

UNIVERSIDAD TÉCNICA DE AMBATO



FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

MAESTRÍA EN MATEMÁTICA APLICADA

COHORTE 2021

Tema: Aplicación de algoritmos de Machine Learning para predecir la deserción estudiantil en alumnos de primer y segundo semestre en universidades públicas del Ecuador.

Trabajo de Titulación, previo a la obtención del Título de Cuarto Nivel de Magíster en Matemática Aplicada

Modalidad del Trabajo de Titulación: Proyecto de desarrollo

Autor: Ing. Cristóbal Alejandro Rodríguez Vásconez, MSc.

Director: Ing. Marco Enrique Benalcázar Palacios, PhD.

Ambato – Ecuador

2023

A la Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

El Tribunal receptor del Trabajo de Titulación, presidido por: la Ingeniera Elsa Pilar Urrutia Urrutia Magíster, e integrado por los señores: Ingeniero Héctor Alberto Luzuriaga Jaramillo Magíster y el Ingeniero Víctor Santiago Manzano Villafuerte Magíster, designados por la Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, para receptar el Trabajo de Titulación con el tema: “APLICACIÓN DE ALGORITMOS DE MACHINE LEARNING PARA PREDECIR LA DESERCIÓN ESTUDIANTIL EN ALUMNOS DE PRIMER Y SEGUNDO SEMESTRE EN UNIVERSIDADES PÚBLICAS DEL ECUADOR” elaborado y presentado por el señor Ingeniero Cristóbal Alejandro Rodríguez Vásconez Magíster, para optar por el Título de cuarto nivel de Magíster en Matemática Aplicada; una vez escuchada la defensa oral del Trabajo de Titulación, el Tribunal aprueba y remite el trabajo para uso y custodia en las bibliotecas de la UTA.

Ing. Elsa Pilar Urrutia Urrutia. Mg.

Presidenta y Miembro del Tribunal

Ing. Héctor Alberto Luzuriaga Jaramillo, Mg.

Miembro del Tribunal

Ing. Víctor Santiago Manzano Villafuerte., Mg.

Miembro del Tribunal

AUTORÍA DEL TRABAJO DE TITULACIÓN

La responsabilidad de las opiniones, comentarios y críticas emitidas en el Trabajo de Titulación presentado con el tema: “Aplicación de algoritmos de Machine Learning para predecir la deserción estudiantil en alumnos de primer y segundo semestre en universidades públicas del Ecuador.”, le corresponde exclusivamente a: Ingeniero Cristóbal Alejandro Rodríguez Vásquez, Magíster, Autor; bajo la dirección del Ingeniero Marco Enrique Benalcázar Palacios, PhD, Director del Trabajo de Titulación; y el patrimonio intelectual a la Universidad Técnica de Ambato.

Ing. Cristóbal Alejandro Rodríguez Vásquez, MSc

c.c.: 1804383089

AUTOR

Ing. Marco Enrique Benalcázar Palacios, PhD.

c.c.:1804029732

DIRECTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que el trabajo de titulación, sirva como un documento disponible para su lectura, consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos de mi trabajo, con fines de difusión pública, además apruebo la reproducción de este, dentro de las regulaciones de la Universidad.

Ing. Cristóbal Alejandro Rodríguez Vásquez, MSc

c.c.: 1804383089

ÍNDICE GENERAL DE CONTENIDOS

Portada.....	i
A la Unidad Académica de Titulación.....	ii
AUTORÍA DEL TRABAJO DE TITULACIÓN	iii
DERECHOS DE AUTOR.....	iv
ÍNDICE GENERAL DE CONTENIDOS.....	v
ÍNDICE DE TABLAS	viii
ÍNDICE DE FIGURAS.....	ix
AGRADECIMIENTO.....	x
DEDICATORIA	xi
RESUMEN EJECUTIVO	xiii
CAPÍTULO I. EL PROBLEMA DE INVESTIGACIÓN.....	14
1.1 Introducción.....	14
1.2 Justificación.....	15
1.3 Objetivos.....	16
1.3.1 Objetivo General	16
1.3.2 Objetivos Específicos:	17
CAPÍTULO II. ANTECEDENTES INVESTIGATIVOS	18
2.1 Estado del arte.....	18
2.2 Deserción estudiantil en Ecuador.....	20
2.3 Redes neuronales artificiales.....	23
2.4 Estandarización y normalización	25
2.5 Aprendizaje de las neuronas.....	27
2.6 Funciones de transferencia/activación	27
2.7 Métricas de rendimiento de un modelo.....	28
2.8 Codificación de variables para clasificadores de redes neuronales	29
CAPÍTULO III. MARCO METODOLÓGICO	31

3.1 Ubicación	31
3.2 Población y muestra	31
3.3 Recolección de información.....	32
3.4 Tratamiento de datos y normalización	32
3.5 Adaptación del código.....	34
3.6 Balanceo de datos.....	35
3.7 Submuestreo y sobremuestreo de datos	37
3.8 Arquitectura del modelo.....	37
3.9 Métricas a evaluar en el modelo.....	38
3.10 Validación cruzada.....	39
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN.....	40
4.1 Procesamiento de datos	40
4.2 Selección de variables predictoras o respuesta	41
4.3 Selección de método para balanceo de datos	42
4.4 Obtención de la mejor arquitectura	43
4.5 Resultados del modelo y matriz de confusión.....	44
4.6 Capas de la validación cruzada	46
4.7 Métricas de evaluación de rendimiento.....	46
4.8 Comparación de datos desbalanceados y balanceados.....	49
4.9 Pruebas con disminución de variables de entrada.....	50
CAPÍTULO V. CONCLUSIONES Y REFERENCIAS BIBLIOGRÁFICAS	52
5.1 Conclusiones	52
5.2 Recomendaciones.....	53
5.3 Referencias Bibliográficas	54
ANEXOS.....	58
Anexo 1. Variables y atributos del modelo. Elegidos o no para el análisis.....	58

Anexo 2. Resultados de 10 pruebas aleatorias para datos desbalanceados y balanceados para entrenamiento y testeo. (Valores más altos de las pruebas pintados en verde)	62
Anexo 3. Matriz de confusión de 10 pruebas del modelo	64
Anexo 4. Resultados de 10 pruebas aleatorias para datos desbalanceados y balanceados para testeo con 17 variables y 10 variables de entrada.....	67

ÍNDICE DE TABLAS

Tabla 1.	Factores y razones de deserción	21
Tabla 2.	Metodología para estudio de datos	23
Tabla 3.	Métricas para evaluación de rendimiento de clasificadores	29
Tabla 4.	Mejores métricas con primera recogida de datos.	36
Tabla 5.	Selección de categorías del estudio.	42
Tabla 6.	Mejor arquitectura para el modelo.	44
Tabla 7.	Métricas para datos desbalanceados con la mejor arquitectura.	44
Tabla 8.	Métricas para datos balanceados con la mejor arquitectura.	44
Tabla 9.	Métricas de entrenamiento del modelo para datos desbalanceados.....	47
Tabla 10.	Métricas de entrenamiento del modelo para datos balanceados	47
Tabla 11.	Métricas del testeo para datos desbalanceados.	48
Tabla 12.	Métricas del testeo para datos balanceados.....	48
Tabla 13.	Clasificación en el testeo para datos desbalanceados	49
Tabla 14.	Clasificación en el testeo para datos balanceados.....	49
Tabla 15.	Comparación de métricas entre datos desbalanceados y balanceados. 50	
Tabla 16.	Métricas de testeo usando 17 variables de entrada.	51
Tabla 17.	Métricas de testeo usando 10 variables de entrada.	51

ÍNDICE DE FIGURAS

Figura 1.	Técnicas de Machine Learning usadas en retención	22
Figura 2.	Estructura biológica de una neurona	24
Figura 3.	Estructura de un sistema neuronal	25
Figura 4.	Características de funciones de activación más utilizadas.	28
Figura 5.	Adecuación de una variable categórica con método <i>One Hot Encoding</i> . 33	
Figura 6.	Datos para las clases 0 y 1 tras la primera recolección de datos.	36
Figura 7.	Datos para las clases 0 y 1 tras la segunda recolección de datos.	37
Figura 8.	Ejemplo de validación cruzada.	39
Figura 9.	Esquema de procesamiento de datos.	41
Figura 10.	Esquema de dos técnicas de balanceo de datos.	43
Figura 11.	Representación de matriz de confusión y ejemplo.	45

AGRADECIMIENTO

Agradezco de todo corazón:

Primero a quienes han hecho este trabajo posible: al Centro de Estudios Quality Up y los alumnos que me otorgaron su ayuda con sus experiencias traducidas en datos, al profesor Marco Benalcázar por su ayuda constante y valiosa y a la Universidad Técnica de Ambato por ser cuna de este proyecto y sueño.

En segundo lugar a los pilares de mi constancia, quienes me recuerdan ser fuerte en cada momento: a mi madre Inés, que gracias a ella los astros están de mi lado y es ejemplo de ser correcto y bondadoso con todos; a mi padre Neún, ejemplo de siempre ser amable con las personas; a mi hermana Magui, que me recuerda equilibrar las risas con la madurez; a mi hermana Lety, que me recuerda lo valioso de ser original; y a mi hermano por elección, Chris, por hacerme ver lo importante que es salir adelante solo, pero también estar a mi lado en este camino de aprendizaje que es la vida, buscando el equilibrio, por las risas y las lágrimas.

Por último, al niño que fui, por haber sido fuerte a pesar de todas las circunstancias y mantener su integridad intacta y un corazón noble. Lo logramos.

DEDICATORIA

Dedico este trabajo a quienes han sido mis alumnos en toda mi trayectoria docente.

Alumnos del pasado, presente y los que vendrán.

A las personas que siempre confían en mi éxito antes de que lo conciba: mis padres

Inés y Neún; mis hermanas Lety y Magui; mi amigo Christian y por último a mi sobrino Emilio, un aprendiz con entusiasmo invaluable.

UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

MAESTRÍA EN MATEMÁTICA APLICADA

COHORTE 2021

TEMA:

APLICACIÓN DE ALGORITMOS DE MACHINE LEARNING PARA PREDECIR
LA DESERCIÓN ESTUDIANTIL EN ALUMNOS DE PRIMER Y SEGUNDO
SEMESTRE EN UNIVERSIDADES PÚBLICAS DEL ECUADOR.

MODALIDAD DE TITULACIÓN: Proyecto de desarrollo

AUTOR: Ing. Cristóbal Alejandro Rodríguez Vásquez, MSc.

DIRECTOR: Ing. Marco Enrique Benalcázar Palacios, PhD.

FECHA: 04 de abril de 2023.

RESUMEN EJECUTIVO

Se estima que en Ecuador la tasa de deserción en los dos primeros semestres de universidad es del 20%. Existen factores socioeconómicos que influyen en el abandono académico de un estudiante. La carencia de programas que atiendan la insatisfacción estudiantil provoca que no se detecten problemas a tiempo y no se puedan aplicar acciones correctivas oportunamente. En este proyecto se aplican técnicas de *Machine Learning* para predecir la deserción estudiantil a partir de factores seleccionados: socioeconómicos, psicológicos, demográficos y académicos. Partimos de la recolección y tratamiento de datos y se usaron Redes Neuronales Artificiales para crear un modelo que clasifica a un estudiante entre desertor o a salvo de deserción. Se evalúan las métricas *Accuracy*, sensibilidad y especificidad para determinar qué tan eficiente es el modelo. El modelo final es capaz de clasificar estudiantes a salvo de deserción de forma correcta el 87% de las veces y logra clasificar a desertores de forma correcta el 60% de las veces.

Palabras Claves: BALANCEO DE DATOS, CLASIFICACIÓN BINARIA, DESERCIÓN ESTUDIANTIL, MACHINE LEARNING, ONE HOT ENCODING, REDES NEURONALES ARTIFICIALES, VALIDACIÓN CRUZADA.

CAPÍTULO I.

EL PROBLEMA DE INVESTIGACIÓN

1.1 Introducción

Uno de los problemas a enfrentar en la universidad es la deserción en los primeros años de las carreras. En Latinoamérica se estima que el 50% de estudiantes deserta en los dos primeros semestres [1]. La deserción universitaria afecta tanto en los ámbitos personales como en los institucionales, sociales y económicos. Anteriormente los niveles de deserción podían reflejar la dificultad y prestigio de una carrera, sin embargo, hoy reflejan ineficiencia de la misma. Aumentar la retención de estudiantes en la educación superior es el primer paso para mejorar las tasas de graduación y poder predecir la deserción brinda la oportunidad de asesorar oportunamente a los estudiantes [2].

Una serie de factores socioeconómicos llevan a la insatisfacción de un estudiante por su desarrollo académico. Es común que tutores y otras autoridades identifiquen estas señales cuando es difícil revertir los problemas. Un reconocimiento tardío conlleva a que no se pueda encaminar a un discente a mejorar su situación.

Adicionalmente, detrás de la problemática de la deserción estudiantil se pueden destapar otros inconvenientes en los servicios de la educación universitaria. Uno de ellos es la falta de asesoría o programas de tutelaje para universitarios novatos. A diferencia de la preocupación por ayudas económicas las instituciones tradicionales suelen carecer de asesoría frente a problemas de insatisfacción académica. Esto desemboca en que un alumno con dificultades no sepa a quién acudir para recibir ayuda.

Es difícil asesorar o incluso saber a quién asesorar por el desconocimiento de indicios de quiénes son susceptibles a la deserción académica. Con herramientas de *Machine Learning* se procura encontrar patrones en datos de entrada para preconcebir quiénes necesitan ayuda. El reto que se presenta es discernir cuáles son esas variables de entrada. Necesitamos recolectar datos de personas que ya se han enfrentado a la deserción y extrapolar estos para crear una base estadística que prediga probabilidad de abandono escolar de un postulante.

Este proyecto se enfoca en aplicar estrategias de *Machine Learning* a la predicción de éxito académico a partir de factores variados. Nos enfocaremos en el uso de factores

socioeconómicos, psicológicos, demográficos y académicos [3]. La metodología que se utilizará comienza con la recopilación y tratamiento de datos de los factores mencionados. Estos datos son procesados con la técnica de aprendizaje estadístico denominada Red Neuronal Artificial (RNA) que, en este trabajo, se usa para clasificar un estudiante en riesgo de deserción o no.

1.2 Justificación

El ámbito de la educación ha evolucionado conforme lo ha hecho la raza humana. Al ser las instituciones educativas el principal medio para la transmisión de conocimientos es común que estas instituciones sufran transformaciones constantes. Las dinámicas sociales actuales requieren de profesionales competentes y las universidades acogen actualmente a un mercado estudiantil mayor que en décadas pasadas. En Ecuador, en los años 50, el cupo de ingreso a la universidad era en promedio de 2000 personas, esta cantidad se quintuplica para 1970 y se vuelve a quintuplicar para el año 2000 [4]. Para la tercera década del siglo XXI, cada año ingresan unas 130000 personas a las universidades ecuatorianas. Los nuevos universitarios requieren formación que los haga destacar frente a un rápido y muy competitivo crecimiento industrial [1].

El presente proyecto se enfoca en el éxito estudiantil, entendiéndose como la probabilidad de que un estudiante logre los objetivos de un curso planteado, que en general implica aprobarlo. Este análisis se hace a partir de datos previos como el cumplimiento inicial del estudiante, la asistencia, así como otros datos sociodemográficos. Para la deserción estudiantil se tienen en cuenta las causas multifactoriales por las que un estudiante toma la decisión de discontinuar sus estudios [1]. Detectar a tiempo el riesgo de fracaso estudiantil nos ayudaría a solventar otros problemas como la falta de motivación a largo plazo y la incertidumbre vocacional.

Una retención estudiantil alta representa calidad de las universidades y refleja la reputación de dichas instituciones. Un índice favorable de retención vuelve una IES atractiva a los ojos de los nuevos demandantes de educación de tercer nivel o posgrados [5]. La deserción estudiantil no es un problema aislado de pocos países, sino que parece ocurrir de forma indistinta alrededor del mundo. Es evidenciable que con el aumento de estudiantes universitarios, también aumenta la deserción. Como media, podemos mencionar que en universidades de Latinoamérica, la cantidad de desertores

respecto a quienes culminan sus estudios es de 1 a 2 [1]; mientras que en ciertos países asiáticos, se ha observado que en las dos últimas décadas la tasa de deserción ha aumentado del 4 al 15%. [6]

Llevar a cabo modelos para la predicción de la deserción, tiene su utilidad en que se puede prever la ocurrencia de estos casos y entonces aplicar planes para la disuasión del abandono estudiantil. Mediante esta acción se pueden contrarrestar efectos como la baja autoestima, proyectos de vida frustrados y más acciones que pueden llevar a estudiantes a episodios de depresión y estos al abandono psicológico, consumo de sustancias psicotrópicas y más elementos dañinos [1].

¿Qué tan influyentes son ciertos factores para detectar el éxito académico? Por ejemplo, sabemos que en Ecuador ingresan a la universidad un 20% más los hijos de hispanohablantes que de quichuahablantes y que hijos de madres que han recibido el bono de desarrollo humano ingresan a la universidad un 16% menos[4]. Sabemos que en Ecuador ingresan a la universidad unas 5000 mujeres más que hombres y que el hijo de una persona en el último quintil de nivel económico tiene cerca del 50% más de probabilidades de obtener un cupo que un hijo de alguien en el quintil 1[4]. ¿Son estos factores causales del ingreso y de la deserción estudiantil?

El campo de la educación puede ser explorado con el uso de estrategias de inteligencias artificiales [7].Tareas que se han evaluado con Aprendizaje Estadístico han abarcado desde la retención estudiantil, la planificación escolar, el desempeño docente, etc. [8]

El ámbito académico ha sido ampliamente explorado a nivel mundial por investigadores, a través de métodos de aprendizaje estadístico. El valor de este estudio se centra en la poca aplicación que se le ha dado de forma local. Al detectar estos problemas en una etapa temprana, también es posible actuar en función de revertir estos eventos, dándoles a los estudiantes la oportunidad de retomar su rumbo educativo y hacerlo con éxito.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un modelo matemático para predecir la deserción estudiantil en alumnos de primer y segundo semestre mediante aplicación de Redes Neuronales Artificiales.

1.3.2 Objetivos Específicos:

- Adecuar distintos tipos de variables de entrada para lograr coherencia entre ellas y que el programa pueda operar con ellas.
- Encontrar la arquitectura que permita obtener los mejores resultados en la Red Neuronal Artificial.
- Aplicar un algoritmo de redes neuronales que distinga a alumnos en riesgo de deserción de alumnos sin riesgo de deserción.
- Comparar los resultados de una red neuronal con datos balanceados y desbalanceados.

CAPÍTULO II.

ANTECEDENTES INVESTIGATIVOS

En el siguiente apartado se analizará el estado del arte y marco teórico del proyecto. Se abarcarán datos sobre la deserción estudiantil en el país y el mundo. También se analizarán cuáles son variables comúnmente utilizadas para la predicción de éxito académico al usar métodos de *Machine Learning*. Se pondrán en contexto nociones básicas sobre las Redes Neuronales Artificiales, su funcionamiento, la estructura de una red y las métricas para evaluar la eficiencia de este modelo.

2.1 Estado del arte

En el artículo “Una aproximación conceptual a la retención estudiantil en Latinoamérica” de Pedraza, Díaz y Cabrales se puntualizan conceptos sobre la retención y deserción estudiantil en países latinos, incluido Ecuador. Se resumen datos sobre la matrícula neta y porcentaje de graduados de diferentes naciones. Los autores analizan diferentes causas por las que un estudiante no logra completar sus estudios y se sugieren acciones para reducir la tasa de deserción en universidades [1].

El estudio realizado por Post llamado “Las Reformas Constitucionales en el Ecuador y las Oportunidades para el Acceso a la Educación Superior desde 1950” resumen la evolución de la educación universitaria en Ecuador en los últimos 70 años [4]. Hace un análisis específico de la deserción en las universidades del país y cómo existen ciertos factores, sobre todo socioeconómicos que pueden tener influencia en que un estudiante no pueda ingresar a educación superior o que no la complete. Post muestra datos sobre cómo la etnia, los estratos económicos y los índices de desarrollo humano están relacionados con las oportunidades de obtener cupos en las universidades ecuatorianas.

En la investigación de Tsai, Chen, Shiao, Ciou y Wu se analiza la deserción estudiantil junto con factores que la causan. Se estudia la evolución de acciones llevadas a cabo para incrementar el éxito estudiantil universitario. También se aplican métodos de *Machine Learning* para predecir la deserción. Se utilizan los métodos de perceptrón multicapa y regresión logística combinando diferentes arquitecturas. En este trabajo se observa un contraste alto entre resultados de sensibilidad (predicción de desertores) y especificidad (predicción de estudiantes que continúan sus estudios) [6]. También se

establece una ruta de acción desde la predicción de riesgo de deserción hasta la intervención estudiantil.

Delen realiza una comparación entre mecanismos de *Machine Learning* para predecir la deserción estudiantil. Utilizando un dataset de más de 16000 datos y 39 variables de entrada analiza qué métodos tienen mejores resultados. En su investigación 4 modelos logran una exactitud de entre 86 y 87% [9]. Los métodos con los que obtuvo mejores resultados son *Support Vector Machines*, Árboles de Decisión, Redes Neuronales y Regresión Logística. Finalmente diferencia qué variables de entrada tienen mayor incidencia en la predicción utilizando la regresión logística.

En la investigación de Gray y Perkins llamada “Utilizing early engagement and machine learning to predict student outcomes” se aplican algoritmos de *Machine Learning* para lograr una detección temprana de estudiantes que requieren seguimiento por estar en riesgo de reprobado un curso. Se usa un modelo de clasificación con 4 variables de entrada asociadas a calificaciones en las primeras semanas de clase y se usa una variable de salida con 5 clases diferentes según los estatus que usa la escuela Bangor para determinar el resultado de un estudiante al final del año. El modelo de los autores logra una exactitud del 97% en la detección de estudiantes en riesgo. Concluyen que aunque el modelo tiene alta eficiencia, cada generación de estudiantes es diferente y hace falta un análisis nuevo para cada una [10].

Musso, Rodríguez y Cascallar hacen una investigación sobre los factores que inciden en la deserción estudiantil y una comparación entre mecanismos tradicionales de detección y métodos de *Machine Learning*. Para las variables que utilizaron concluyen que los pesos relativos de las variables individuales son todas menores al 7% y que es más bien la combinación de factores la que tiene incidencia en la predicción. Los autores hacen énfasis en cómo la autopercepción de sentirse apoyado y de tener un grupo de amigos estable influyen en la decisión de mantenerse dentro de los programas académicos [11].

González y Peñaloza aplican mecanismos de *Machine Learning* para predecir la deserción estudiantil en la materia de Mecánica de la Universidad Nacional Abierta. En este estudio utilizaron variables de entrada a partir de factores individuales, académicos, socioeconómicos e institucionales. Aplicaron las técnicas de Árbol de

Decisión, Random Forest y Regresión Logística logrando una exactitud del 52, 59 y 54% respectivamente [12].

En la investigación de Cardona, Cudney, Hoerl y Snyder llamada “Data Mining and Machine Learning Retention Models in Higher Education” se hace una revisión sistemática de mecanismos de *Machine Learning* utilizados para predecir la probabilidad de que un estudiante abandone sus estudios. En esta investigación se analizan 60 investigaciones diferentes de las cuáles se observa a las Redes Neuronales Artificiales como método más popular [5]. También se analizan las variables de entrada utilizadas en las distintas investigaciones.

Alenezi y Faizal analizan de forma sistemática el uso de la colaboración abierta para la obtención de datasets, además analizan los mecanismos utilizados para predecir el éxito académico [7].

En la investigación “Una lectura sobre deserción universitaria en estudiantes de pregrado desde la perspectiva de minería de datos, realizada por Timarán se hace un estudio sobre deserción estudiantil en la Universidad de Nariño utilizando minería de datos [13]. Se hace un aporte valioso sobre el tratamiento de datos y la selección de atributos para los modelos utilizados.

2.2 Deserción estudiantil en Ecuador

En Ecuador la Secretaría de Educación Superior, Ciencia Tecnología e Innovación (SENESCYT), es el órgano regulador del Sistema Nacional de Nivelación y Admisión (SNNA), que actualmente rige para todas las instituciones del sistema educativo a nivel nacional. Tiene como objetivo principal garantizar el acceso a la educación superior gratuita basado en igualdad de oportunidades, meritocracia y transparencia, a través del uso de nuevas tecnologías. El abandono a la universidad por parte de los estudiantes impacta económicamente a las universidades, pues representan un recurso desperdiciado. En la investigación realizada en [14] se determinan algunos factores principales para elevar el rendimiento académico como son la tutoría y la motivación académica.

Cuando un estudiante se convierte en desertor, es necesario indagar sobre los factores y razones que ocasionaron el abandono. En [3] se indica que entre los factores principales que inciden en la deserción, se encuentran aquellos que pertenecen a

problemas personales, académicos, institucionales y económicos, como se muestra en la Tabla 1.

Tabla 1. Factores y razones de deserción [3]

Factores	Razón	Definición
Académica	Bajo rendimiento	Aquellas causas relacionadas con el bajo rendimiento e insatisfacción con el programa.
	Cambio de universidad	
	Insatisfacción con el programa	
	Pérdida de derecho al programa	
	Orientación vocacional y motivación	
Económica	Dificultad con el pago de transporte	Todos los aspectos que implican el dinero, es decir desde el pago de la matrícula hasta el sostenimiento de los estudiantes dentro de cada uno de los programas.
	Insolvencia económica	
	Pérdida de empleo	
Laboral	Cambio de empleo	Aspectos relacionados con inconvenientes a asuntos laborales.
	Cambio de horario laboral	
	Capacitación laboral	
	Dificultad con el desplazamiento	
	Viaje	
Personal	Cambio de domicilio	Asuntos relacionados con temas personales
	Embarazo	
	Problemas de salud	
	Falta de tiempo para estudiar	
	Problemas familiares	

En la investigación realizada por [5], se analizan las siguientes técnicas de “*Machine Learning*” usadas para predecir la finalización de estudios, considerando el rendimiento del modelo.

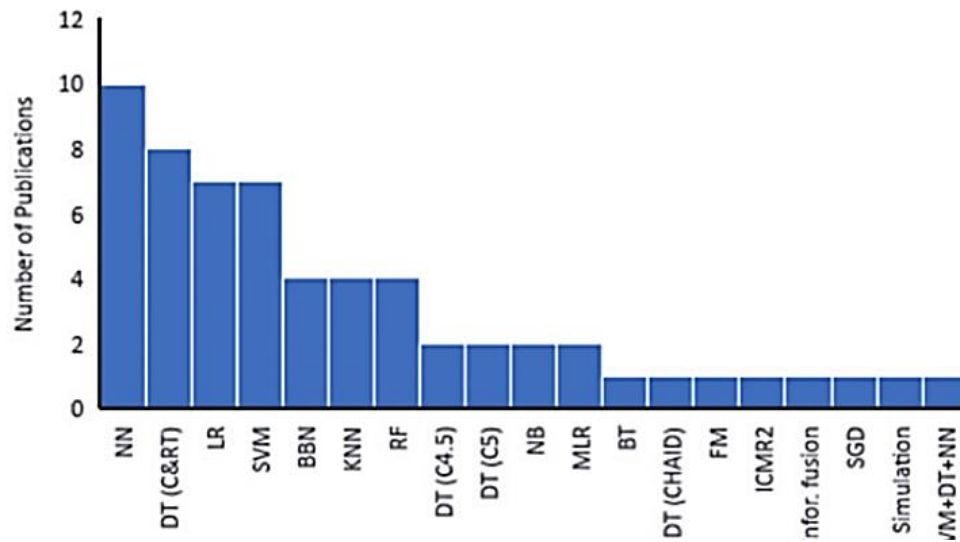


Figura 1. Técnicas de *Machine Learning* usadas en retención [5].

Donde:

NN: *Neural networks*

DT: *Decision tree*

LR: *Logistic regression*

SVM: *Support vector machines*

BBN: *Bayesian belief network*

KNN: *K-nearest neighbour*

RF: *Random forest*

La metodología propuesta en [2] sugiere dar atención a las actividades generales para el desarrollo de proyectos de minería de datos educativos.

Tabla 2. Metodología para estudio de datos [2]

Análisis de datos	<p>Recolección inicial de datos: Capturar al momento de la inscripción del estudiante.</p> <p>Verificar la calidad de los datos: Establecer un mecanismo de acceso a otras bases de datos para validar la calidad de los datos ingresados, tales como la nota del récord académico y la nota de examen de ingreso a la educación superior.</p>
Preparación de los datos	<p>Integrar datos: Integración de la nota de educación media con el objeto de ponderar el atributo de nota media.</p>
Modelado	<p>Selección de la técnica de modelado: Árbol de decisión Redes neuronales Cluster K – medianas</p>
Evaluación	<p>Evaluar en modelo: Considerar que el atributo de la nota de educación media tiene una capacidad predictiva sobre el rendimiento de los estudiantes.</p>

2.3 Redes neuronales artificiales

Las redes neuronales artificiales (RNA's) son una de las ramas y herramientas de la Inteligencia Artificial. Su comportamiento trata de replicar el funcionamiento del cerebro y específicamente de las neuronas, que actúan como procesadores de información [15]. Este paralelismo entre el funcionamiento del cerebro y una computadora busca replicar funciones de aprendizaje que aplica nuestro sistema cognitivo. La diferencia fundamental es que las neuronas biológicas tienen un proceso de aprendizaje por medio de los estímulos que el individuo recibe mediante

experiencias, mientras que las redes neuronales tienen una programación, simulación o arquitectura que le permite llevar a cabo un aprendizaje [15].

Las neuronas de una red tratan de asemejarse a la función de una neurona biológica, mientras que su estructura es un paralelismo de esta. En la figura 2 se muestra la estructura de una neurona. Las funciones analógicas a la red neuronal son que la dendrita es un canal de recepción, es decir que son las responsables de obtener la información de entrada o *inputs*; los somas son los encargados de computar la información de entrada y luego conformar la información de salida; por último, los axones funcionan como emisores de señales de salida, ya sea a otras dendritas o a una capa final [15].

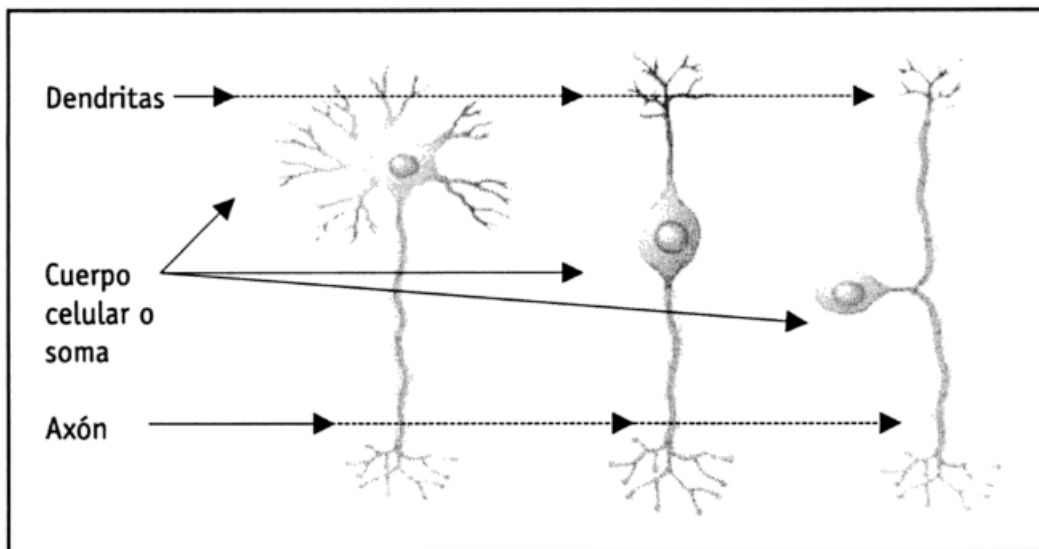


Figura 2. Estructura biológica de una neurona [15].

La RNA está conformada por unidades mínimas de procesamiento de información denominadas neuronas artificiales [15], estas actúan de forma paralela y pueden ser de tres tipos: de entrada cuando reciben información del entorno y lo procesan; de salida cuando reciben información de otras neuronas y emiten información fuera del sistema, es decir que están en el último nivel de procesamiento; y neuronas ocultas cuando actúan como un intermedio entre las neuronas de entrada y salida [16].

Cierta cantidad de neuronas son utilizadas dentro de una misma capa. La capa de entrada, capa de salida y capas ocultas conforman la red neuronal y esta red junto con

una arquitectura y datos de entrada y salida forman un sistema neuronal. Como se muestra en la Figura 3.

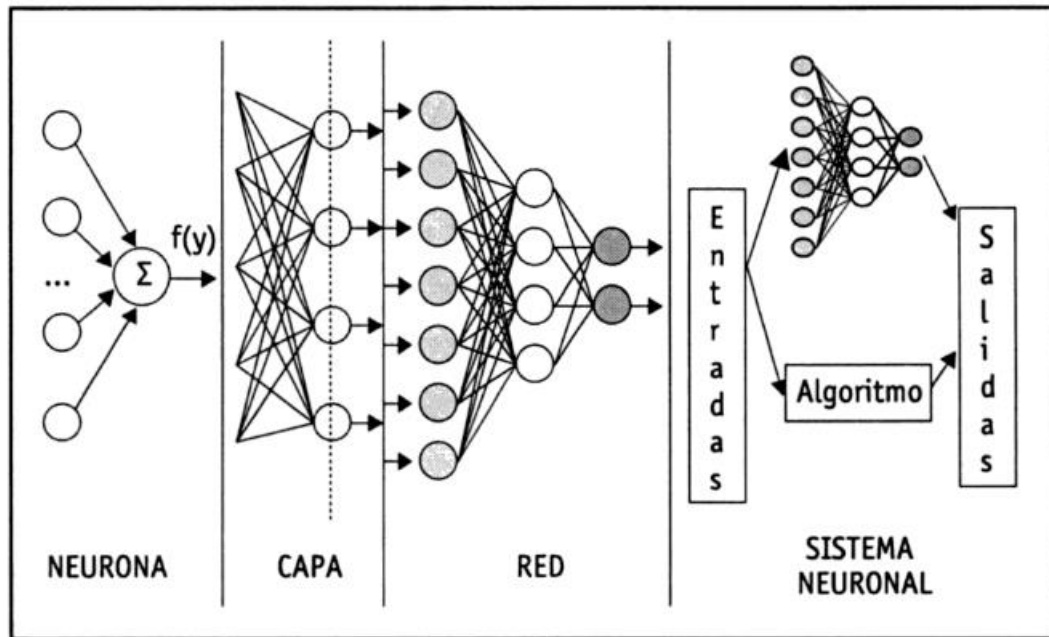


Figura 3. Estructura de un sistema neuronal [16].

Cuando una RNA es alimentada con información de entrada y la red es entrenada por sus neuronas en diferentes capas, se pueden resolver diferentes tipos de problemas: aprendizaje adaptativo, autoorganización, tolerancia a fallos, operaciones en tiempo real, inserción en la tecnología actual, entre otros [16].

2.4 Estandarización y normalización

En el tratamiento de datos estadísticos nos encontramos con que las variables tienen diferentes características y formatos, y por lo tanto su interpretación puede verse afectada por la diversidad en las variables utilizadas. Para solventar esta diversidad de tipos de entradas se busca formatear los datos de manera que tengan coherencia entre sí y la obtención de salidas cumpla con las directrices que busca la empresa o entidad que desea obtener interpretaciones de los datos estadísticos [17].

La estandarización de datos es una estrategia que asigna puntuaciones a los datos de entrada a partir de conocer la media aritmética y la desviación típica. Se busca establecer una distribución estandarizada normal de los datos desde el más bajo al más alto, siguiendo el patrón de una curva estadística normal [18]. La puntuación otorgada

será una representación de lo alejada que está la data de la media que se logra mediante una codificación que luego puede ser devuelta hasta el formato original deseado [18].

La estandarización de un dato se lleva a cabo con la ecuación 1:

$$Z_X = \frac{X - \bar{X}}{S_X} \quad (1)$$

donde:

Z_X : Puntuación estandarizada

X : Puntuación directa

\bar{X} : Media de la muestra

S_X : Desviación típica de la muestra

Las puntuaciones Z_X siempre tendrán una media aritmética de 0 puesto que todos sus valores han sido regulados según cuánto se aleja de la media aritmética. Su desviación estándar será de 1 [18].

La normalización de datos se diferencia en que el rango de valores formateados estará entre de 0 y 1, valores asignados para el valor mínimo y máximo respectivamente. Esta técnica será útil cuando buscamos que nuestra variable de salida sea binaria como en los casos de clasificación. Para normalizar un conjunto de datos usaremos la ecuación 2.

$$Z_X = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2)$$

donde:

Z_X : Puntuación normalizada

X : Puntuación directa

X_{max} : Valor máximo de la muestra

X_{min} : Valor mínimo alto de la muestra

2.5 Aprendizaje de las neuronas

El aprendizaje en las neuronas se da a través de las experiencias obtenidas a través de toda la red neuronal. Este aprendizaje debe cumplir con dos características que son el “Ser significativo” que implica una cantidad relevante de datos según la complejidad del problema que estamos analizando. También debe “Ser representativo” haciendo referencia a que los datos con que alimentamos el sistema neuronal deben ser equilibrados, diversos y aleatorios [16].

Existen dos tipos iniciales de aprendizaje que son el supervisado y no supervisado, dependiendo de si el arquitecto o diseñador de la red supervisa los resultados de la red [16]. En el aprendizaje supervisado este arquitecto evalúa las salidas obtenidas y cuando estas no son las esperadas se corrige la arquitectura hasta obtener mejores respuestas [19].

Dentro del aprendizaje supervisado catalogamos el aprendizaje por corrección de error, por refuerzo y el estocástico. En el aprendizaje por corrección de error se asigna al modelo las entradas y salidas deseadas y se busca que las salidas obtenidas tengan la diferencia mínima respecto a la establecida en un inicio, para ello se modificarán los pesos sinápticos. En el aprendizaje por refuerzo se califican las salidas de la red según su cercanía con el resultado esperado, retroalimentando cada vez si la salida está más cerca de éxito o fracaso afinando estos resultados. En el aprendizaje estocástico se hacen variaciones mínimas a la arquitectura de la red y se observa la variación entre las salidas, cuanto menor sea o sea decreciente el modelo será mejor [16].

Los modelos no supervisados no requieren de una definición inicial de salidas deseadas y de una regularización por parte de un diseñador, sino que el modelo se autorregula en función de la activación que tengan pares de neuronas [16].

2.6 Funciones de transferencia/activación

La información procesada en las neuronas debe pasar por una etapa de clasificación en la que se le asignará un valor cuando supera un umbral o se queda debajo de él [19]. Entre las funciones más comunes nos encontramos con la lineal, función escalón, función mixta, función sigmoidea, función gaussiana y función sinusoidal. En la Figura 4 se describen las características de cada tipo de función [15].


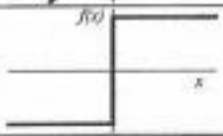
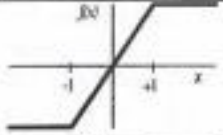
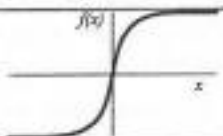
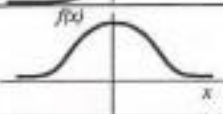
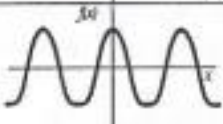
	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1+e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-ax^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(\omega x + \varphi)$	$[-1, +1]$	

Figura 4. Características de funciones de activación más utilizadas. [16]

2.7 Métricas de rendimiento de un modelo

Para calificar el modelo utilizaremos métricas de referencia que pautan qué tanto nos acercamos a un resultado deseado y poder recalibrarlo hasta obtener el mejor modelo posible. Las métricas más utilizadas en RNA son el *accuracy*, el *recall* o sensibilidad, la especificidad, la medida *F1-score*, la curva ROC y el índice kappa, aunque existen muchas otras útiles según el objetivo de análisis que se necesite [20].

La sensibilidad nos muestra la razón entre valores clasificados positivamente entre el total de datos, dándonos una idea de que tan certera ha sido la obtención de datos positivos, mientras que la especificidad nos indica la proporción de valores negativos clasificados correctamente, entre el total de datos clasificados como negativos. En un modelo el diseñador busca que estas dos métricas sean lo más altas posibles sin que la precisión disminuya su valor [20].

En la Tabla 3 se especifican características de las métricas mencionadas y las fórmulas de aplicación para las mismas:

Métrica	Fórmula	Descripción
Accuracy	$\frac{f_{11} + f_{00}}{N}$	Proporción de clasificaciones predichas de manera correcta sobre el total de instancias.
Recall (sensibilidad)	$\frac{f_{11}}{f_{11} + f_{01}}$	Proporción de casos positivos bien clasificados.
Especificidad	$\frac{f_{00}}{f_{00} + f_{10}}$	Proporción de casos negativos bien clasificados.
1-especificidad	$\frac{f_{10}}{f_{10} + f_{00}}$	Proporción de casos positivos mal clasificados (error Tipo I).
F1 – score	$2 * \frac{\text{accuracy} * \text{recall}}{\text{accuracy} + \text{recall}}$	Media armónica de las métricas accuracy y recall.
Índice kappa	$k = \frac{Po - Pe}{1 - Pe}$	Po = proporción de accuracy observado. Por lo tanto Po = accuracy.
Pe	$\frac{F_{1T} * f_{1T} + F_{0T} * f_{0T}}{N^2}$	Proporción de accuracy esperado por puro azar.
AUC	$\frac{\text{recall} - (1 - \text{especificidad}) + 1}{2}$	Probabilidad de clasificar correctamente una clase positiva al azar más que una negativa escogida al azar.

Tabla 3. Métricas para evaluación de rendimiento de clasificadores [20].

2.8 Codificación de variables para clasificadores de redes neuronales

Para el tratamiento de datos para una red neuronal artificial se usan métodos de codificación de variables con el fin de que las diversas variables tengan coherencia entre sí. Entre tipos de variables podemos mencionar las cuantitativas cuando tenemos valores numéricos reales, o cualitativas cuando se describen características no numéricas. Estas a su vez pueden ser ordinales cuando a pesar de no ser numéricas se las puede catalogar en jerarquías, por ejemplo, el grado de satisfacción, donde “Totalmente satisfecho” tiene jerarquía mayor que “Parcialmente satisfecho” y esta mayor que “Insatisfecho”; las variables cualitativas son no ordinales cuando no podemos jerarquizar los atributos en un orden específico, como la variable “Ciudad”, con atributos como “Ambato”, “Quito”, “Riobamba” no se pueden clasificar uno por encima de otro.

Para las variables cualitativas no ordinales, se pueden aplicar técnicas de codificación como el *One Hot Encoding*, la Codificación ordinal, la Codificación Suma, Polinomial, *Helmert*, Binaria, etc. [21].

El *One Hot Encoding* consiste en darle una categorización binaria a cada uno de los atributos de una variable. Esta técnica transforma una variable de n atributos en n variables con dos atributos, 0 y 1, para describir la ausencia o carencia de un atributo respectivamente [21]. Tiene mucha utilidad para tratar datos de una variable cualitativa no ordinal, aunque una desventaja es el aumento significativo de categorías que el modelo tendrá que analizar.

En el estudio realizado en [21] se define que los mejores métodos de codificación, que logran una precisión mayor al 90% son la Codificación Binaria, el *Backward Coding*, la Codificación polinomial, el *One hot encoding* y el *Sum Coding*, mientras que métodos como Codificación Ordinal y Codificación *Helmert* logran una precisión inferior al 90%.

CAPÍTULO III. MARCO METODOLÓGICO

En este capítulo se detallará el proceso para la clasificación de un estudiante como desertor en riesgo o no. Se abarca desde la recolección de los datos hasta la obtención de resultados en la clasificación binaria para las clases estudiante desertor (0) o a salvo de deserción (1).

3.1 Ubicación

El presente proyecto se lleva a cabo en la ciudad de Ambato.

3.2 Población y muestra

Este proyecto se enfoca en estudiantes de nuevo ingreso a instituciones de Educación Superior ecuatorianas. La matrícula histórica ha presentado una tendencia creciente en el último lustro. Para el año 2018, la tasa bruta de matrícula es del 37,34% y representa a 794.092 inscritos en instituciones nacionales. De este grupo se contabilizan a 113.259 (tasa bruta de matrícula en primer año del 5,32%) dentro de los inscritos en el primer año de carreras [22]. Este último dato corresponde a nuestra población.

Para el cálculo de la muestra partimos de que será de tipo probabilístico, puesto que todas las combinaciones tienen la misma posibilidad de ser elegidas [23]. Para este caso el método de muestreo será a partir de una muestra aleatoria simple donde se eviten sesgos sistemáticos y donde el tamaño de la muestra es decisivo para obtener generalizaciones [24].

Para saber cuántos datos necesitamos recolectar nos guiamos en la literatura que adecúa el cálculo de tamaño muestral. En [24] se nos indica la fórmula para muestras aleatorias y es:

$$n = \frac{z_{\alpha}^2 p(1 - p)}{d^2} \quad (3)$$

Donde:

n = número de sujetos necesarios

z_{α} = valor del coeficiente z según el nivel de confianza deseado

p = valor poblacional esperado

d = margen de error

Ya que hemos mencionado la población, también determinamos que el nivel de confianza será del 90%. El margen de error será del 8%. Estos valores son los comúnmente usados en modelos de aprendizaje por diversos autores al analizar la deserción universitaria [5].

Al aplicar estos datos, tenemos que:

$$n = \frac{1,96^2 * 0,0532(1 - 0.0532)}{0.05^2}$$

El tamaño mínimo de muestra resulta en 77,4 y la muestra mínima será de 78 estudiantes. Cabe recordar que este dato es el mínimo aplicable para la encuesta. Más adelante se comentará la cantidad con la que debe funcionar el modelo estadístico.

3.3 Recolección de información

Primero se establecieron las preguntas que se evalúan en la encuesta para obtención de datos. Responden a tres criterios: datos sociodemográficos del estudiante, su desempeño en la escuela secundaria y su desarrollo en los primeros semestres de la Universidad. Estas preguntas combinan factores académicos, laborales, sociales y extracurriculares, que influyen en la decisión de abandonar la Universidad [3]. Inicialmente se eligieron 23 preguntas y tras una limpieza de datos se eliminaron 2 preguntas. Las variables y atributos aplicados en la encuesta se describen en el Anexo 1.

De las 21 preguntas seleccionadas, 6 son cuantitativas, 3 cualitativas ordinales y 12 cualitativas no ordinales. Esta distinción influye al realizar el tratamiento de los datos antes del análisis en el software.

3.4 Tratamiento de datos y normalización

Tenemos dos opciones para simplificar los rangos de las variables, puesto que hay variables como la nota de ingreso a la Universidad cuyos mínimos y máximos van desde 410 a 1000 respectivamente. Se decide normalizar los datos, es decir establecer el rango de cada variable entre 0 y 1.

Para las variables cuantitativas se ha aplicado la fórmula:

$$x_{norm} = \frac{x - x_{mín}}{x_{máx} - x_{mín}} \quad (4)$$

donde:

x_{norm} : dato normalizado

x : dato sin normalizar

$x_{mín}$: dato mínimo de la variable

$x_{máx}$: dato máximo de la variable

Estas variables son: Edad, cantidad de hermanos, nota de grado del colegio, edad de ingreso a la Universidad, nota de ingreso a la Universidad y promedio con que se finalizó el primer año de Universidad.

Para los datos cualitativos ordinales se asignó valores equidistantes entre 0 y 1. Estas variables son: Estrato socioeconómico, lugar que ocupa entre sus hermanos y satisfacción con el programa.

Por último, a las variables cualitativas no ordinales se les aplicó el método *One Hot Encoding* lo que provocó que cada variable se despliegue en su cantidad de atributos respectiva [25], ampliando notablemente la cantidad de parámetros de cada respuesta. Un ejemplo de aplicación de este tipo de codificación de muestra en la Figura 5. Al final cada fila se compone de 57 dimensiones, 56 entradas y 1 salida.

		Ciudad de residencia			
	Ciudad de residencia	Ambato	Quito	Riobamba	Guayaquil
1	Ambato	1	0	0	0
2	Quito	0	1	0	0
3	Riobamba	0	0	1	0
4	Guayaquil	0	0	0	1
5	Ambato	1	0	0	0
6	Guayaquil	0	0	0	1

Figura 5. Adecuación de una variable categórica con método *One Hot Encoding*. Fuente: Propia

En este ejemplo tenemos una categoría llamada “Ciudad de residencia” cuyos atributos no se pueden ubicar de forma ordinal, el método convierte esta variable de una columna con x atributos en x categorías de dos atributos cada una: uno (1) y cero (0).

Por último, en el tratamiento de datos se ha elegido la variable de deserción como predictora y se han asignado las etiquetas: 0 para “Nunca he abandonado mi carrera” y 1 para “Abandoné mi carrera en algún momento”. El último paso del tratamiento de datos consistió en adecuarlos para la lectura del software.

3.5 Adaptación del código

Mediante el software especializado utilizamos una plantilla para la aplicación de Redes Neuronales Artificiales. Debido a la poca cantidad de datos disponibles es necesario aplicar un balanceo de datos, de manera que exista equilibrio en la cantidad de datos de las clases 0 y 1. Cabe recalcar que al tener 56 entradas el análisis es bastante complejo y demanda de más datos de los que tenemos disponibles.

La codificación que usaremos realiza los siguientes pasos:

1. Carga de librería.
2. Lectura de datos de entrada y salida.
3. División de datos de entrada de cada clase
4. División de datos para entrenamiento, validación y testeo
5. Estimación del error de validación usando validación cruzada y balanceo
6. Entrenamiento del clasificador para datos desbalanceados
7. Entrenamiento del clasificador para datos balanceados
8. Entrenamiento y testeo del modelo final desbalanceado
9. Entrenamiento y testeo del modelo final balanceado

En primer lugar, cargamos la librería para redes neuronales artificiales. Luego, en la lectura de entradas y salidas describimos la cantidad de variables involucradas (56 para entradas y 1 para salidas), los nombres y tipos de las mismas y la fuente desde la que las extraeremos.

Para la división de datos en grupos de entrenamiento y validación, y datos de testeo debemos fijar el porcentaje de datos que se utilizarán para cada grupo. El rango de este valor está entre 0 y 1, siendo un valor bastante común el de 0,7 y se definirá tras hacer pruebas al buscar la mejor arquitectura. En este paso se toman valores aleatorios para conformar cada bloque, cuidando así que no hay sesgos entre los datos.

Otro parámetro que fijaremos es el de la cantidad de capas en que dividiremos los datos para la validación cruzada. Para la cantidad de datos que tenemos la cantidad de capas no puede ser mayor a 47. Este valor corresponde a la cantidad máxima de datos con etiqueta 1.

En este paso también conformaremos la arquitectura del modelo. Definiremos la cantidad de capas (principal y ocultas), la cantidad de neuronas por cada capa, la función de transferencia de cada capa, el factor de regularización por decaimiento de pesos lambda, la cantidad de iteraciones en cada corrida y la cantidad de corridas del modelo.

Con los datos de error de entrenamiento buscaremos una configuración que nos arroje los mejores valores de precisión del modelo, especificidad y sensibilidad y valor F del modelo, tanto para datos desbalanceados y balanceados. Por último, al haber encontrado la mejor arquitectura podemos generar corridas del modelo para ver resultados finales. En últimas instancias podremos observar la matriz de confusión y puntajes finales del modelo.

3.6 Balanceo de datos

La problemática de este proyecto es la deficiente cantidad de datos frente a la cantidad de dimensiones que se tratarán. Además, ocurre que la clase 1, desertores, tuvo pocas respuestas en la primera recolección de datos comparados con los datos de no desertores. A esto llamamos datos desbalanceados, en los que una clase tiene una cantidad predominante de datos frente a otra. En la Figura 6 podemos ver la cantidad de datos recogidos para cada clase tras la primera encuesta.



Figura 6. Datos para las clases 0 y 1 tras la primera recolección de datos.

Fuente: Propia.

Inicialmente la proporción entre datos de clase 0 a clase 1 se aproximaba a cinco por cada uno. Al realizar pruebas de entrenamiento iniciales los resultados eran bastante deficientes sobre todo para la sensibilidad. Como se aprecia en la Tabla 4 los porcentajes de especificidad y sensibilidad eran en promedio del 93% y 38% respectivamente.

Tabla 4. Mejores métricas con primera recogida de datos. Fuente: Propia

fraction	num folds	Neuronas por capa		función	Lambda	Accuracy	Specificity	Sensitivity	Score Model
		Capa 1	Capa 2						
0,7	10	20		tanh	0	0,8769	0,9382	0,45	0,3792
0,7	10	20		tanh	0	0,8831	0,9444	0,33	0,4795
0,7	10	30		tanh	0	0,8631	0,9444	0,33	0,3128
0,7	10	40		tanh	0	0,8715	0,9208	0,42	0,4786

Para buscar mejores resultados se efectuó una segunda encuesta procurando focalizarla en datos de clase 1. La relación de los nuevos datos se observa en la Figura 7.

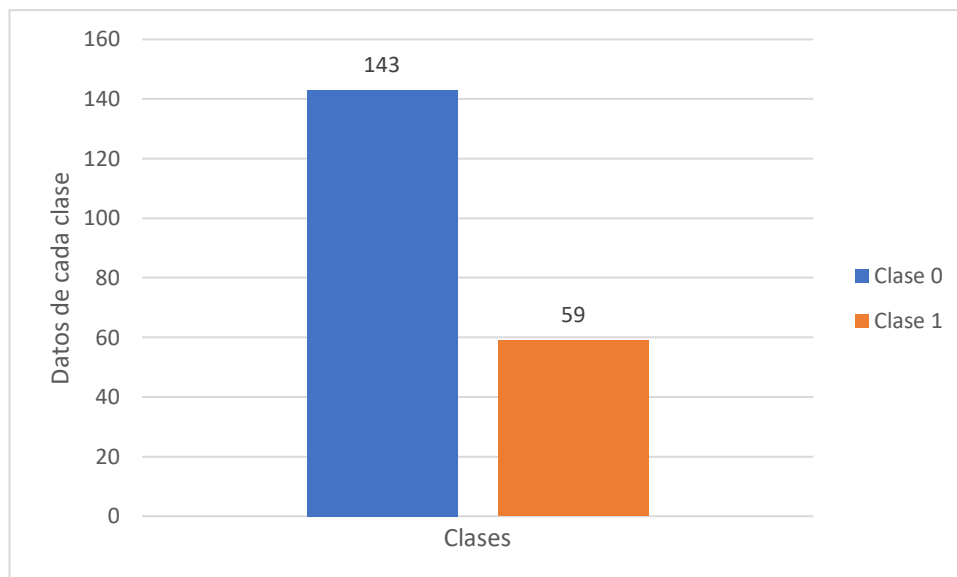


Figura 7. Datos para las clases 0 y 1 tras la segunda recolección de datos.

Fuente: Propia

Se logra una mejora significativa en la sensibilidad en el tratamiento con los nuevos datos. En el apartado de resultados se describen las métricas obtenidas tras la segunda encuesta.

3.7 Submuestreo y sobremuestreo de datos

Con el nuevo contingente de datos procedemos a balancear los datos, para lo cuál se tienen dos opciones iniciales. El submuestreo buscaría reducir los datos de la clase 0 para igualarlos con los datos de clase 1. Las desventajas de este proceso son que se pierde información de la data predominante y que el modelo podría verse sesgado.

Al contrario, al hacer un sobremuestreo provocamos que la muestra minoritaria, para la clase 1, replique sus datos para alcanzar la cantidad de datos de la muestra predominante. La desventaja de esta estrategia es que exista un sobreequipamiento de data. En el apartado de resultados se detalla el método utilizado en este proyecto.

3.8 Arquitectura del modelo

Tras la comprensión del código buscamos los valores de los parámetros que dan mejor resultados en el modelo. Esto lo hacemos sin incluir en el código el entrenamiento final del modelo, sino que lo ejecutaremos hasta cuando estimamos el error del mismo y hacemos el entrenamiento inicial. Esto lo hacemos con el objetivo de no obtener un criterio sesgado de los mejores valores para la arquitectura. Los parámetros que están sujetos a variación son:

- Fracción para testeo y validación
- Número de capas para validación cruzada
- Cantidad de capas ocultas
- Cantidad de neuronas por capa
- Función de transferencia de cada capa
- Factor de regularización de decaimiento de pesos (λ)
- Número de épocas para el entrenamiento de red
- Número de corridas para el modelo

Inicialmente buscamos que ciertos valores que cuantifican el modelo sean los más altos posibles, estos son la precisión del modelo, sensibilidad y especificidad. Arrancamos iterando la fracción de testeo y validación con valores de 0,6, 0,7 y 0,8; el número de capas para validación cruzada, con valores de 10, 20 y 30; la cantidad de neuronas por capa, con valores de 10, 50, 100 y 1000; la cantidad de capas ocultas y las funciones de transferencia de estas capas con funciones tangencial y relu.

En primera instancia las mejores configuraciones arrojaban un puntaje del modelo cercano al 0,42 que es bastante deficiente para el cumplimiento de objetivos que buscamos. La especificidad del modelo se situaba en el 0,90, mientras que la sensibilidad el 0,45, lo que implica que el modelo no logra clasificar a desertores con precisión. Por este problema se establece la posibilidad de incrementar la cantidad de respuestas con datos de desertores, y así tener una mejor base para la clasificación.

3.9 Métricas a evaluar en el modelo

Para calcular qué tan bien clasifica el modelo a un estudiante en riesgo de deserción utilizaremos la sensibilidad y especificidad. Lo ideal es que ambos valores sean lo más altos posible.

La sensibilidad ha sido problemática en este estudio debido a ser la clase minoritaria. Esta nos indicará los verdaderos positivos, esto es las personas en riesgo de deserción clasificadas como tal.

La especificidad tuvo valores bastante favorables desde las primeras pruebas. Esta nos indica la cantidad de verdaderos falsos, es decir estudiantes que no están en peligro de deserción y clasificados como tal.

El *accuracy* nos sirve para evaluar el total de datos clasificados correctamente de los 41 datos aplicados en el testeo.

3.10 Validación cruzada

Con el objetivo de evitar sesgos en caso de que un grupo de datos tanto el entrenamiento como el testeo esté sobrecargado con datos de la clase 0 o la clase 1 se aplican estrategias para validar el entrenamiento del modelo y aplicar luego el testeo. La *cross validation* o validación cruzada consiste en dividir los datos del modelo en k grupos. A partir de esta división se formarán combinaciones en las que se dejará un grupo fuera para el testeo y el resto de grupos ($k-1$) se utilizarán para el entrenamiento y validación [26]. El modelo elegirá la combinación que mejor generalice la clasificación.

Por ejemplo, podemos elegir dividir los datos en grupos $k=5$. En cada ronda se combinarán 4 grupos para el entrenamiento y 1 para la validación. En este caso se efectúan 5 modelos posibles y se hará una media ponderada de los errores obtenidos para definir el error del modelo, como se muestra en la Figura 8.

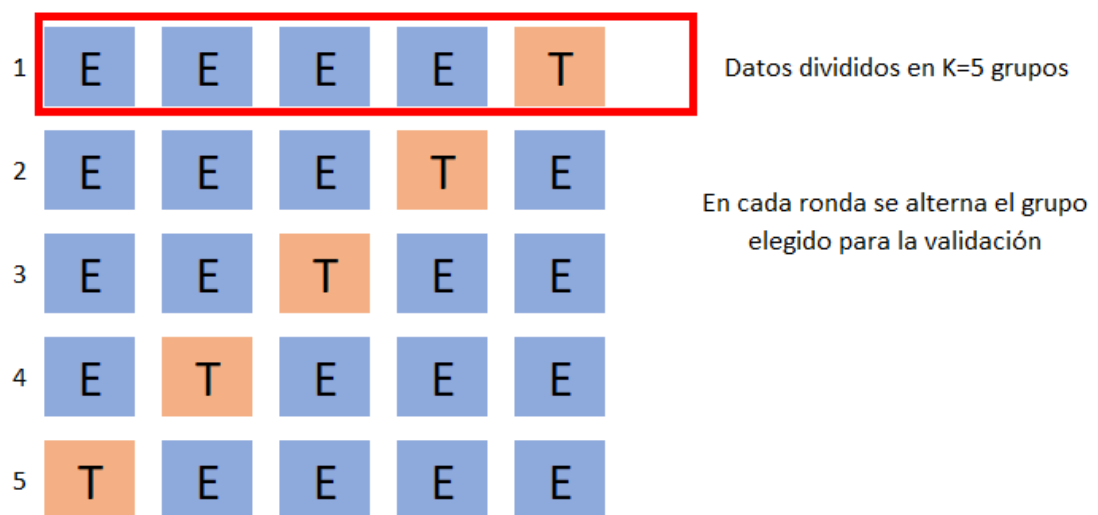


Figura 8. Ejemplo de validación cruzada. Fuente: Propia

CAPÍTULO IV.

RESULTADOS Y DISCUSIÓN

En este apartado se detalla el proceso por el que pasan los datos de inicio a fin. También se describe el procedimiento para encontrar la mejor arquitectura y los datos obtenidos con el modelo de RNA haciendo énfasis en los promedios de métricas de evaluación del mismo. Se especifica el proceso completo aplicado a los datos recolectados. Se detalla la razón de haber elegido la especificidad y sensibilidad para medir la eficiencia del modelo. También se hará una comparación de los resultados obtenidos con datos desbalanceados y balanceados.

4.1 Procesamiento de datos

Los datos desde su recolección hasta la generación del modelo siguen el proceso descrito en la Figura 9.

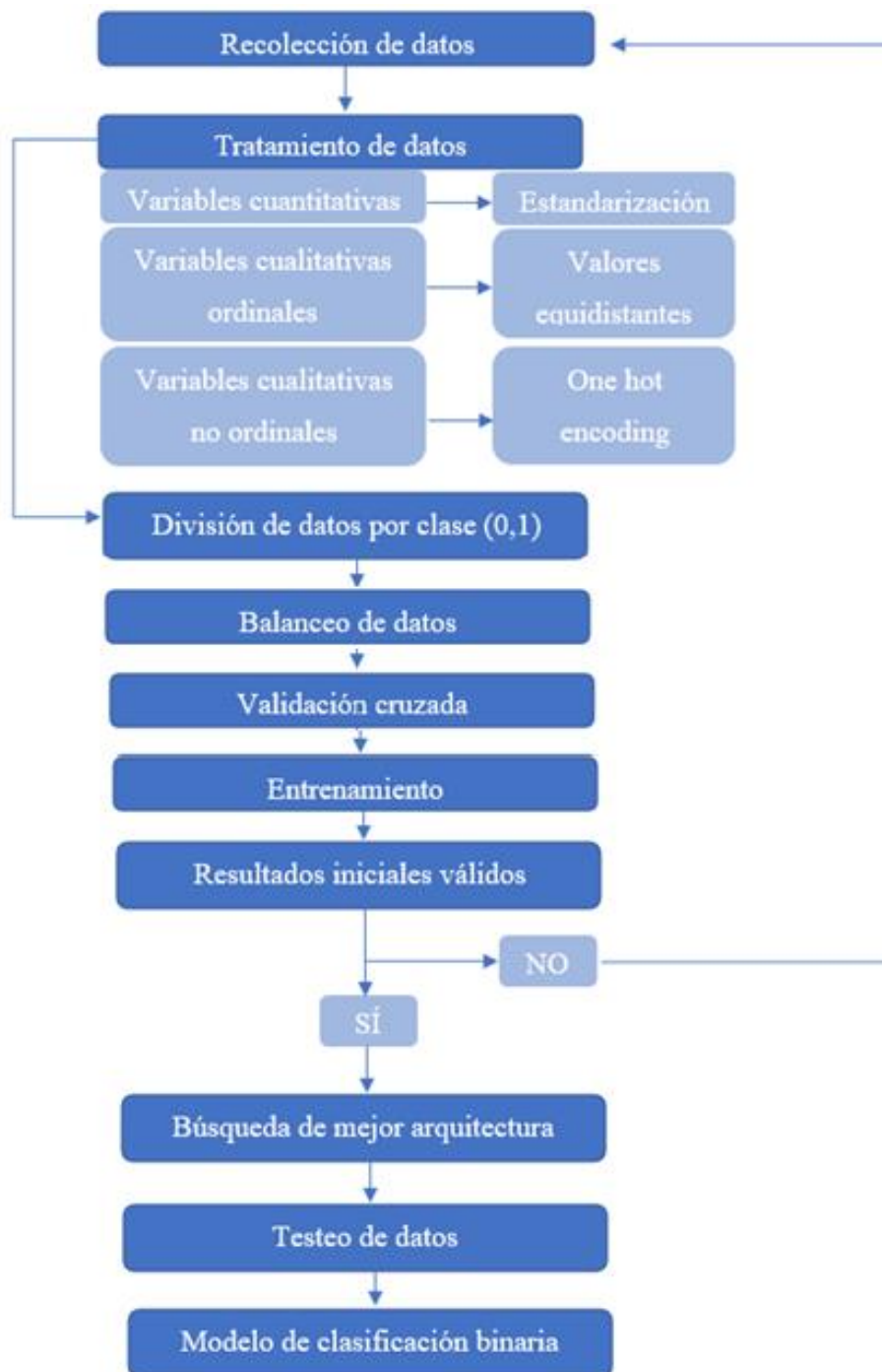


Figura 9. Esquema de procesamiento de datos. Fuente: Propia.

4.2 Selección de variables predictoras o respuesta

En primera instancia se pensaron las variables con que se tratará de estimar la deserción estudiantil. Se creó un formulario con 22 variables y sus atributos respectivos. Estas categorías y su delimitación como variables predictoras o de respuesta se detallan en la Tabla 5.

Tabla 5. Selección de categorías del estudio. Fuente: Propia

Categoría	Atributos	Tipo
Edad	Edad	Predictora
Género	Masculino, Femenino	Predictora
Estrato socioeconómico	Estrato	Predictora
¿Cuántos hermanos tiene?	Hermanos	Predictora
Lugar que ocupa entre los hermanos	Lugar	Predictora
Escolaridad de la madre	Esc. Madre	Predictora
Escolaridad del padre	Esc. Padre	Predictora
Tipo de colegio	Público, Fiscomisional, Privado	Predictora
¿Con qué nota se graduó del colegio?	Nota Grado	Predictora
¿A qué edad entró a la Universidad?	Edad Ingreso	Predictora
Recibió Orientación Vocacional	OV	Predictora
Certeza al elegir la carrera	Certeza	Predictora
¿Con qué puntaje sobre 1000 ingresó a la Universidad?	Puntaje Admisión	Predictora
Tipo de universidad	U. Pública, U. Privada	Predictora
Estado civil	Soltero, Casado, Unión libe, Divorciado, Viudo	Predictora
Con quiénes vivía	Convive	Predictora
Apoyo de la familia por la carrera	Apoyo	Predictora
Aporte económico al hogar	Aporte	Predictora
Ocupaciones simultáneas	Extracurricular	Predictora
Satisfacción con el programa	Satisfacción	Predictora
Nota de primer semestre sobre 10	Promedio	Predictora
Característica de deserción estudiantil	Nunca he abandonado, He abandona en algún momento	Respuesta

4.3 Selección de método para balanceo de datos

Analizaremos qué técnica de balanceo se utilizará. En la Figura 10 observamos lo que hace un código para equilibrar las muestras de datos. Recordemos que la clase

predominante es la clase 0, los no desertores, incluso tras la recopilación de nuevos datos.

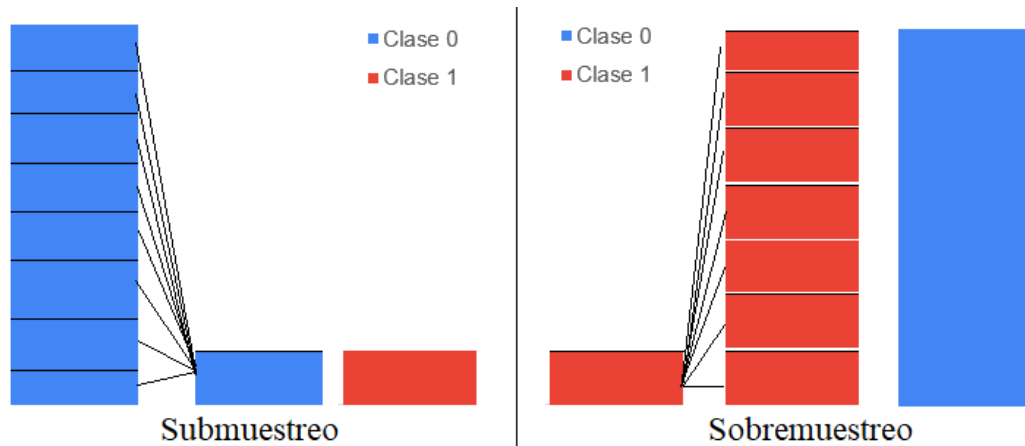


Figura 10. Esquema de dos técnicas de balanceo de datos. Fuente: Propia

Un problema recurrente de este proyecto es la poca cantidad de datos, por ello no sería razonable usar un submuestreo, que menguaría aún más el conjunto de datos. Optamos por usar un sobremuestreo. Lo que esperamos de la técnica de *oversampling* es que podamos obtener mejores resultados al clasificar a desertores sin que se afecte la clasificación de no desertores.

4.4 Obtención de la mejor arquitectura

Con los nuevos datos se estableció una estrategia de iteraciones entre valores exponenciales primero y luego refinaremos estos resultados. Para hacer el refinamiento variaremos los parámetros con valores más próximos entre sí. Haciendo combinaciones entre los diferentes parámetros, hacemos pruebas del modelo y tomamos nota de las puntuaciones del mismo hasta encontrar los resultados más favorables.

Se realizaron pruebas alternando la cantidad de neuronas en 10, 50, 100 y 1000. Se iteró la cantidad de capas ocultas en una, dos y hasta tres. Además se combinan estos parámetros con una fracción para entrenamiento y testeo de 0,6, 0,7, 0,8 y 0,9.

La primera generalización que se notó es que con la fracción de testeo de 0.8 y una sola capa oculta se obtienen mejores resultados. En esta instancia empezamos a hacer iteraciones más finas con los parámetros de cantidad de neuronas y función de transferencia.

La configuración idónea junto con los valores para calificación del modelo se muestra en la Tabla 6:

Tabla 6. Mejor arquitectura para el modelo. Fuente: Propia

fraction	num folds	Neuronas por capa			función	Lambda
		Capa 1	Capa 2	Capa 3		
0,8	20	80	-	-	tanh	0
0,8	20	90	-	-	tanh	0
0,8	20	100	-	-	tanh	0

Con una fracción de datos para validación y testeo de 0.8, una sola capa con 100 neuronas, función tangencial y un valor lambda de 0 se obtuvieron los resultados mostrados en las Tablas 7 y 8, para *Accuracy*, Especificidad, Sensibilidad y *F1 Score*, para datos desbalanceados y balanceados.

Tabla 7. Métricas para datos desbalanceados con la mejor arquitectura.

Datos no balanceados			
Accuracy	Specificity	Sensitivity	Score Model
0,7911	0,8621	0,6139	0,5968
0,7304	0,7995	0,5583	0,5432
0,7679	0,86	0,5417	0,5267

Tabla 8. Métricas para datos balanceados con la mejor arquitectura.

Datos balanceados			
Accuracy	Specificity	Sensitivity	Score Model
0,7696	0,8421	0,5889	0,5585
0,7625	0,8247	0,6083	0,6086
0,7929	0,85	0,6528	0,616

Una vez conocida la arquitectura con mejores resultados se completará el código para ejecutar ahora sí el entrenamiento final y testeo del modelo.

4.5 Resultados del modelo y matriz de confusión

Con la distribución de 20 capas definida en la arquitectura los 202 datos se distribuyen 161 para el entrenamiento y 41 para el testeo. Con el entrenamiento obtenido estos 41 datos son clasificados con la etiqueta 0 cuando están por debajo del umbral 0.5, caso contrario con la etiqueta 1, que implica riesgo de deserción ($P \geq 0.5$).

La matriz de confusión nos muestra cuántos de los datos separados como desertores (1) o no desertores (0) han sido clasificados de forma correcta o incorrecta. Un mapa de la matriz y un ejemplo se muestran en la Figura 11.

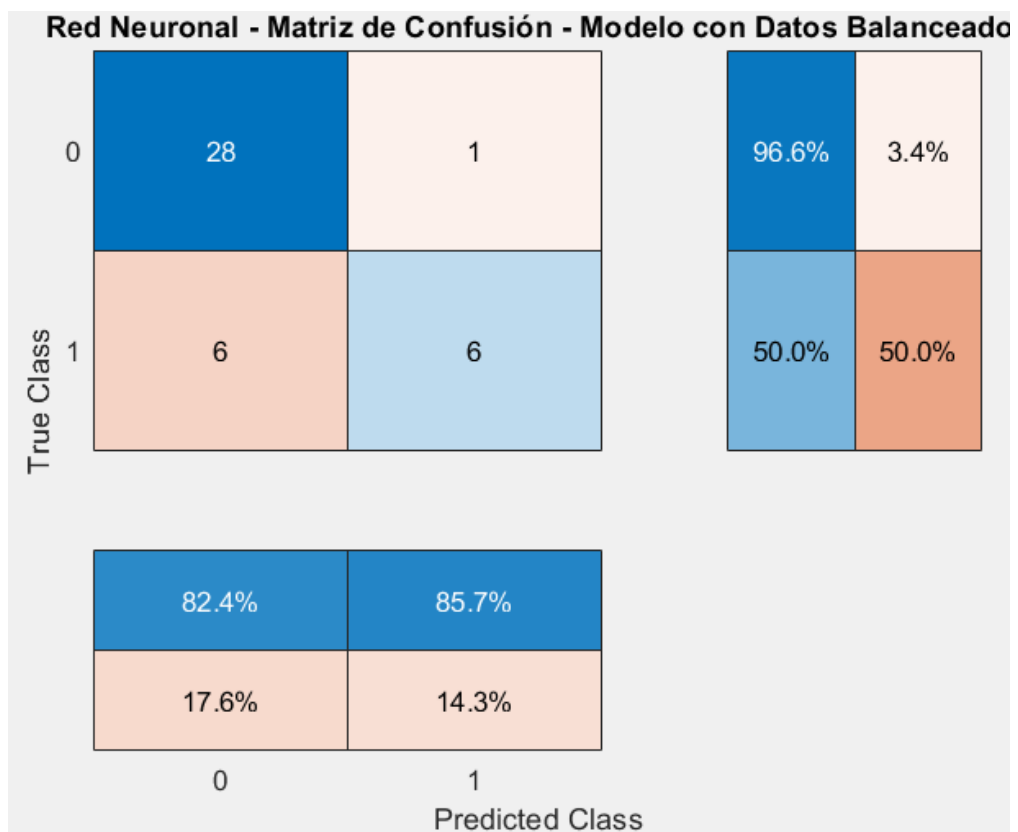
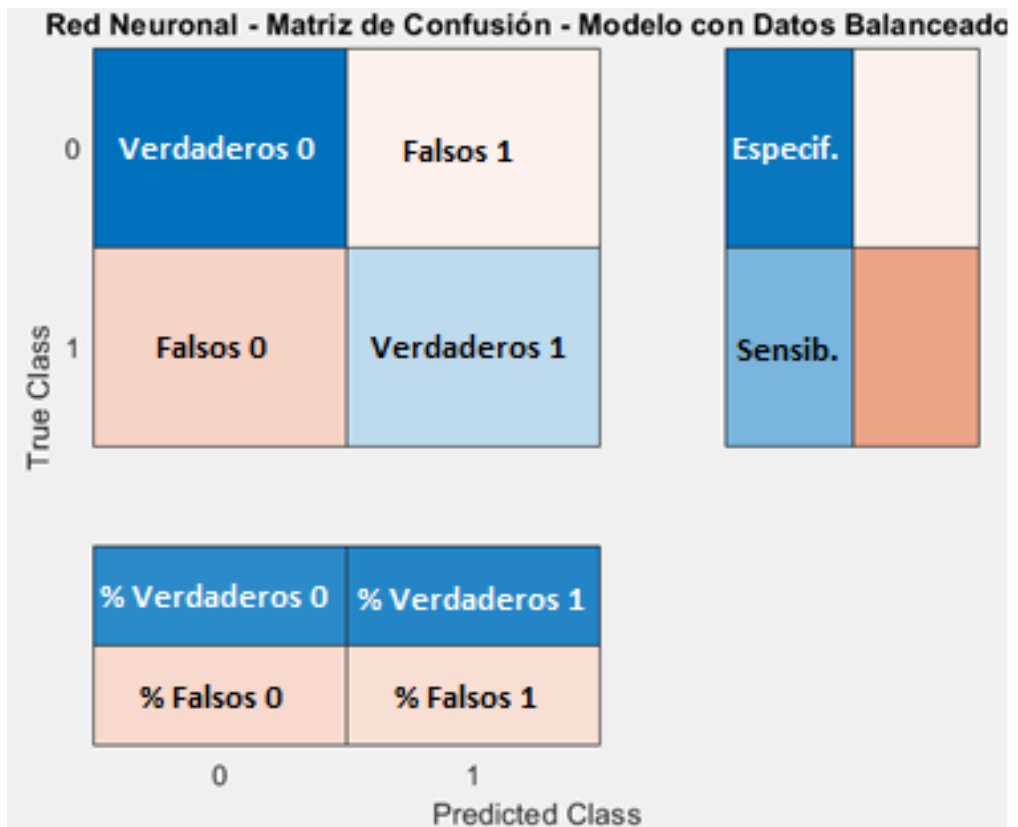


Figura 11. Representación de matriz de confusión y ejemplo. Fuente: Propia

En la primera fila observamos los datos que realmente corresponden a no desertores y en la segunda a la de desertores, mientras que en las columnas conocemos cómo el modelo ha clasificado los datos, en la primera columna como no desertores y en la segunda como desertores, a partir del ejemplo, podemos ver que tenemos: 0's clasificados como 0's: 28 (verdaderos)

0's clasificados como 1's: 1 (falsos)

1's clasificados como 0's: 6 (falsos)

1's clasificados como 1's: 6 (verdaderos)

A partir de ello también podemos observar los valores de especificidad y sensibilidad. El porcentaje de verdaderos 0's en la primera fila nos indica la especificidad, mientras que la sensibilidad será el porcentaje de verdaderos 1's en la segunda fila. Para el ejemplo tendremos:

Especificidad: 96,6%, esto nos indica que el modelo clasificará a un no desertor de forma correcta 97 de cada 100 veces que probemos el modelo.

Sensibilidad: 50%, esto nos indica que el modelo clasificará a un desertor de forma correcta la mitad de las veces.

4.6 Capas de la validación cruzada

Para la división de los datos en capas se probaron 10, 20 y 30 capas. Se considerará que el máximo de capas en que podemos dividir los datos es de 47 puesto que al aplicar una fracción para entrenamiento y validación, y testeo de 0,8, la cantidad de datos de clase 1 disponible será de 47. La clase minoritaria 1 define el máximo de capas que podemos aplicar en la validación cruzada.

Durante las pruebas observamos que con 10 capas la especificidad del modelo tiende a disminuir. Cuando trabajamos con capas a partir de 30 en cambio se observa que no hay mejoría entre datos balanceados y desbalanceados y que en ocasiones empeoran. Finalmente, se decide usar 20 capas para la validación cruzada.

4.7 Métricas de evaluación de rendimiento

Con el modelo logrado realizamos 10 pruebas y recogemos los valores de las métricas elegidas y se han promediado los resultados. Estas pruebas completas se resumen en

el Anexo 2. Se han extraído los valores para *Accuracy*, *Specificity*, *Sensitivity* y *F1 Score* separados en cuatro estratos: Entrenamiento con datos desbalanceados, entrenamiento con datos balanceados, testeo con datos desbalanceados y testeo con datos balanceados.

Los resultados para entrenamiento con datos desbalanceados (Tabla 9) y datos balanceados (Tabla 10) no presentan una variación significativa.

Tabla 9. Métricas de entrenamiento del modelo para datos desbalanceados

Prueba	Accuracy	Specificity	Sensitivity	F1 Score
Prom.	0.70089	0.79269	0.47334	0.43691
Desv.	0,024	0,020	0,049	0,046

Tabla 10. Métricas de entrenamiento del modelo para datos balanceados

Prueba	Accuracy	Specificity	Sensitivity	F1 Score
Prom.	0,68751	0,77242	0,47694	0,44039
Desv.	0,017	0,014	0,050	0,049

Para el entrenamiento de datos el mejor resultado en precisión fue de 73% y ocurrió con datos desbalanceados. En promedio el entrenamiento con datos desbalanceados tuvo mejores resultados con un 70% frente al 68% obtenido con datos balanceados.

La especificidad también tuvo mejores resultados con datos desbalanceados, con un 79% frente a un 77%. La sensibilidad es virtualmente igual para ambos grupos de datos, levemente mejor para balanceados con un 47,7% frente a un 47,3%. El mejor resultado de sensibilidad es de 54,7% con datos balanceados.

Por último el *F1 Score* es también ligeramente mejor en datos balanceados, con un 44% frente a un 43,7%. El mejor resultado de *F1 Score* es de 51,2% y se obtuvo con datos balanceados frente a un 49,9% con datos desbalanceados.

Luego, al observar las métricas de testeo con datos desbalanceados (Tabla 11) y datos balanceados (Tabla 12) muestran mejores valores que en el entrenamiento.

Tabla 11. Métricas del testeo para datos desbalanceados.

Métricas del testeo para datos desbalanceados				
Prueba	Accuracy	Specificity	Sensitivity	F1 Score
Prom.	0,77562	0,88623	0,50833	0,56715
Desv.	0,054	0,056	0,121	0,113

Tabla 12. Métricas del testeo para datos balanceados.

Métricas del testeo para datos balanceados				
Prueba	Accuracy	Specificity	Sensitivity	F1 Score
Prom.	0,79026	0,86898	0,6	0,62564
Desv.	0,035	0,063	0,086	0,048

En el testeo de los modelos observamos que justamente aquellos con datos balanceados tienen un mejor rendimiento que los de datos desbalanceados, con un 79,02% de *Accuracy* a pesar de que el valor más alto fue de 85,4% con datos desbalanceados.

En cuanto a la especificidad, se obtienen mejores resultados con datos desbalanceados con un 88,6% frente al 86,8% para datos balanceados. El mejor resultado de especificidad fue del 100% en una de las pruebas con datos desbalanceados (se puede apreciar en el Anexo 2). Mientras que la sensibilidad mejora notablemente con datos balanceados con un promedio de 60% frente a un 50,8% para datos desbalanceados. El mejor resultado para sensibilidad fue de 75% con datos balanceados.

El *Score Model* presenta una mejora significativa con datos balanceados, en promedio se tiene un puntaje de 0,63 frente a un 0,57 con datos desbalanceados. Este valor es relativamente bajo en precisión y se puede explicar como consecuencia de la complejidad del modelo y la cantidad insuficiente de datos obtenidos.

Adicionalmente también analizamos la matriz de confusión de las 10 pruebas tomadas. A partir de las mismas podemos evaluar la precisión del modelo en la clasificación de los 41 datos utilizados en el testeo. Las matrices de confusión para los datos balanceados y desbalanceados, además de un resumen de la cantidad de datos clasificados en cada tipo en las pruebas realizadas se puede observar en el Anexo 3.

En la Tabla 13 y Tabla 14 podemos observar un esquema de cómo clasifica los datos el modelo de los 41 datos que usa para testeo.

Tabla 13. Clasificación en el testeo para datos desbalanceados

Clasificación de datos de testeo con datos desbalanceados					
Prueba	Verdaderos 0	Falsos 0	Falsos 1	Verdaderos 1	Total Verdaderos
Prom.	25,7	3,3	5,9	6,1	31,8
Desv.	1,636	1,636	1,449	1,449	2,201

Tabla 14. Clasificación en el testeo para datos balanceados.

Clasificación de datos de testeo con datos balanceados					
Prueba	Verdaderos 0	Falsos 0	Falsos 1	Verdaderos 1	Total Verdaderos
Prom.	25,2	3,9	5,1	6,8	32
Desv.	1,814	2,025	1,197	1,398	2,404

En promedio el modelo logra clasificar de forma correcta 25 datos de no desertores y 7 de desertores, mientras que falla en 4 de no desertores y 5 de desertores. En total logra clasificar de forma correcta en promedio 32 de los datos (78%) y falla en 9 (22%) de ellos. El mejor resultado de datos bien clasificados en total fue de 35 y ocurrió con datos no balanceados. Para datos balanceados el mejor resultado fue de 34 datos bien clasificados ocurrido en varias de las pruebas.

Se recalca que en la séptima prueba el modelo logró clasificar los 29 datos de no desertores de forma correcta, siendo este el mejor resultado para verdaderos 0's. Es interesante que este resultado se logró con datos desbalanceados. De entre los datos balanceados, la mejor clasificación de verdaderos 0's fue de 28.

Para los verdaderos 1's el mejor resultado fue de 9 valores bien clasificados con datos balanceados, mientras que con datos desbalanceados el mejor resultado fue de 8 datos en varias ocasiones.

4.8 Comparación de datos desbalanceados y balanceados

Se buscó comparar las métricas entre datos desbalanceados y balanceados. Por lo general con datos balanceados se deberían observar mejores resultados. Para este

proyecto en varios casos se obtuvieron mejores métricas con datos desbalanceados, sobre todo en el entrenamiento. La Tabla 15 resume una comparación del promedio de las métricas obtenidas para datos desbalanceados y balanceados. En color verde se resalta el valor más alto de cada métrica y se menciona también la diferencia entre los datos:

Tabla 15. Comparación de métricas entre datos desbalanceados y balanceados.

Métrica	Con datos desbalanceados	Con datos balanceados	Diferencia
Para entrenamiento			
Acurracy	0,70	0,688	0,012
Especificidad	0,793	0,772	0,021
Sensibilidad	0,473	0,477	0,004
F1 Score	0,437	0,44	0,003
Para testeo			
Acurracy	0,776	0,79	0,014
Especificidad	0,886	0,869	0,017
Sensibilidad	0,508	0,6	0,092
F1 Score	0,567	0,626	0,059

Para el entrenamiento con datos desbalanceados la precisión y especificidad del modelo fueron mejores en un 1,2% y 2,1% respectivamente, la sensibilidad y el *F1 Score* fueron relativamente iguales. En cambio, para el testeo de datos la precisión fue mejor con datos balanceados en un 1,4%, la especificidad mejor con datos desbalanceados en un 1,7%. La diferencia más notable ocurre con la sensibilidad que con datos balanceados mejora en un 9,2%. El *Score* también mejora notablemente con datos balanceados en un 5,9%.

4.9 Pruebas con disminución de variables de entrada

Para comprobar si la gran cantidad de variables de entrada influye en la eficiencia del modelo haremos nuevas pruebas disminuyendo variables. Se decidió eliminar las variables que tras la codificación por *One Hot Encoding* aumentaban en mayor medida las dimensiones del modelo. En las Tablas 16 y 17 se observan las métricas promedio

para el testeo del modelo usando solamente 17 variables de entrada y 10 variables de entrada respectivamente.

Tabla 16. Métricas de testeo usando 17 variables de entrada.

Prueba	Datos desbalanceados				Datos balanceados			
	Accuracy	Specificity	Sensitivity	F1 Score	Accuracy	Specificity	Sensitivity	F1 Score
Prom.	0,72438	0,84483	0,43334	0,4748	0,78538	0,81725	0,70834	0,66033
Desv.	0,049	0,086	0,110	0,089	0,036	0,075	0,071	0,029

Tabla 17. Métricas de testeo usando 10 variables de entrada.

Prueba	Datos desbalanceados				Datos balanceados			
	Accuracy	Specificity	Sensitivity	F1 Score	Accuracy	Specificity	Sensitivity	F1 Score
Prom.	0,75366	0,90001	0,40001	0,49047	0,77562	0,78275	0,75833	0,66466
Desv.	0,051	0,062	0,053	0,075	0,034	0,065	0,083	0,039

CAPÍTULO V.

CONCLUSIONES Y REFERENCIAS BIBLIOGRÁFICAS

En el último apartado se comentan observaciones finales de este proyecto. Las conclusiones abarcarán qué tan eficiente consideramos al modelo y las causas de haber obtenido las métricas descritas en resultados. Se detallará ciertas consideraciones que se deberían tener en cuenta al hacer un proyecto similar a partir de problemáticas que surgieron en este. Por último se mencionan observaciones interesantes y hallazgos que no eran objetivo inicial del proyecto pero que han surgido durante la investigación.

5.1 Conclusiones

Mediante este proyecto se logró crear un modelo con redes neuronales que clasifica un estudiante bajo el caso de no estar en riesgo de deserción o de sí estarlo. Para los mejores resultados, el 97% de las veces un alumno que no está en riesgo de deserción será catalogado como tal (especificidad) y el 75% de alumnos en riesgo de deserción obtendrán esta etiqueta (sensibilidad). Luego, en promedio el modelo clasificará un no desertor de forma correcta el 86% de las veces y a un desertor de forma correcta el 60% de las veces. La exactitud del modelo fue del 79% y tuvo un *F1-Score* de 0,62.

En esta investigación se determinó que el analizar un mayor número de variables de entrada no implica que nuestra predicción será mejor, sino más bien lo contrario. El modelo original utiliza 56 variables. Se hicieron pruebas eliminando algunas variables de entrada hasta tener 17 y 10, aplicando la misma arquitectura que en el modelo original. Al utilizar 17 variables de entrada se obtuvo una exactitud de 79%, especificidad de 82%, sensibilidad de 70% y un *F1 Score* de 0,66. Con 10 variables se obtuvo una exactitud de 78%, especificidad de 78%, sensibilidad de 75% y un *F1 Score* de 0,66. En ambos casos el modelo mejora. Es curioso que al disminuir variables el modelo es más eficiente clasificando desertores pero menos hábil para clasificar no desertores. Mientras menos variables se utilizaron en el modelo, la especificidad disminuyó (hasta en un 8%) mientras que la sensibilidad aumentó (hasta en un 15%). Las variables finales elegidas fueron el estrato económico, la cantidad de hermanos, tipo de colegio, nota de grado, proceso de Orientación Vocacional, puntaje de admisión a la Universidad y tipo de Universidad.

El primer dataset estaba desbalanceado debido a la predominancia de datos de no desertores. Con este dataset, la especificidad en el entrenamiento llegó a ser de un

promedio de 92%, mientras que la sensibilidad alcanzaba un promedio de 41%. Tras una segunda recopilación enfocada en nuevos datos de la clase 1 mejoraron estas métricas. Aplicando sobremuestreo la sensibilidad con los nuevos datos se elevó hasta el 60% (en promedio), que supone un aumento de cerca del 20% para clasificar de forma correcta a desertores. Concluimos que recopilar nuevos datos para una muestra más equilibrada junto con el balanceo de datos permite obtener mejores resultados.

Fue inesperado en un inicio descubrir que tanto en el entrenamiento como en el testeo se obtuvieron mejores métricas con datos desbalanceados algunas veces. Esta mejoría fue siempre en porcentajes inferiores o iguales al 2,1%. Sin embargo, el cambio más relevante ocurre con la sensibilidad en el testeo, que con datos balanceados mejora en promedio un 10%. Esta mejora significativa también incide en el *Score*. Las mejoras con datos desbalanceados no tienen mayor incidencia en el modelo. Las mejoras con datos balanceados ocurren sobre todo en la sensibilidad del modelo, que ha sido una métrica problemática para este proyecto. Por tanto, concluimos que los datos balanceados son naturalmente preferibles para este modelo.

La mejor arquitectura fue de 100 neuronas en una sola capa oculta y un factor de regularización de pesos de 0. Al aumentar las neuronas a 500 o a 1000 se obtuvieron resultados similares pero con un tiempo de ejecución superior. Por debajo de 80 neuronas los resultados no eran malos, pero sí inestables, con una diferencia entre valores máximo y mínimo de *F1-Score* de 15%. Adicionalmente, al aumentar la cantidad de capas el costo computacional aumenta considerablemente y los resultados son muy similares que con una sola capa.

Respecto a las funciones de activación utilizamos la función logística sigmoidea (logsig) en la capa final por ser la que mejor funciona para clasificación binaria. Para la capa oculta probamos las funciones logsig, tangencial hiperbólica (tanh) y la función de unidad lineal rectificadora (relu). A pesar de que teóricamente la función relu es más eficiente en la clasificación, para este modelo que tiene pocos datos ocurre que la sensibilidad disminuye (con datos balanceados) en cerca del 5%. Por lo tanto, para este trabajo la función que mejor funciona es tanh.

5.2 Recomendaciones

Para analizar deserción estudiantil, es beneficioso tener un modelo que logra identificar de forma correcta y temprana a posibles desertores. Debido a que se aplicó

un modelo de caja negra (Redes Neuronales Artificiales), no podemos concluir qué variables se deberían eliminar para tener mejores resultados. Para futuros proyectos se recomendaría aplicar un modelo de regresión lineal que permita identificar las variables que tienen poca incidencia en la predicción.

El método de codificación llamado *One Hot Encoding* a priori es la solución más común para trabajar con variables *dummy*, sin embargo provoca que las dimensiones aumenten notablemente. Se plantea para futuras investigaciones qué otros métodos pueden usarse para la codificación.

Para trabajos que como este tienen la problemática de la carencia de datos, lo más razonable es usar sobremuestreo para el balanceo de los datos. Usar submuestreo reduciría más los datos con que trabajaría el modelo.

Para un mejor estudio se entiende que en el caso de no poder conseguir suficientes datos se debería pensar en reducir las categorías que se quieren analizar. Se recomendaría repetir el proyecto usando solo variables de incidencia académica. Luego comparar solo factores sociales, luego solo demográficos, etc.

5.3 Referencias Bibliográficas

- [1] A. P. Ortiz, V. R. D. Pérez, and O. C. Salazar, “Una aproximación conceptual a la retención estudiantil en Latinoamérica.” *Revista Interamericana de Investigación, Educación y Pedagogía*, vol. 7, no. 2, 2014.
- [2] E. S. Fischer-Angulo, “MODELO PARA LA AUTOMATIZACIÓN DEL PROCESO DE DETERMINACIÓN DE RIESGO DE DESERCIÓN EN ESTUDIANTES UNIVERSITARIOS,” Universidad de Chile, Santiago de Chile, 2012.
- [3] T. Gonzalez and S. Alvarez, “Modelo para la evaluación del riesgo de deserción en la educación superior Model for the evaluation of the risk of dropping out in higher education.”
- [4] D. Post, “Las Reformas Constitucionales en el Ecuador y las Oportunidades para el Acceso a la Educación Superior desde 1950,” *EPAA*, vol. 19, 2010, [Online]. Available: <http://epaa.asu.edu/ojs/article/view/814>

- [5] T. Cardona, E. A. Cudney, R. Hoerl, and J. Snyder, “Data Mining and Machine Learning Retention Models in Higher Education,” *J Coll Stud Ret*, 2020, doi: 10.1177/1521025120964920.
- [6] S. C. Tsai, C. H. Chen, Y. T. Shiao, J. S. Ciou, and T. N. Wu, “Precision education with statistical learning and deep learning: a case study in Taiwan,” *International Journal of Educational Technology in Higher Education*, vol. 17, no. 1, Dec. 2020, doi: 10.1186/s41239-020-00186-2.
- [7] H. S. Alenezi and M. H. Faisal, “Utilizing crowdsourcing and machine learning in education: Literature review,” *Educ Inf Technol (Dordr)*, vol. 25, no. 4, pp. 2971–2986, Jul. 2020, doi: 10.1007/s10639-020-10102-w.
- [8] Y. Cui, F. Chen, A. Shiri, and Y. Fan, “Predictive analytic models of student success in higher education: A review of methodology,” *Information and Learning Science*, vol. 120, no. 3–4. Emerald Group Holdings Ltd., pp. 208–227, May 15, 2019. doi: 10.1108/ILS-10-2018-0104.
- [9] D. Delen, “A comparative analysis of machine learning techniques for student retention management,” *Decis Support Syst*, vol. 49, no. 4, pp. 498–506, May 2010, doi: 10.1016/j.dss.2010.06.003.
- [10] C. C. Gray and D. Perkins, “Utilizing early engagement and machine learning to predict student outcomes,” *Comput Educ*, vol. 131, pp. 22–32, Apr. 2019, doi: 10.1016/j.compedu.2018.12.006.
- [11] M. F. Musso, C. F. R. Hernández, and E. C. Cascallar, “Predicting key educational outcomes in academic trajectories: a machine-learning approach,” *High Educ (Dordr)*, vol. 80, no. 5, pp. 875–894, Nov. 2020, doi: 10.1007/s10734-020-00520-7.
- [12] J. Carlos González Sánchez and M. Javier Peñaloza Pérez, “IDENTIFICATION AND PREDICTION OF STUDENTS AT RISK OF ACADEMIC DROPOUT THROUGH MODELS BASED ON MACHINE LEARNING.”
- [13] R. T. Pereira, “Una lectura sobre deserción universitaria en estudiantes de pregrado desde la perspectiva de la minería de datos * An insight on university drop-out of undergraduate students from the perspective of data mining.”

- [14] M. Arias-Pérez, M. Bastidas-Ramos, and ; César Salazar-Mejía, “Estudio sobre la deserción estudiantil universitaria y sus implicaciones académicas, económicas y sociales,” Pág, 2018.
- [15] R. Florez-Lopez and J. M. Fernández, *Las redes neuronales artificiales fundamentos teóricos y aplicaciones prácticas*. La Coruña, 2008. Accessed: Feb. 12, 2023. [Online]. Available: <https://books.google.es/books?hl=es&lr=&id=X0uLwi1Ap4QC&oi=fnd&pg=PA11&dq=funciones+de+transferencia+redes+neuronales&ots=gOOCgsku0d&sig=q2LQLkXA1SFLz-nxc6RTLei8wGI#v=onepage&q&f=false>
- [16] E. Serna, *DESARROLLO E INNOVACIÓN EN INGENIERÍA.*, Segunda. Medellín, Antioquia, 2017.
- [17] IBM Corporation, “Establecer datos coherentes mediante estandarización,” Feb. 28, 2021.
- [18] S. Valero, “Transformación e interpretación de las puntuaciones,” Cataluña, 2013.
- [19] F. Izaurieta and C. Saavedra, “Redes Neuronales Artificiales,” Concepción, Chile, 2000.
- [20] R. Borja-Robalino, A. Monleon-Getino, A. Monleón-Getino, and J. Rodellar, “Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning,” Jun. 2020. [Online]. Available: <https://www.researchgate.net/publication/342009715>
- [21] K. Potdar, T. S., and C. D., “A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers,” *Int J Comput Appl*, vol. 175, no. 4, pp. 7–9, Oct. 2017, doi: 10.5120/ijca2017915495.
- [22] Secretaría Nacional de Planificación, “FICHA METODOLÓGICA PLAN NACIONAL DE DESARROLLO 2021-2025”.
- [23] A. Del, S. De, and R. Kazez, “LOS ESTUDIOS DE CASOS Y EL PROBLEMA DE LA SELECCION DE LA MUESTRA CASE STUDY AND THE PROBLEM OF SAMPLE SELECTION APORTATIONS OF DATA MATRICES SYSTEM.”

- [24] T. Seoane, J. L. R. Martín, E. Martín-Sánchez, S. Lurueña-Segovia, and F. J. Alonso Moreno, “Capítulo 5: Selección de la muestra: técnicas de muestreo y tamaño muestral,” *Semergen*, vol. 33, no. 7. pp. 358–361, Aug. 01, 2007. doi: 10.1016/s1138-3593(07)73915-1.
- [25] C. Seger, “An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing,” 2018.
- [26] R. Vínculos Vol ; □ Enero-Junio, “Pronóstico de series de tiempo con redes neuronales regularizadas y validación cruzada Time series forecasting with neural networks regularized and cross validation,” 2013.

ANEXOS

Anexo 1. Variables y atributos del modelo. Elegidos o no para el análisis.

Variable	Nombre de Variable	Atributo	Estado
Edad	Edad	Número entero	Elegida
Género	Género	Masculino	Elegida
		Otro	
		Femenino	
Estrato socioeconómico	Estrato socioeconómico	Pobre	Elegida
		Bajo	
		Medio	
		Alto	
		Muy pudiente	
¿Cuántos hermanos tiene?	Hermanos	Número entero	Elegida
Lugar que ocupa entre los hermanos	Lugar entre hermanos	Soy hijo/a único/a	Elegida
		Primero/a	
		Segundo/a	
		Tercero/a	
		Cuarto/a	
		Después del cuarto/a	
Escolaridad de la madre	Escolaridad de la madre	No lo conozco o es ninguno	
		Escuela/Básica	
		Bachillerato	
		Tercer nivel (grado universitario)	
		Cuarto nivel (maestría)	
		PHD o doctorado	
Escolaridad del padre	Escolaridad del padre	No lo conozco o es ninguno	
		Escuela/Básica	
		Bachillerato	
		Tercer nivel (grado universitario)	

		Cuarto nivel (maestría)	
		PHD o doctorado	
Tipo de colegio	Colegio	Público	Elegida
		Fiscomisional	
		Privado	
¿Con qué nota se graduó del colegio, escalada de 1 a 10?	Nota de grado	Número entero	Elegida
¿A qué edad entró a la Universidad?	Edad de ingreso	Número entero	Elegida
Deserción estudiantil	Deserción estudiantil	Ha abandonado	Elegidas como predictoras
		No ha abandonado	
Recibió Orientación Vocacional	Recibió Orientación Vocacional	No	Elegida
		Sí	
Certeza al elegir la carrera	Certeza	No tenía certeza	Elegida
		Medianamente desconvencido/a*	
		Medianamente convencido/a	
		Con certeza total	
¿Con qué puntaje sobre 1000 ingresó a la Universidad?	Puntaje de ingreso	Número entero	Elegida
Tipo de universidad	Tipo de universidad	Público	Elegida
		Fiscomisional	Atributo eliminado
		Privado	Elegida
Rama de su carrera universitaria	Rama de su carrera universitaria	Agronomía y veterinaria.	Variable eliminada del análisis
		Artes y humanidades.	
		Ciencias naturales, exactas y de la computación.	

		Ciencias sociales, administración y derecho.	
		Educación.	
		Ingeniería, manufactura y construcción.	
		Salud	
		Servicios	
		Otra rama	
Estado civil	Estado civil	Soltero/a	Elegida
		Casado/a	
		Unión libre	
		Divorciado/a	
		Viudo/a	
Con quiénes vivía	Convivencia	Vivía solo/a	Elegida
		Padres y/o hermanos	
		Pareja y/o hijos	
		Otros familiares	
		Amigos, compañeros, etc	
		Otros inquilinos	
Apoyo de la familia por la carrera	Apoyo de la familia	No	Elegida
		Sí	
Aporte económico al hogar	Aporte al hogar	Recibí apoyo de mis familiares, me proveían las necesidades básicas totales.	Elegida
		Aportaba, mis ingresos económicos los destinaba a colaborar en mi casa o gastos personales.	
		Debía responder por mí. No recibía ayuda económica de mi familia ni aportaba a ella	
		Mi familia dependía de mí económicamente.	
		Ninguna de las opciones se ajusta a mi estado económico	

Ocupaciones simultáneas	Ocupaciones simultáneas	Ninguna	Elegida
		Trabajo	
		Deportes	
		Otros estudios	
		Manutención familiar	
		Otras ocupaciones	
Satisfacción con su programa académico	Satisfacción	Muy insatisfecho	Elegida
		Ligeramente insatisfecho	
		Neutral	
		Satisfecho	
		Muy satisfecho	
¿Con qué nota sobre 10 aprobó los primeros semestres?	Nota U	Número racional positivo	Elegida

Anexo 2. Resultados de 10 pruebas aleatorias para datos desbalanceados y balanceados para entrenamiento y testeo. (Valores más altos de las pruebas pintados en verde)

Parámetros de entrenamiento del modelo para datos desbalanceados				
Prueba	Accuracy	Specificity	Sensitivity	F1 Score
1	0.6964	0.8042	0.4278	0.4350
2	0.6750	0.7842	0.4028	0.3883
3	0.6839	0.7774	0.4556	0.4010
4	0.6536	0.7447	0.4278	0.3789
5	0.7214	0.8016	0.5194	0.4750
6	0.7125	0.8095	0.4722	0.4435
7	0.7036	0.7995	0.4667	0.3907
8	0.7179	0.7868	0.5472	0.4994
9	0.7125	0.8095	0.4722	0.4585
10	0.7321	0.8095	0.5417	0.4988
Prom.	0.70089	0.79269	0.47334	0.43691
Desv.	0,024	0,020	0,049	0,046

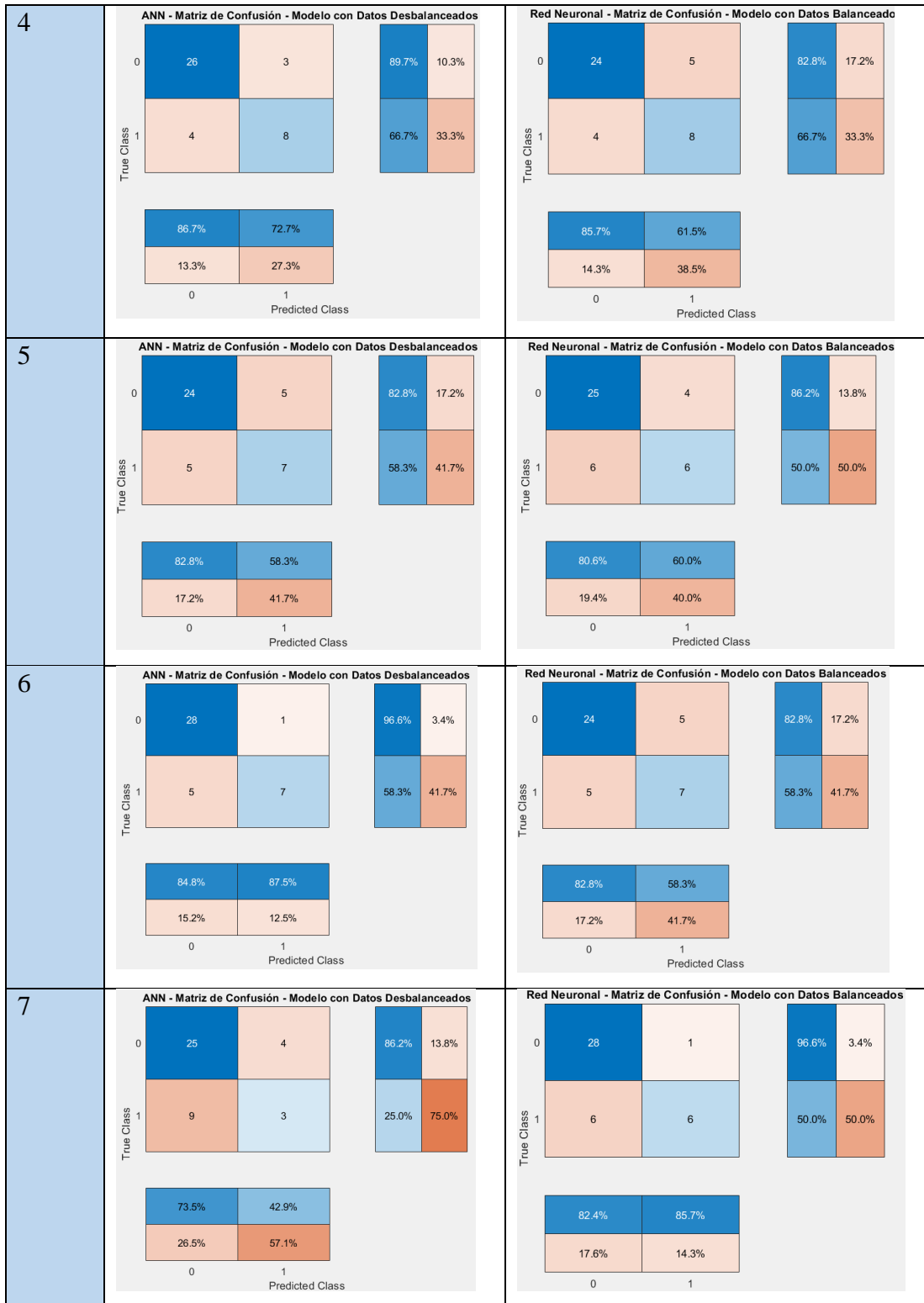
Parámetros de entrenamiento del modelo para datos balanceados				
Prueba	Accuracy	Specificity	Sensitivity	F1 Score
1	0.6768	0.7668	0.4528	0.4332
2	0.6750	0.7668	0.4472	0.4039
3	0.6875	0.7674	0.4917	0.4564
4	0.6661	0.7695	0.4083	0.3775
5	0.7179	0.7942	0.5278	0.5083
6	0.6875	0.7821	0.4528	0.4552
7	0.6839	0.7695	0.4722	0.4135
8	0.6857	0.7416	0.5472	0.4702
9	0.7179	0.7868	0.5472	0.5122
10	0.6768	0.7795	0.4222	0.3735
Prom.	0,68751	0,77242	0,47694	0,44039
Desv.	0,017	0,014	0,050	0,049

Parámetros del testeo para datos desbalanceados				
Prueba	Acurracy	Specificity	Sensitivity	F1 Score
1	0.7805	0.8621	0.5833	0.6087
2	0.7561	0.8621	0.5000	0.5455
3	0.8049	0.8966	0.5833	0.6364
4	0.8293	0.8966	0.6667	0.6957
5	0.7561	0.8276	0.5833	0.5833
6	0.8537	0.9655	0.5833	0.7000
7	0.6829	0.8621	0.2500	0.3158
8	0.7317	0.8276	0.5000	0.5217
9	0.7317	0.8621	0.4167	0.4762
10	0.8293	1.0000	0.4167	0.5882
Prom.	0,77562	0,88623	0,50833	0,56715
Desv.	0,054	0,056	0,121	0,113

Parámetros del testeo para datos balanceados				
Prueba	Acurracy	Specificity	Sensitivity	F1 Score
1	0.7317	0.7586	0.6667	0.5926
2	0.8049	0.8966	0.5833	0.6364
3	0.8049	0.8621	0.6667	0.6667
4	0.7805	0.8276	0.6667	0.6400
5	0.7561	0.8621	0.5000	0.5455
6	0.7561	0.8276	0.5833	0.5833
7	0.8293	0.9655	0.5000	0.6316
8	0.8293	0.8621	0.7500	0.7200
9	0.7805	0.8621	0.5833	0.6087
10	0.8293	0.9655	0.5000	0.6316
Prom.	0,79026	0,86898	0,6	0,62564
Desv.	0,035	0,063	0,086	0,048

Anexo 3. Matriz de confusión de 10 pruebas del modelo

Matriz de confusión																																										
Prueba	Con datos desbalanceados	Con datos balanceados																																								
0	<p>ANN - Matriz de Confusión - Modelo con Datos Desbalanceados</p> <table border="1"> <tr> <td>True Class \ Predicted Class</td> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>0</td> <td>23</td> <td>6</td> <td>79.3% 20.7%</td> </tr> <tr> <td>1</td> <td>9</td> <td>3</td> <td>25.0% 75.0%</td> </tr> <tr> <td></td> <td>71.9%</td> <td>33.3%</td> <td></td> </tr> <tr> <td></td> <td>28.1%</td> <td>66.7%</td> <td></td> </tr> </table>	True Class \ Predicted Class	0	1		0	23	6	79.3% 20.7%	1	9	3	25.0% 75.0%		71.9%	33.3%			28.1%	66.7%		<p>Red Neuronal - Matriz de Confusión - Modelo con Datos Balanceados</p> <table border="1"> <tr> <td>True Class \ Predicted Class</td> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>0</td> <td>21</td> <td>8</td> <td>72.4% 27.6%</td> </tr> <tr> <td>1</td> <td>8</td> <td>4</td> <td>33.3% 66.7%</td> </tr> <tr> <td></td> <td>72.4%</td> <td>33.3%</td> <td></td> </tr> <tr> <td></td> <td>27.6%</td> <td>66.7%</td> <td></td> </tr> </table>	True Class \ Predicted Class	0	1		0	21	8	72.4% 27.6%	1	8	4	33.3% 66.7%		72.4%	33.3%			27.6%	66.7%	
True Class \ Predicted Class	0	1																																								
0	23	6	79.3% 20.7%																																							
1	9	3	25.0% 75.0%																																							
	71.9%	33.3%																																								
	28.1%	66.7%																																								
True Class \ Predicted Class	0	1																																								
0	21	8	72.4% 27.6%																																							
1	8	4	33.3% 66.7%																																							
	72.4%	33.3%																																								
	27.6%	66.7%																																								
1	<p>ANN - Matriz de Confusión - Modelo con Datos Desbalanceados</p> <table border="1"> <tr> <td>True Class \ Predicted Class</td> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>0</td> <td>25</td> <td>4</td> <td>86.2% 13.8%</td> </tr> <tr> <td>1</td> <td>5</td> <td>7</td> <td>58.3% 41.7%</td> </tr> <tr> <td></td> <td>83.3%</td> <td>63.6%</td> <td></td> </tr> <tr> <td></td> <td>16.7%</td> <td>36.4%</td> <td></td> </tr> </table>	True Class \ Predicted Class	0	1		0	25	4	86.2% 13.8%	1	5	7	58.3% 41.7%		83.3%	63.6%			16.7%	36.4%		<p>Red Neuronal - Matriz de Confusión - Modelo con Datos Balanceados</p> <table border="1"> <tr> <td>True Class \ Predicted Class</td> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>0</td> <td>22</td> <td>7</td> <td>75.9% 24.1%</td> </tr> <tr> <td>1</td> <td>4</td> <td>8</td> <td>66.7% 33.3%</td> </tr> <tr> <td></td> <td>84.6%</td> <td>53.3%</td> <td></td> </tr> <tr> <td></td> <td>15.4%</td> <td>46.7%</td> <td></td> </tr> </table>	True Class \ Predicted Class	0	1		0	22	7	75.9% 24.1%	1	4	8	66.7% 33.3%		84.6%	53.3%			15.4%	46.7%	
True Class \ Predicted Class	0	1																																								
0	25	4	86.2% 13.8%																																							
1	5	7	58.3% 41.7%																																							
	83.3%	63.6%																																								
	16.7%	36.4%																																								
True Class \ Predicted Class	0	1																																								
0	22	7	75.9% 24.1%																																							
1	4	8	66.7% 33.3%																																							
	84.6%	53.3%																																								
	15.4%	46.7%																																								
2	<p>ANN - Matriz de Confusión - Modelo con Datos Desbalanceados</p> <table border="1"> <tr> <td>True Class \ Predicted Class</td> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>0</td> <td>25</td> <td>4</td> <td>86.2% 13.8%</td> </tr> <tr> <td>1</td> <td>6</td> <td>6</td> <td>50.0% 50.0%</td> </tr> <tr> <td></td> <td>80.6%</td> <td>60.0%</td> <td></td> </tr> <tr> <td></td> <td>19.4%</td> <td>40.0%</td> <td></td> </tr> </table>	True Class \ Predicted Class	0	1		0	25	4	86.2% 13.8%	1	6	6	50.0% 50.0%		80.6%	60.0%			19.4%	40.0%		<p>Red Neuronal - Matriz de Confusión - Modelo con Datos Balanceados</p> <table border="1"> <tr> <td>True Class \ Predicted Class</td> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>0</td> <td>26</td> <td>3</td> <td>89.7% 10.3%</td> </tr> <tr> <td>1</td> <td>5</td> <td>7</td> <td>58.3% 41.7%</td> </tr> <tr> <td></td> <td>83.9%</td> <td>70.0%</td> <td></td> </tr> <tr> <td></td> <td>16.1%</td> <td>30.0%</td> <td></td> </tr> </table>	True Class \ Predicted Class	0	1		0	26	3	89.7% 10.3%	1	5	7	58.3% 41.7%		83.9%	70.0%			16.1%	30.0%	
True Class \ Predicted Class	0	1																																								
0	25	4	86.2% 13.8%																																							
1	6	6	50.0% 50.0%																																							
	80.6%	60.0%																																								
	19.4%	40.0%																																								
True Class \ Predicted Class	0	1																																								
0	26	3	89.7% 10.3%																																							
1	5	7	58.3% 41.7%																																							
	83.9%	70.0%																																								
	16.1%	30.0%																																								
3	<p>ANN - Matriz de Confusión - Modelo con Datos Desbalanceados</p> <table border="1"> <tr> <td>True Class \ Predicted Class</td> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>0</td> <td>26</td> <td>3</td> <td>89.7% 10.3%</td> </tr> <tr> <td>1</td> <td>5</td> <td>7</td> <td>58.3% 41.7%</td> </tr> <tr> <td></td> <td>83.9%</td> <td>70.0%</td> <td></td> </tr> <tr> <td></td> <td>16.1%</td> <td>30.0%</td> <td></td> </tr> </table>	True Class \ Predicted Class	0	1		0	26	3	89.7% 10.3%	1	5	7	58.3% 41.7%		83.9%	70.0%			16.1%	30.0%		<p>Red Neuronal - Matriz de Confusión - Modelo con Datos Balanceados</p> <table border="1"> <tr> <td>True Class \ Predicted Class</td> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>0</td> <td>25</td> <td>4</td> <td>86.2% 13.8%</td> </tr> <tr> <td>1</td> <td>4</td> <td>8</td> <td>66.7% 33.3%</td> </tr> <tr> <td></td> <td>86.2%</td> <td>66.7%</td> <td></td> </tr> <tr> <td></td> <td>13.8%</td> <td>33.3%</td> <td></td> </tr> </table>	True Class \ Predicted Class	0	1		0	25	4	86.2% 13.8%	1	4	8	66.7% 33.3%		86.2%	66.7%			13.8%	33.3%	
True Class \ Predicted Class	0	1																																								
0	26	3	89.7% 10.3%																																							
1	5	7	58.3% 41.7%																																							
	83.9%	70.0%																																								
	16.1%	30.0%																																								
True Class \ Predicted Class	0	1																																								
0	25	4	86.2% 13.8%																																							
1	4	8	66.7% 33.3%																																							
	86.2%	66.7%																																								
	13.8%	33.3%																																								



8	<p>ANN - Matriz de Confusión - Modelo con Datos Desbalanceados</p> <table border="1"> <tr> <td rowspan="2">True Class</td> <td>0</td> <td>24</td> <td>5</td> <td>82.8%</td> <td>17.2%</td> </tr> <tr> <td>1</td> <td>6</td> <td>6</td> <td>50.0%</td> <td>50.0%</td> </tr> <tr> <td></td> <td></td> <td>80.0%</td> <td>54.5%</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>20.0%</td> <td>45.5%</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>0</td> <td>1</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td colspan="2">Predicted Class</td> <td></td> <td></td> </tr> </table>	True Class	0	24	5	82.8%	17.2%	1	6	6	50.0%	50.0%			80.0%	54.5%					20.0%	45.5%					0	1					Predicted Class				<p>Red Neuronal - Matriz de Confusión - Modelo con Datos Balanceados</p> <table border="1"> <tr> <td rowspan="2">True Class</td> <td>0</td> <td>25</td> <td>4</td> <td>86.2%</td> <td>13.8%</td> </tr> <tr> <td>1</td> <td>3</td> <td>9</td> <td>75.0%</td> <td>25.0%</td> </tr> <tr> <td></td> <td></td> <td>89.3%</td> <td>69.2%</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>10.7%</td> <td>30.8%</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>0</td> <td>1</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td colspan="2">Predicted Class</td> <td></td> <td></td> </tr> </table>	True Class	0	25	4	86.2%	13.8%	1	3	9	75.0%	25.0%			89.3%	69.2%					10.7%	30.8%					0	1					Predicted Class			
True Class	0		24	5	82.8%	17.2%																																																																		
	1	6	6	50.0%	50.0%																																																																			
		80.0%	54.5%																																																																					
		20.0%	45.5%																																																																					
		0	1																																																																					
		Predicted Class																																																																						
True Class	0	25	4	86.2%	13.8%																																																																			
	1	3	9	75.0%	25.0%																																																																			
		89.3%	69.2%																																																																					
		10.7%	30.8%																																																																					
		0	1																																																																					
		Predicted Class																																																																						
9	<p>ANN - Matriz de Confusión - Modelo con Datos Desbalanceados</p> <table border="1"> <tr> <td rowspan="2">True Class</td> <td>0</td> <td>25</td> <td>4</td> <td>86.2%</td> <td>13.8%</td> </tr> <tr> <td>1</td> <td>7</td> <td>5</td> <td>41.7%</td> <td>58.3%</td> </tr> <tr> <td></td> <td></td> <td>78.1%</td> <td>55.6%</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>21.9%</td> <td>44.4%</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>0</td> <td>1</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td colspan="2">Predicted Class</td> <td></td> <td></td> </tr> </table>	True Class	0	25	4	86.2%	13.8%	1	7	5	41.7%	58.3%			78.1%	55.6%					21.9%	44.4%					0	1					Predicted Class				<p>Red Neuronal - Matriz de Confusión - Modelo con Datos Balanceados</p> <table border="1"> <tr> <td rowspan="2">True Class</td> <td>0</td> <td>25</td> <td>4</td> <td>86.2%</td> <td>13.8%</td> </tr> <tr> <td>1</td> <td>5</td> <td>7</td> <td>58.3%</td> <td>41.7%</td> </tr> <tr> <td></td> <td></td> <td>83.3%</td> <td>63.6%</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>16.7%</td> <td>36.4%</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>0</td> <td>1</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td colspan="2">Predicted Class</td> <td></td> <td></td> </tr> </table>	True Class	0	25	4	86.2%	13.8%	1	5	7	58.3%	41.7%			83.3%	63.6%					16.7%	36.4%					0	1					Predicted Class			
True Class	0		25	4	86.2%	13.8%																																																																		
	1	7	5	41.7%	58.3%																																																																			
		78.1%	55.6%																																																																					
		21.9%	44.4%																																																																					
		0	1																																																																					
		Predicted Class																																																																						
True Class	0	25	4	86.2%	13.8%																																																																			
	1	5	7	58.3%	41.7%																																																																			
		83.3%	63.6%																																																																					
		16.7%	36.4%																																																																					
		0	1																																																																					
		Predicted Class																																																																						
10	<p>ANN - Matriz de Confusión - Modelo con Datos Desbalanceados</p> <table border="1"> <tr> <td rowspan="2">True Class</td> <td>0</td> <td>29</td> <td></td> <td>100.0%</td> <td></td> </tr> <tr> <td>1</td> <td>7</td> <td>5</td> <td>41.7%</td> <td>58.3%</td> </tr> <tr> <td></td> <td></td> <td>80.6%</td> <td>100.0%</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>19.4%</td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>0</td> <td>1</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td colspan="2">Predicted Class</td> <td></td> <td></td> </tr> </table>	True Class	0	29		100.0%		1	7	5	41.7%	58.3%			80.6%	100.0%					19.4%						0	1					Predicted Class				<p>Red Neuronal - Matriz de Confusión - Modelo con Datos Balanceado</p> <table border="1"> <tr> <td rowspan="2">True Class</td> <td>0</td> <td>28</td> <td>1</td> <td>96.6%</td> <td>3.4%</td> </tr> <tr> <td>1</td> <td>6</td> <td>6</td> <td>50.0%</td> <td>50.0%</td> </tr> <tr> <td></td> <td></td> <td>82.4%</td> <td>85.7%</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>17.6%</td> <td>14.3%</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>0</td> <td>1</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td colspan="2">Predicted Class</td> <td></td> <td></td> </tr> </table>	True Class	0	28	1	96.6%	3.4%	1	6	6	50.0%	50.0%			82.4%	85.7%					17.6%	14.3%					0	1					Predicted Class			
True Class	0		29		100.0%																																																																			
	1	7	5	41.7%	58.3%																																																																			
		80.6%	100.0%																																																																					
		19.4%																																																																						
		0	1																																																																					
		Predicted Class																																																																						
True Class	0	28	1	96.6%	3.4%																																																																			
	1	6	6	50.0%	50.0%																																																																			
		82.4%	85.7%																																																																					
		17.6%	14.3%																																																																					
		0	1																																																																					
		Predicted Class																																																																						

Anexo 4. Resultados de 10 pruebas aleatorias para datos desbalanceados y balanceados para testeo con 17 variables y 10 variables de entrada.

Parámetros del testeo para datos balanceados (17 var. De entrada)								
Prueba	Datos desbalanceados				Datos balanceados			
	Accuracy	Specificity	Sensitivity	F1 Score	Accuracy	Specificity	Sensitivity	F1 Score
1	0.7805	0.9655	0.3333	0.4706	0.8537	0.9310	0.6667	0.7273
2	0.7805	0.9310	0.4167	0.5263	0.8049	0.8621	0.6667	0.6667
3	0.7317	0.8276	0.5000	0.5217	0.7561	0.7586	0.7500	0.6429
4	0.7317	0.8276	0.5000	0.5217	0.8049	0.8276	0.7500	0.6923
5	0.6341	0.6897	0.5000	0.4444	0.7317	0.6897	0.8333	0.6452
6	0.7073	0.7931	0.5000	0.5000	0.7561	0.7586	0.7500	0.6429
7	0.6829	0.7586	0.5000	0.4800	0.7561	0.7586	0.7500	0.6429
8	0.7317	0.8276	0.5000	0.5217	0.7805	0.8276	0.6667	0.6400
9	0.7805	0.9310	0.4167	0.5263	0.8049	0.8621	0.6667	0.6667
10	0.6829	0.8966	0.1667	0.2353	0.8049	0.8966	0.5833	0.6364
Prom.	0,72438	0,84483	0,43334	0,4748	0,78538	0,81725	0,70834	0,66033
Desv.	0,049	0,086	0,110	0,089	0,036	0,075	0,071	0,029
Parámetros del testeo para datos balanceados (10 var. De entrada)								
Prueba	Datos desbalanceados				Datos balanceados			
	Accuracy	Specificity	Sensitivity	F1 Score	Accuracy	Specificity	Sensitivity	F1 Score
1	0.6829	0.7931	0.4167	0.4348	0.7805	0.7586	0.8333	0.6897
2	0.7561	0.8966	0.4167	0.5000	0.7561	0.7931	0.6667	0.6154
3	0.7805	0.9310	0.4167	0.5263	0.7561	0.7931	0.6667	0.6154
4	0.8049	0.9655	0.4167	0.5556	0.7805	0.8276	0.6667	0.6400
5	0.7073	0.8621	0.3333	0.4000	0.7561	0.7241	0.8333	0.6667
6	0.7805	0.9310	0.4167	0.5263	0.7805	0.7586	0.8333	0.6897
7	0.8293	1.000	0.4167	0.5882	0.8537	0.9310	0.6667	0.7273
8	0.7317	0.8966	0.3333	0.4211	0.8049	0.7931	0.8333	0.7143
9	0.7805	0.8966	0.5000	0.5714	0.7561	0.7586	0.7500	0.6429
10	0.6829	0.8276	0.3333	0.3810	0.7317	0.6897	0.8333	0.6452
Prom.	0,75366	0,90001	0,40001	0,49047	0,77562	0,78275	0,75833	0,66466
Desv.	0,051	0,062	0,053	0,075	0,034	0,065	0,083	0,039