + str(i) + str(j) + " = " + str(
    Matriz_ADA[i,j]) + "\n")
```

```
    textoADA.write("\n")
else:
    textoADA.write("TS = " + str(Matriz_ADA[0,0]) + "\nTA = " + str(
    Matriz_ADA[1,1]) + "\nFS = " + str(Matriz_ADA[1,0])
                    + "\nFA = " + str(Matriz_ADA[0,1]) + "\n\n")

textoADA.write("Precisión = " + str(accADA) + "\n")

if len(Matriz_ADA) == 2:
    textoADA.write("Tasa Verdaderos Aprobados = " + str(TArate) + "\n")
    textoADA.write("Tasa Verdaderos Suspensos = " + str(TSrate) + "\n")
else:
    for i in range(len(Trate)):
        textoADA.write("Tasa Verdadero" + str(i) + " = " + str(Trate[i]) +
    "\n")

textoADA.write("Media Geométrica = " + str(GM) + "\n")
# textoADA.write("Área bajo la curva ROC = " + str(AUC) + "\n")
textoADA.write("Coeficiente de cohen kappa = " + str(kappa) + "\n")
textoADA.close()
#
    ================================================================


#            REDES NEURONALES
#
    ================================================================



NN = MLPClassifier(hidden_layer_sizes=(9,), activation = "logistic",
    solver = "lbfgs", alpha = 0.1, max_iter = int(10000), learning_rate='
    constant', learning_rate_init=0.0001)
#consultar referencias para parámetros
NN.fit(X_train, Y_train)
Y_predict_NN = NN.predict(X_test)
Matriz_NN = confusion_matrix(Y_test, Y_predict_NN)
accNN = accuracy_score(Y_test, Y_predict_NN)
if len(Matriz_NN[0,]) == 2:
    TArate = Matriz_NN[1,1]/(Matriz_NN[1,1] + Matriz_NN[1,0])
    TSrate = Matriz_NN[0,0]/(Matriz_NN[0,0] + Matriz_NN[0,1])
    GM = np.sqrt(TArate*TSrate)
    AUC = roc_auc_score(Y_test, Y_predict_NN)
else:
    Trate = np.empty(len(Matriz_NN))
```

```
    for i in range(len(Trate)):
        N = 0
        for j in range(len(Trate)):
            N += Matriz_NN[i,j]
        Trate[i] = Matriz_NN[i,i]/N
    M = 1
    for i in range(len(Trate)):
        if Trate[i] != 0:
            M *= Trate[i]
    GM = np.sqrt(M)
    # AUC = roc_auc_score(Y_test, Y_predict_NN, average = "macro",
    multi_class="ovr")
kappa = cohen_kappa_score(Y_test,Y_predict_NN)
print("Matriz de confusión Red neuronal\n", Matriz_NN)
print("Precisión Red Neuronal: ", accNN)
textoNN.write("Matriz de confusión\n\n" + str(Matriz_NN) + "\n\n")
if len(Matriz_NN[0,]) > 2:
    for i in range(len(Matriz_NN[0,])):
        for j in range(len(Matriz_NN[0,])):
            if i == j:
                textoNN.write("T" + str(i) + str(j) + " = " + str(Matriz_NN
    [i,j]) + "\n")
            else:
                textoNN.write("F" + str(i) + str(j) + " = " + str(Matriz_NN
    [i,j]) + "\n")
    textoNN.write("\n")
else:
    textoNN.write("TS = " + str(Matriz_NN[0,0]) + "\nTA = " + str(Matriz_NN
    [1,1]) + "\nFS = " + str(Matriz_NN[1,0])
                    + "\nFA = " + str(Matriz_NN[0,1]) + "\n\n")
textoNN.write("Precisión = " + str(accNN) + "\n")
if len(Matriz_NN) == 2:
    textoNN.write("Tasa Verdaderos Aprobados = " + str(TArate) + "\n")
    textoNN.write("Tasa Verdaderos Suspensos = " + str(TSrate) + "\n")
else:
    for i in range(len(Trate)):
        textoNN.write("Tasa Verdadero" + str(i) + " = " + str(Trate[i]) + "
    \n")

textoNN.write("Media Geométrica = " + str(GM) + "\n")
# textoNN.write("Área bajo la curva ROC = " + str(AUC) + "\n")
textoNN.write("Coeficiente de cohen kappa = " + str(kappa) + "\n")
textoNN.close()
```

```
#
    ================================================================

# Representación de las matrices de confusión
#
    ================================================================

if len(Matriz_Arbol) == 2:
    fig, ax = plt.subplots()
    ax.imshow(Matriz_Arbol, cmap = 'turbo_r')
    ticks = ["supensos", "aprobados"]
    ax.set_xticks([0,1], labels = ticks, fontsize = 13)
    ax.set_yticks([0,1], labels = ticks, fontsize = 13)
    plt.setp(ax.get_yticklabels(), rotation=45, ha="right", rotation_mode="
    anchor")
    ax.set_ylabel("Variables reales", fontsize = 15)
    ax.set_xlabel("Variables predichas", fontsize = 15)
    ax.set_title("Matriz de confusión del Árbol de decisión", fontsize =
    15)
    for i in range(len(Matriz_Arbol[0,])):
        for j in range(len(Matriz_Arbol[0,])):
            ax.text(j,i,Matriz_Arbol[i,j],ha="center", va="center", color="
    w", fontsize = 17)
    plt.savefig("MatrizConfusiónBinarioArbol.png",bbox_inches = "tight")
    plt.show()


    fig, ax = plt.subplots()
    ax.imshow(Matriz_ADA, cmap = 'turbo_r')
    ticks = ["supensos", "aprobados"]
    ax.set_xticks([0,1], labels = ticks, fontsize = 13)
    ax.set_yticks([0,1], labels = ticks, fontsize = 13)
    plt.setp(ax.get_yticklabels(), rotation=45, ha="right", rotation_mode="
    anchor")
    ax.set_ylabel("Variables reales", fontsize = 15)
    ax.set_xlabel("Variables predichas", fontsize = 15)
    ax.set_title("Matriz de confusión del ADABoost", fontsize = 15)
    for i in range(len(Matriz_ADA[0,])):
        for j in range(len(Matriz_ADA[0,])):
            ax.text(j,i,Matriz_ADA[i,j],ha="center", va="center", color="w"
    , fontsize = 17)
    plt.savefig("MatrizConfusiónBinarioADA.png",bbox_inches = "tight")
    plt.show()
```

```
    fig, ax = plt.subplots()
    ax.imshow(Matriz_NN, cmap = 'turbo_r')
    ticks = ["supensos", "aprobados"]
    ax.set_xticks([0,1], labels = ticks, fontsize = 13)
    ax.set_yticks([0,1], labels = ticks, fontsize = 13)
    plt.setp(ax.get_yticklabels(), rotation=45, ha="right", rotation_mode="
   anchor")
    ax.set_ylabel("Variables reales", fontsize = 15)
    ax.set_xlabel("Variables predichas", fontsize = 15)
    ax.set_title("Matriz de confusión de la Red Neuronal", fontsize = 15)
    for i in range(len(Matriz_NN[0,])):
        for j in range(len(Matriz_NN[0,])):
            ax.text(j,i,Matriz_NN[i,j],ha="center", va="center", color="w",
    fontsize = 17)
    plt.savefig("MatrizConfusiónBinarioNN.png",bbox_inches = "tight")
    plt.show()
else:
    fig, ax = plt.subplots()
    ax.imshow(Matriz_Arbol, cmap = 'turbo_r')
    ticks = ["supenso", "suficiente", "bien", "notable", "sobresaliente"]
    ax.set_xticks([0,1,2,3,4], labels = ticks, fontsize = 13)
    ax.set_yticks([0,1,2,3,4], labels = ticks, fontsize = 13)
    plt.setp(ax.get_yticklabels(), rotation=45, ha="right", rotation_mode="
   anchor")
    plt.setp(ax.get_xticklabels(), rotation=45, ha="right", rotation_mode="
   anchor")
    ax.set_ylabel("Variables reales", fontsize = 15)
    ax.set_xlabel("Variables predichas", fontsize = 15)
    ax.set_title("Matriz de confusión del Árbol de decisión", fontsize =
   15)
    for i in range(len(Matriz_Arbol[0,])):
        for j in range(len(Matriz_Arbol[0,])):
            ax.text(j,i,Matriz_Arbol[i,j],ha="center", va="center", color="
   w", fontsize = 17)
    plt.savefig("MatrizConfusión5valoresArbol.png", bbox_inches = "tight")
    plt.show()

    fig, ax = plt.subplots()
    ax.imshow(Matriz_ADA, cmap = 'turbo_r')
    ticks = ["supenso", "suficiente", "bien", "notable", "sobresaliente"]
    ax.set_xticks([0,1,2,3,4], labels = ticks, fontsize = 13)
    ax.set_yticks([0,1,2,3,4], labels = ticks, fontsize = 13)
```

```
plt.setp(ax.get_yticklabels(), rotation=45, ha="right", rotation_mode="
anchor")
plt.setp(ax.get_xticklabels(), rotation=45, ha="right", rotation_mode="
anchor")
ax.set_ylabel("Variables reales", fontsize = 15)
ax.set_xlabel("Variables predichas", fontsize = 15)
ax.set_title("Matriz de confusión del ADABoost", fontsize = 15)
for i in range(len(Matriz_ADA[0,])):
    for j in range(len(Matriz_ADA[0,])):
        ax.text(j,i,Matriz_ADA[i,j],ha="center", va="center", color="w"
, fontsize = 17)
plt.savefig("MatrizConfusión5valoresADA.png", bbox_inches = "tight")
plt.show()

fig, ax = plt.subplots()
ax.imshow(Matriz_NN, cmap = 'turbo_r')
ticks = ["supenso", "suficiente", "bien", "notable", "sobresaliente"]
ax.set_xticks([0,1,2,3,4], labels = ticks, fontsize = 13)
ax.set_yticks([0,1,2,3,4], labels = ticks, fontsize = 13)
plt.setp(ax.get_yticklabels(), rotation=45, ha="right", rotation_mode="
anchor")
plt.setp(ax.get_xticklabels(), rotation=45, ha="right", rotation_mode="
anchor")
ax.set_ylabel("Variables reales", fontsize = 15)
ax.set_xlabel("Variables predichas", fontsize = 15)
ax.set_title("Matriz de confusión de la Red Neuronal", fontsize = 15)
for i in range(len(Matriz_NN[0,])):
    for j in range(len(Matriz_NN[0,])):
        ax.text(j,i,Matriz_NN[i,j],ha="center", va="center", color="w",
 fontsize = 17)
plt.savefig("MatrizConfusión5valoresNN.png",bbox_inches = "tight")
plt.show()
```

Código para la predicción del dataset de Portugal

```
# -*- coding: utf-8 -*-
"""
Created on Thu May 16 10:55:40 2024

@author: enriq
"""

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import roc_auc_score
from sklearn.metrics import cohen_kappa_score

#
    ================================================================================

# BASE DE DATOS
#
    ================================================================================

base = "datasets/Bolivia/data_academic_performance.xlsx"

#SE LEEN LOS DATOS DEL EXEL
df = pd.read_excel(base)

#SE PROCEDE A CAMBIAR ALGUNOS DATOS A NUMÉRICOS
lista_educa = ["Ninguno", "Complete primary ", "Complete professional
    education", "Complete Secundary", "Complete technique or technology",
                "Incomplete primary ", "Incomplete Professional Education",
    "Incomplete Secundary", "Incomplete technical or technological",
                "Not sure", "Postgraduate education"]
lista_ocup = ["Auxiliary or Administrative", "Entrepreneur ", "Executive",
    "Home", "Independent", "Independent professional",
              "Operator","Small entrepreneur ", "Technical or professional
    level employee", "Other occupation","Retired"]
sisben = ["Esta clasificada en otro Level del SISBEN", "It is not
```

```
    classified by the SISBEN", "Level 1", "Level 2", "Level 3"]
people = ["One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "
    Nueve", "Ten", "Once", "Twelve or more"]
ingresos = ["less than 1 LMMW", "Between 1 and less than 2 LMMW", "Between
    2 and less than 3 LMMW", "Between 3 and less than 5 LMMW",
            "Between 5 and less than 7 LMMW", "Between 7 and less than 10
   LMMW", "10 or more LMMW"]
trabajo = ["No", "Yes, less than 20 hours per week", "Yes, 20 hours or more
     per week"]
grados = ["AERONAUTICAL ENGINEERING", "AUTOMATION ENGINEERING", "CATASTRAL
    ENGINEERING AND GEODESY", "CHEMICAL ENGINEERING",
            "CIVIL CONSTRUCTIONS", "CIVIL ENGINEERING", "CONTROL ENGINEERING"
    , "ELECTRIC ENGINEERING",
            "ELECTRIC ENGINEERING AND TELECOMMUNICATIONS", "ELECTROMECHANICAL
     ENGINEERING", "ELECTRONIC ENGINEERING",
            "INDUSTRIAL AUTOMATIC ENGINEERING", "INDUSTRIAL CONTROL AND
    AUTOMATION ENGINEERING", "INDUSTRIAL ENGINEERING",
            "MECHANICAL ENGINEERING", "MECHATRONICS ENGINEERING", "PRODUCTION
     ENGINEERING", "PRODUCTIVITY AND QUALITY ENGINEERING",
            "TEXTILE ENGINEERING", "TOPOGRAPHIC ENGINEERY", "TRANSPORTATION
    AND ROAD ENGINEERING"]
sino = ["INTERNET","TV","COMPUTER", "WASHING_MCH","MIC_OVEN","CAR", "DVD","
    FRESH", "PHONE", "MOBILE"]
matriz = np.zeros((10,10))
count = 1
for i in range(10):
    for j in range(10):
        matriz[i,j] = count
        count += 1

#ELIMINAMOS LAS VARIABLES QUE NO DESEAMOS
df = df.drop(['COD_S11', 'STRATUM', 'SCHOOL_NAME', 'Cod_SPro', 'UNIVERSITY'
    , '2ND_DECILE', 'QUARTILE'], axis = 1)

#CAMBIAMOS AQUELLAS QUE ESTÁN EN FORMATO TEXTO
df["GENDER"] = df["GENDER"].replace(["M","F"],[0,1])
df[sino] = df[sino].replace(["Yes", "No"], [1,0])
df[["EDU_FATHER", "EDU_MOTHER"]] = df[["EDU_FATHER", "EDU_MOTHER"]].replace
    (lista_educa, [0,1,2,3,4,-1,-2,-3,-4,5,6])
df[["OCC_FATHER", "OCC_MOTHER"]] = df[["OCC_FATHER", "OCC_MOTHER"]].replace
    (lista_ocup, list(np.arange(1,len(lista_ocup)+1, 1)))
df["SISBEN"] = df["SISBEN"].replace(sisben,[-1,0,1,2,3])
df["PEOPLE_HOUSE"] = df["PEOPLE_HOUSE"].replace(people, list(np.arange
```

```
    (1,13,1)))
df["REVENUE"] = df["REVENUE"].replace(ingresos, list(np.arange(1,len(
    ingresos) + 1, 1)))
df["JOB"] = df["JOB"].replace(trabajo, [0,1,2])
df["SCHOOL_NAT"] = df["SCHOOL_NAT"].replace(["PRIVATE", "PUBLIC"], [0,1])
df["SCHOOL_TYPE"] = df["SCHOOL_TYPE"].replace(["ACADEMIC", "TECHNICAL", "
    TECHNICAL/ACADEMIC", "Not apply"], [1,2,3,0])
df["ACADEMIC_PROGRAM"] = df["ACADEMIC_PROGRAM"].replace(grados, list(np.
    arange(1,len(grados)+1,1)))
for i in range(10):
    df["PERCENTILE"] = df["PERCENTILE"].replace(matriz[i,:],i)
#
    ==========================================================================

# CODIGO INICIAL
#
    ==========================================================================

X = df.values[:,:36]
Y = df["PERCENTILE"].values

X_train, X_test, Y_train, Y_test = train_test_split(X,Y)

textoArbol = open("Boliviaarbolpercentiles", 'w')
textoADA = open("BoliviaADApercentiles", 'w')
textoNN = open("BoliviaNNpercetiles", 'w')
textoArbol.write("Variable a predecir: percentiles del 1 al 100 (divididos
    en 10 rangos)\n\n")
textoADA.write("Variable a predecir: percentiles del 1 al 100 (divididos en
    10 rangos)\n\n")
textoNN.write("Variable a predecir: percentiles del 1 al 100 (divididos en
    10 rangos)\n\n")

#
    ==========================================================================

# ÁRBOL DE DECISIÓN
#
    ==========================================================================


Arbol = DecisionTreeClassifier(criterion='entropy', max_depth=3)
Arbol.fit(X_train, Y_train)
```

```
Y_predict_Arbol = Arbol.predict(X_test)
ACC_ARBOL = accuracy_score(Y_test, Y_predict_Arbol)
kappa = cohen_kappa_score(Y_test,Y_predict_Arbol)
Matriz_Arbol = confusion_matrix(Y_test, Y_predict_Arbol)

Trate = np.empty(len(Matriz_Arbol))
for i in range(len(Trate)):
    N = 0
    for j in range(len(Trate)):
        N += Matriz_Arbol[i,j]
    Trate[i] = Matriz_Arbol[i,i]/N
M = 1
for i in range(len(Trate)):
    if Trate[i] != 0:
        M *= Trate[i]
GM = np.sqrt(M)

print("Matriz de confusión Arbol de decisión\n", Matriz_Arbol)
print("Precisión Arbol = ", ACC_ARBOL)
print("Coeficiente kappa Arbol = ", kappa)
print("Media geométrica Arbol = ", GM)

textoArbol.write("Matriz de confusión\n\n" + str(Matriz_Arbol) + "\n\n")
textoArbol.write("Precisión = " + str(ACC_ARBOL) + "\n")
textoArbol.write("Media Geométrica =" + str(GM) + "\n")
textoArbol.write("Coeficiente de cohen kappa = " + str(kappa) + "\n")
textoArbol.close()
#
    ================================================================================

# ADABOOST
#
    ================================================================================


ADA = AdaBoostClassifier(DecisionTreeClassifier(max_depth=3, random_state
    =0), n_estimators=int(2000), learning_rate=0.0001) #consultar
    referencias
ADA.fit(X_train, Y_train)
Y_predict_ADA = ADA.predict(X_test)
Matriz_ADA = confusion_matrix(Y_test, Y_predict_ADA)
ACC_ADA = accuracy_score(Y_test, Y_predict_ADA)
kappa = cohen_kappa_score(Y_test,Y_predict_ADA)
```

```
Trate = np.empty(len(Matriz_ADA))
for i in range(len(Trate)):
    N = 0
    for j in range(len(Trate)):
        N += Matriz_ADA[i,j]
    Trate[i] = Matriz_ADA[i,i]/N
M = 1
for i in range(len(Trate)):
    if Trate[i] != 0:
        M *= Trate[i]
GM = np.sqrt(M)

print("Matriz de confusión AdaBoost\n", Matriz_ADA)
print("Precisión AdaBoost = ", ACC_ADA)
print("Coeficiente kappa AdaBoost = ", kappa)
print("Media geométrica AdaBoost = ", GM)

textoADA.write("Matriz de confusión\n\n" + str(Matriz_ADA) + "\n\n")
textoADA.write("Precisión = " + str(ACC_ADA) + "\n")
textoADA.write("Media Geométrica =" + str(GM) + "\n")
textoADA.write("Coeficiente de cohen kappa = " + str(kappa) + "\n")
textoADA.close()
#
    ================================================================================


# RED NEURONAL
#
    ================================================================================



NN = MLPClassifier(hidden_layer_sizes= (13,), activation = "logistic",
    solver = "lbfgs", alpha = 0.1, max_iter = int(10000), learning_rate='
    constant', learning_rate_init=0.001)
#consultar referencias para parámetros
NN.fit(X_train, Y_train)
Y_predict_NN = NN.predict(X_test)
Matriz_NN = confusion_matrix(Y_test, Y_predict_NN)
ACC_NN = accuracy_score(Y_test, Y_predict_NN)
kappa = cohen_kappa_score(Y_test,Y_predict_NN)

Trate = np.empty(len(Matriz_NN))
for i in range(len(Trate)):
```

```
    N = 0
    for j in range(len(Trate)):
        N += Matriz_NN[i,j]
    Trate[i] = Matriz_NN[i,i]/N
M = 1
for i in range(len(Trate)):
    if Trate[i] != 0:
        M *= Trate[i]
GM = np.sqrt(M)

print("Matriz de confusión Redes Neuronales\n", Matriz_NN)
print("Precisión Redes Neuronales = ", ACC_NN)
print("Coeficiente kappa Redes Neuronales = ", kappa)
print("Media geométrica Redes Neuronales = ", GM)

textoNN.write("Matriz de confusión\n\n" + str(Matriz_NN) + "\n\n")
textoNN.write("Precisión = " + str(ACC_NN) + "\n")
textoNN.write("Media Geométrica =" + str(GM) + "\n")
textoNN.write("Coeficiente de cohen kappa = " + str(kappa) + "\n")
textoNN.close()

#
    ================================================================

# GRÁFICAS MATRICES
#
    ================================================================



fig, ax = plt.subplots()
ax.imshow(Matriz_Arbol, cmap = 'turbo_r')
ticks = ["0-10%", "10-20%", "20-30%","30-40%","40-50%","50-60%","60-70%","
    70-80%","80-90%","90-100%"]
ax.set_xticks(list(np.arange(0,10,1)), labels = ticks, fontsize = 13)
ax.set_yticks(list(np.arange(0,10,1)), labels = ticks, fontsize = 13)
plt.setp(ax.get_yticklabels(), rotation=45, ha="right", rotation_mode="
    anchor")
plt.setp(ax.get_xticklabels(), rotation=45, ha="right", rotation_mode="
    anchor")
ax.set_ylabel("Variables reales", fontsize = 15)
ax.set_xlabel("Variables predichas", fontsize = 15)
ax.set_title("Matriz de confusión del Árbol de decisión", fontsize = 15)
for i in range(len(Matriz_Arbol[0,])):
```

```
    for j in range(len(Matriz_Arbol[0,])):
        ax.text(j,i,Matriz_Arbol[i,j],ha="center", va="center", color="w",
    fontsize = 13)
plt.savefig("MatrizConfusiónBoliviaArbol.png", bbox_inches = "tight")
plt.show()

fig, ax = plt.subplots()
ax.imshow(Matriz_ADA, cmap = 'turbo_r')
ticks = ["0–10%", "10–20%", "20–30%","30–40%","40–50%","50–60%","60–70%","
    70–80%","80–90%","90–100%"]
ax.set_xticks(list(np.arange(0,10,1)), labels = ticks, fontsize = 13)
ax.set_yticks(list(np.arange(0,10,1)), labels = ticks, fontsize = 13)
plt.setp(ax.get_yticklabels(), rotation=45, ha="right", rotation_mode="
    anchor")
plt.setp(ax.get_xticklabels(), rotation=45, ha="right", rotation_mode="
    anchor")
ax.set_ylabel("Variables reales", fontsize = 15)
ax.set_xlabel("Variables predichas", fontsize = 15)
ax.set_title("Matriz de confusión del ADABoost", fontsize = 15)
for i in range(len(Matriz_ADA[0,])):
    for j in range(len(Matriz_ADA[0,])):
        ax.text(j,i,Matriz_ADA[i,j],ha="center", va="center", color="w",
    fontsize = 13)
plt.savefig("MatrizConfusiónBoliviaADA.png", bbox_inches = "tight")
plt.show()

fig, ax = plt.subplots()
ax.imshow(Matriz_NN, cmap = 'turbo_r')
ticks = ["0–10%", "10–20%", "20–30%","30–40%","40–50%","50–60%","60–70%","
    70–80%","80–90%","90–100%"]
ax.set_xticks(list(np.arange(0,10,1)), labels = ticks, fontsize = 13)
ax.set_yticks(list(np.arange(0,10,1)), labels = ticks, fontsize = 13)
plt.setp(ax.get_yticklabels(), rotation=45, ha="right", rotation_mode="
    anchor")
plt.setp(ax.get_xticklabels(), rotation=45, ha="right", rotation_mode="
    anchor")
ax.set_ylabel("Variables reales", fontsize = 15)
ax.set_xlabel("Variables predichas", fontsize = 15)
ax.set_title("Matriz de confusión de la Red Neuronal", fontsize = 15)
for i in range(len(Matriz_NN[0,])):
    for j in range(len(Matriz_NN[0,])):
        ax.text(j,i,Matriz_NN[i,j],ha="center", va="center", color="w",
    fontsize = 13)
```

```
plt.savefig("MatrizConfusiónBoliviaNN.png",bbox_inches = "tight")
plt.show()
```

Código para el dataset de Bolivia