Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Maestría en Sistemas Computacionales

Método basado en detección de prototipos para interpretabilidad de modelos predictivos

TRABAJO RECEPCIONAL que para obtener el **GRADO** de

MAESTRO EN SISTEMAS COMPUTACIONALES

Presenta: **EDUARDO ANGELES CABRERA**

Asesor **DR. ISMAEL MORENO NÚÑEZ**

Tlaquepaque, Jalisco. 12 de Enero de 2021.

AGRADECIMIENTOS

El autor desea dar las gracias al asesor de este trabajo, el Dr. Ismael Moreno Núñez, cuyo apoyo estuvo presente durante todo el desarrollo del mismo.

El autor también desea dar las gracias a CONACYT por el apoyo recibido por medio de la beca número 498424, así como al ITESO y a la empresa LUXOFT por el apoyo recibido por medio de un descuento empresarial y brindando los tiempos y herramientas necesarias para completar el trabajo.

De igual forma, le desea dar gracias a los coordinadores de la maestría y profesores que nos dieron el conocimiento y su paciencia necesarios para desarrollar el presente trabajo.

DEDICATORIA

El autor dedica este trabajo a todas las personas que dieron su apoyo espiritual o económico para poder realizar satisfactoriamente el objetivo de éste. Familiares y amigos, especialmente a los padres y pareja que estuvieron presentes moralmente y que dieron su apoyo en momentos de estrés durante la realización del mismo.

Así como a compañeros de trabajo que escucharon hablar varias veces del tema, a veces sin entenderlo completamente pero mostrando su interés por la salud física y mental del autor durante el tiempo de la realización de este trabajo.

RESUMEN

Para que los modelos predictivos o Máquinas de Aprendizaje sean adoptados por los usuarios, en la industria se han abordado dos estrategias principales: mejorar la precisión de las predicciones del modelo u obtener una explicación o contexto en torno a éstas. Recientemente este último enfoque, conocido como interpretabilidad de máquina de aprendizaje, ha obtenido mucha relevancia para mejorar la adopción de modelos predictivos en la toma de decisiones.

En este trabajo se propone un método que adopta el enfoque de *Deep Learning for Case-Based Reasoning through Prototypes* para mejorar la interpretación de los resultados predictivos con el uso de prototipos. Se propone una solución particular al problema de la interpretabilidad en modelos de redes neuronales cuando los conjuntos de datos no están balanceados.

El método propuesto se validó con dos casos de estudio. El enfoque propuesto conduce a resultados satisfactorios. Finalmente se presentan las conclusiones del trabajo en cuanto al desempeño predictivo y la interpretabilidad del mismo. Finalmente, se plantean trabajos futuros.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

1. INTRODUCCIÓN

Resumen: *En este capítulo se presentan brevemente los antecedentes del objeto de estudio, justificación del objeto de estudio y la definición del problema.*

1.1. Antecedentes

Recientemente, la interpretabilidad de modelos predictivos está ganando cada vez más importancia en los procesos de toma de decisiones basados en datos [1] [2] [3]. Para que los modelos predictivos o máquinas de aprendizaje sean adoptados por los usuarios, en la industria se han abordado dos estrategias principales: mejorar la exactitud de las predicciones del modelo y crear una explicación o contexto en torno a éstas [1].

Se pueden encontrar actualmente múltiples métodos que nos permiten desarrollar la interpretabilidad alrededor de un modelo, agrupándose en diferentes categorías de acuerdo a la etapa de la creación del modelo en la que se implementan [4] [5].

En el primer grupo de métodos, ubicado previo al desarrollo del modelo, se puede encontrar un enfoque cuya base parte de analizar y visualizar los datos de entrenamiento con la finalidad de encontrar cualidades que permitan descubrir patrones y características principales que ayuden a interpretar la información que pasará al modelo, así mismo, asistiendo en la elección del modelo que nos permita obtener una mayor precisión. Podemos encontrar ejemplos como K-means [6] y la correlación de las variables observadas [7].

El segundo grupo de métodos interviene durante el desarrollo del modelo predictivo. Estos métodos ayudan en el análisis de la factibilidad de interpretabilidad de los algoritmos de aprendizaje automático. Aunque existe la posibilidad de obtener una penalización en la precisión, esto permite obtener una interpretación del modelo predictivo. En algunos casos, la secuencia lógica del modelo podría ser observada para determinar la razón de la predicción. Los arboles de decisiones, *RuleFit*, *Naive Bayes*, entre otros, son algunos de los modelos interpretables que se pueden encontrar en la literatura [8] [9].

En la categoría posterior a la construcción del modelo, se pueden observar métodos que analizan la entrada y salida del mismo así como la sensibilidad a perturbaciones en las entradas con el fin de obtener información en torno a su comportamiento, con la ventaja de no depender del tipo de modelo para el uso de estos métodos ni intercambiar la interpretabilidad de éste a cambio de la precisión de las predicciones, a pesar de esto, puede que la información obtenida no sea fiel al modelo. Algunos de ellos son “explicaciones interpretables locales agnósticas al modelo” (LIME por sus siglas en ingles: Local Interpretable Model-agnostic Explanations) [10] y algoritmos de extracción de reglas [11]. Estos métodos son independientes del modelo predictivo y forman parte de un bloque extra a la etapa de entrenamiento y generación de estimaciones del modelo predictivo o algoritmo de Máquina de Aprendizaje [4].

1.2. Justificación

La interpretabilidad de un modelo puede ayudar a disminuir la resistencia a la adopción de éste y a su vez mejorar la precisión de las predicciones. Lo anterior se reflejaría en la adopción de un elemento nuevo en el flujo de trabajo de toma de decisiones que podría elevar la confiabilidad [12].

En el contexto de Inteligencia Artificial Aumentada (*Augmented Artificial Intelligence*) donde los modelos predictivos toman un rol de asistencia en la toma de decisiones, la interpretabilidad de un modelo puede ayudar a disminuir la resistencia al despliegue y adopción de éste y a su vez mejorar el diseño y precisión del mismo. La interpretabilidad debe ser considerada como un paso en el desarrollo de un sistema de inteligencia artificial ya que ayuda a elevar la certeza en la toma de decisiones basadas en éste.

Existen principalmente dos partes interesadas en la interpretabilidad de los modelos, el usuario final y el desarrollador del modelo [2]. Con la implementación de métodos de interpretabilidad se busca incrementar la confianza del usuario en la predicción devuelta por el modelo[13]. Ya que se logra un mayor entendimiento de la predicción, aumentando la capacidad del usuario de validar la veracidad de la predicción, ya sea por su experiencia previa o por información que le permita analizar el caso específico de la predicción. Con esto se facilita la adopción del modelo y se reduce la resistencia de su implementación [12] [14].

Por otro lado, el desarrollador también se beneficia de la interpretabilidad del modelo. Entender como llegó al resultado facilita hacer las correcciones pertinentes para mejorarlo [2] [15]. En ciertos casos (dependiendo del método utilizado para la interpretabilidad del modelo), es posible incluso ver las variables con mayor importancia para la generación de la predicción como en el caso de los arboles de decisión [16]. Así, la ayuda de un experto en el ámbito del cuál sean los datos, ayudaría a realizar los arreglos necesarios incluso desde la fuente de los datos o en el preprocesamiento de éstos [2] [14].

En la Tabla 1 [2] se observan algunos de los métodos de interpretabilidad que pueden ser encontrados en la literatura. Se puede apreciar que hay una diferencia entre la cantidad de métodos que no dependen de un modelo en especifico y la cantidad de aquellos que son intrínsecos al modelo.

Tabla 1 Resumen de métodos de interpretabilidad

Técnicas	Intrínseco / Post-hoc	Global / Local	Modelo-especifico / Modelo-agnóstico
Decision trees	I	G	SP
Rule lists	I	G	SP
LIME	H	L	AG
Shapely explanations	H	L	AG
Saliency map	H	L	AG
Activation maximization	H	G	AG
Surrogate models	H	G/L	AG
Partial Dependence Plot	H	G/L	AG
Individual Conditional Expectation	H	L	AG

Rule extraction	H	G/L	AG
Decomposition	H	L	AG
Model distillation	H	G	AG
Sensitive analysis	H	G/L	AG
Layer-wise Relevant Propagation	H	G/L	AG
Feature importance	H	G/L	AG
Prototype and criticism	H	G/L	AG
Counterfactuals explanations	H	L	AG
I: Intrinseco, H: Post-hoc, G: Global, L: Local, SP: Modelo-especifico, AG: Modelo-agnostico			

Un enfoque que no se ha explorado lo suficiente es el de una metodología que permita la implementación de la interpretabilidad y a su vez mejorar la precisión, dentro de un mismo bloque [9] [17].

1.3. Planteamiento del Problema

El desarrollo de métodos para la toma de decisiones usando datos ha tenido un impacto mayúsculo en la industria [3] [18]. En muchos casos donde se utilizan los modelos, las acciones basadas en sus predicciones no conllevan un riesgo significativo. No obstante, existen ocasiones donde las decisiones tienen consecuencias graves, como es el caso de ciertas áreas del ámbito médico y decisiones económicas (ver los casos de uso en [2] [5] [8] [19] [20]). En este tipo de aplicaciones es necesario entender y generar confianza en el modelo, además de poder requerirse del conocimiento de una persona experta en el dominio [14].

Regularmente en las etapas de desarrollo de modelos predictivos y de máquinas de aprendizaje el principal objetivo es generar predicciones exactas tomando en cuenta diferentes métricas de desempeño para entrenar los modelos y generar un alto desempeño predictivo. Por otro lado, este proceso requiere de métodos de interpretabilidad en diferentes fases del desarrollo. Como un método externo o ajeno al modelo predictivo, los métodos de interpretabilidad permiten obtener información para la selección, implementación y sintonización del modelo predictivo. Posterior a este proceso, los métodos de interpretabilidad permiten generar explicaciones de las predicciones obtenidas.

En cada una de las etapas mencionadas anteriormente se describe a los métodos de interpretabilidad como métodos ajenos a los modelos predictivos que intentan explicar la relación entre los datos de entrada-salida con el modelo predictivo. Sin embargo, existe la necesidad de desarrollar métodos que involucren la interpretabilidad dentro de los modelos

predictivos. Este enfoque indicaría una clara ventaja durante el desarrollo de los modelos predictivos ya que podría asociar la sintonización del modelo con la interpretabilidad del mismo.

El presente trabajo aborda la problemática de la interpretabilidad de modelos predictivos dentro de la estructura del algoritmo de aprendizaje de máquina para tomar en cuenta la precisión del algoritmo de aprendizaje de máquina y la interpretación del mismo.

1.4. Hipótesis

Al integrar métodos de clustering o estimación de prototipos para detección de patrones comunes entre los datos con los modelos predictivos del tipo regresión, se aborda la sintonización de parámetros e híper parámetros como un enfoque de interpretabilidad dentro de la función de optimización y arquitectura del modelo predictivo.

De esta forma, la interpretabilidad ayuda en el diseño de la arquitectura de una red neuronal y en la sintonización de parámetros, reduciendo de esta forma el espacio de búsqueda dentro de métodos como grid-search u otros.

1.5. Objetivos

1.5.1. Objetivo General:

En el presente trabajo se busca desarrollar un método centrado en la interpretabilidad para abordar problemas de sintonización de parámetros, híper parámetros y medición de desempeño de los modelos predictivos. Se acota a los modelos de clasificación de tipo redes neuronales. Se aborda la problemática de balance entre desempeño predictivo e interpretabilidad de los modelos. Se pretende proponer un marco de trabajo adoptando el enfoque *Deep Learning for Case-Based Reasoning through Prototypes,* que involucra una mejora en la interpretación de los resultados con el uso de prototipos [21].

1.5.2. Objetivos Específicos:

1. Proponer un método de análisis de interpretabilidad adoptando el enfoque de *Deep Learning for Case-Based Reasoning through Prototypes* [21].
2. Proponer una asociación entre la clusterización de los datos con los ciertos parámetros de la red neuronal.
3. Determinar un método que ayude a encontrar el número de clústeres o prototipos ideales para el conjunto de datos.
4. Proponer un método que permita mitigar los problemas de conjuntos de datos no balanceados.
5. Establecer una metodología que permita agrupar los puntos anteriores como parte del proceso aprendizaje de máquina.

1.6. Novedad científica, tecnológica o aportación

Los modelos de máquinas de aprendizaje utilizados en procesos de toma de decisiones suelen abordar el problema de la precisión o exactitud de las predicciones. Sin embargo, tienen problemas en su adopción debido a la falta de confianza del usuario. Un factor importante es la implementación de la interpretabilidad que permita facilitar su adopción al entender el contexto de las predicciones generadas.

En este trabajo se propone un método para satisfacer las necesidades actuales de obtener un modelo de máquina de aprendizaje que implemente un método de interpretabilidad para ayudar a la adopción de éste, aunado a métodos que permitan resolver problemas de desequilibrio en el conjunto de datos y optimizar el número de prototipos que representan la variabilidad de los datos de entrada.

2. ESTADO DEL ARTE

Resumen: *En este capítulo se presenta un breve resumen de trabajos recientes sobre interpretabilidad en modelos predictivos. Se hace una revisión de las definiciones principales, la clasificación de los métodos y las principales problemáticas.*

2.1. Interpretabilidad de Máquinas de Aprendizaje

En la literatura de inteligencia artificial y máquinas de aprendizaje no existe una sola definición matematica de interpretabilidad y responde más bien a una noción específica del dominio en el que se aplica. Sin embargo, la importancia de la implementación de interpretabilidad se aborda en diferentes áreas de aplicación [9]. Sin embargo, una de las definiciones que se le ha dado es: el grado en el que un tomador de decisión o analista puede entender la causa de una decisión [12], otra definición es: el grado en el que un usuario puede entender consistentemente el resultado de un modelo [8].

2.2. Importancia de la Interpretabilidad de Máquinas de Aprendizaje

Ciertamente, no todos los sistemas de máquinas de aprendizaje necesitan la implementación de la interpretabilidad, ya sea porque las consecuencias de las malas predicciones no son significativas o porque el problema está tan bien estudiado que se confía en el modelo incluso cuando este no sea perfecto [22]. Sin embargo, una nula o poco precisa explicación en ocasiones donde es necesaria, puede llevarnos a la falta o pérdida de confianza hacia el modelo que se está intentando explicar [9].

De acuerdo a la literatura, las principales razones por las que se busca la interpretabilidad que pueden ser:

a. Explicar para justificar:
   Encontrar las razones o justificaciones para una predicción obtenida [2]. En aquellos escenarios donde son graves las consecuencias de las acciones tomadas a partir de una predicción, es importante que desde la perspectiva del usuario se tenga cierta certeza del funcionamiento del modelo para que exista la confianza necesaria [23], creada con base en las justificaciones dadas por el método de interpretabilidad implementado, y de esta forma aceptar el resultado del modelo [24].
b. Explicar para controlar:
   Pretende mejorar el control que se tiene del modelo al obtener una mejor visión de ciertas vulnerabilidades [2]. En ciertas situaciones donde la predicción del modelo sorprende al usuario, la explicación o interpretabilidad de ésta podría ayudar a captar una predicción sesgada debido a la información de las observaciones del conjunto de datos con el que fue entrenado el modelo [12].
c. Explicar para mejorar:
   Un modelo que puede ser tanto explicado como entendido también es fácil de mejorar al comprender cómo se produjo cierta predicción [2]. Al tener contexto de la predicción se facilita la identificación de artefactos presentes en el modelo [25], al igual que permite tomar ventaja del razonamiento humano para poder promover un mejor desempeño y razonamiento del modelo [26].
d. Explicar para descubrir:

El pedir explicaciones también nos ayuda a obtener nuevo conocimiento [2]. Se podría encontrar que una predicción, que es correcta, sorprende al usuario debido a una expectativa o a un sesgo en su criterio, ayudando así a la confianza hacia el modelo gracias a una explicación que justifique el resultado y a su vez mejorando el conocimiento del usuario. También se podrían identificar aquellas variables o características que son más importante para determinar el resultado del modelo [25], por ejemplo, observando las reglas de un modelo de árbol de decisiones o analizando las características de un prototipo [26].

2.3. Métodos de Interpretabilidad de Aprendizaje de Máquina

Los métodos de interpretabilidad se pueden clasificar en 3 categorías de acuerdo a la etapa de la creación del modelo predictivo en la que se implementan. Antes de la construcción del modelo, durante su construcción y posterior al desarrollo de éste [4].

A. Adadi y M. Berrada mencionan otras formas en las que se pueden clasificar a los métodos interpretables dependiendo de ciertas características de los mismos. Se pueden visualizar algunas de ellas en la Tabla 1 [2].

Por complejidad de los modelos, la cual puede llegar a ser subjetiva y pudiera ser que para ciertos casos un método sea más útil (más interpretable) que otro. Generalmente, entre más complejo sea un modelo, más difícil es poder interpretarlo o explicarlo [2].

Por el alcance del método, donde se tienen 2 variaciones con base en el alcance de la interpretabilidad, se puede buscar entender el comportamiento del modelo o el de una sola predicción [2].

Por la limitación del método con respecto a los modelos, se pueden encontrar métodos que son específicos para cierto tipo de modelos en particular u otros que son agnósticos del modelo [2]. En esta misma clasificación se pueden encontrar, aunado a los métodos que son útiles para modelos específicos, aquellos métodos que se basan en la interpretabilidad intrínseca del modelo, en los cuales la explicación surge del modelo mismo, y no de un bloque ajeno a éste [9].

2.4. Análisis de Métodos de Interpretabilidad de Aprendizaje de Máquina

Recientemente ha habido una explosión en trabajos relativos a la interpretabilidad de máquinas de aprendizaje. Lamentablemente muchos de los métodos de interpretabilidad son externos al modelo, lo cual pudiera ser problemático ya que muchas veces las explicaciones pueden ser confusas o no ser confiables. Por otro lado, los modelos que son inherentemente interpretables podrían ofrecer explicaciones más cercanas al comportamiento real de éstos [9].

Otro de los problemas que se tienen acerca de la interpretabilidad de los modelos de máquinas de aprendizaje es la forma en la que es evaluada, actualmente puede caer en 2 categorías. La primera siendo por su utilidad, donde es apreciada por medio de una aplicación práctica o una versión simplificada de ésta, si es útil, entonces "de alguna forma" es interpretable. En la segunda categoría se evalúa la interpretabilidad por medio del uso de un proxy cuantificable, donde un experto podría determinar que un modelo es interpretable y presentar algoritmos para optimizarlo. Sin embargo, ambas categorías siguen teniendo el mismo enfoque de "lo sabrás cuando lo veas" [22].

3. MARCO TEÓRICO

Resumen: *En este capítulo se presenta la teoría y métodos usados para el modelo Deep Learning for Case-Based Reasoning through Prototypes en este trabajo. Principalmente, se describe la teoría y conceptos fundamentales de los modelos intrínsecamente interpretables.*

3.1. Modelos intrínsecamente interpretables

El desarrollo de modelos intrínsecamente interpretables ha sido un área no tan explorada en comparación a los métodos de interpretabilidad agnósticos al modelo [9], probablemente a causa de que estos últimos no afectan la capacidad predictiva del modelo. Se puede apreciar en la Tabla 1 que la mayoría de los métodos de interpretabilidad que se encuentran en la literatura son *post-hoc*, que regularmente no están ligados a ser usados con un modelo en específico y que al representar un bloque extra al modelo predictivo pueden ser utilizados, uno o varios, en conjunto con prácticamente cualquier modelo sin afectar la precisión predictiva del modelo.

Los modelos intrínsecamente interpretables pueden ser directamente analizados para entender o tener mayor noción de su funcionamiento y contexto de las predicciones, comparándolos con modelos de aprendizaje profundo donde la complejidad del modelo lleva a sacrificar la interpretabilidad del mismo por un mejor desempeño predictivo [27].

Es en estos casos donde se podría optar por un enfoque de interpretabilidad *post-hoc.* Sin embargo, se podrían encontrar algunas ocasiones donde el modelo interpretable tiene el mismo desempeño predictivo que el modelo no interpretable (para mas detalles ver [28]). En tales casos, se preferirían los primeros porque al tener una mejor certeza de su funcionamiento general más confianza generaría en el usuario.

La interpretabilidad de un modelo se puede alcanzar incorporándola directamente en el diseño de éste, lo que podría ser una capa en un modelo de aprendizaje profundo que nos permita interpretar el funcionamiento o razonamiento de dicho modelo [26]. Estos diseños que incorporan la interpretabilidad en el modelo permiten tener un contexto o explicación global del comportamiento del modelo o una explicación de una predicción individual, con lo cual permite que se puedan clasificar por su alcance (Global o Local) [2].

3.2. Deep Learning for Case-Based Reasoning through Prototypes

Como se ha mencionado anteriormente, las redes neuronales han sido regularmente diseñadas y usadas por su capacidad predictiva pero cuya complejidad y naturaleza no lineal han propiciado que sean consideradas como cajas negras, agregando bloques de métodos *post-hoc* para su interpretación lo cual nos deja con interpretaciones que pueden no siempre ser fieles a la naturaleza o razonamiento del modelo.

Se puede definir un prototipo como algo muy cercano o idéntico a una observación de nuestro conjunto de datos, y a su vez, nuestro conjunto de prototipos representaría a nuestro conjunto de datos ya que contienen las características más representativas de los clústeres [21].

El enfoque de *Deep Learning for Case-Based Reasoning through Prototypes* [29] permite la incorporación de una capa especial de prototipos, esta capa ayuda a adaptar el modelo a un

razonamiento basado en ejemplos, haciendo uso de estos prototipos para dotarlo no sólo poder predictivo, también de interpretabilidad.

3.2.1. Arquitectura del *Deep Learning for Case-Based Reasoning through Prototypes*

La arquitectura del modelo *Deep Learning for Case-Based Reasoning through Prototypes* puede ser visualizada en la Figura 1, se pueden observar 2 bloques principales, un *autoencoder* que ayudará a trasladar las observaciones hacia un espacio latente en el que se encuentran los prototipos y a su vez permitirá mover estos últimos al espacio original de las observaciones para poder obtener el contexto de las predicciones. La segunda parte que conforma la arquitectura del modelo *Deep Learning for Case-Based Reasoning through Prototypes* es la red de clasificación que contiene una capa de prototipos, que es la base del enfoque de *Deep Learning for Case-Based Reasoning through Prototypes*, la cual permitiría no sólo clasificar las observaciones, sino también encontrar los prototipos para la cantidad de clústeres m que se haya especificado.

Se supone un conjunto de entrenamiento $D = \{(x_i, y_i)\}_{i=1}^{n}$, donde $x_i \in \mathbb{R}^p$ son las observaciones del conjunto, $y_i \in \{1, \dots, K\}$ es la clasificación correcta de cada observación, i in $\{1, \dots, n\}$, donde n representa el número de observaciones en el conjunto de datos y K es la cantidad de clases que existen en el conjunto de datos.

El primer bloque, el *autoencoder*, incluye un codificador ($f: \mathbb{R}^p \to \mathbb{R}^q$) y un decodificador ($g: \mathbb{R}^q \to \mathbb{R}^p$). El *autoencoder* permite trasladar los prototipos del espacio latente hacia el espacio p-dimensional de los datos de entrada, permitiendo visualizar los prototipos $\mathbf{p}$. Además, el *autoencoder* permite reducir las dimensiones de la entrada y concentrar las características más representativas para estimación de la salida.

La red utiliza la entrada codificada por el *autoencoder* en un espacio latente para calcular las distancias entre la entrada codificada y los m prototipos $\mathbf{p}$ usando el cuadrado de la distancia Euclidiana (*squared Euclidean distance*) para ello se plantea la siguiente formulación

$$\|z - \mathbf{p}_m\|_2^2$$

donde $z = f(x_i)$ es el resultado de la función obtenida del codificador f para la observación x_i y $p(z) = [\|z - \mathbf{p}_1\|_2^2, \|z - \mathbf{p}_2\|_2^2, \ \cdots \ \|z - \mathbf{p}_m\|_2^2,]$ es un vector, obtenido con la formulación anterior, que contiene las distancias entre la observación codificada z y los m prototipos $\mathbf{p}$.

El segundo bloque de la arquitectura del modelo *Deep Learning for Case-Based Reasoning through Prototypes* es la red de clasificación $h: \mathbb{R}^q \to \mathbb{R}^K$ que está conformada por la capa de prototipos $p: \mathbb{R}^q \to \mathbb{R}^m$ que contiene m prototipos, una capa densa o completamente conectada $w: \mathbb{R}^m \to \mathbb{R}^K$ cuya cantidad de neuronas es igual a la cantidad de clases K.

La última capa corresponde a una función *softmax* $s: \mathbb{R}^K \to \mathbb{R}^K$, lo que permite hacer la clasificación de K clases. Sin embargo, también se podría tener una variante que utilice la función sigmoide como función de activación de la capa densa que contenga una sola neurona

y eliminar la capa de *softmax* para clasificaciones binarias, esta variante del modelo puede ser visualizada en la Figura 2.

La capa densa permite estimar la suma ponderada de las distancias mencionadas anteriormente $Wp(z)$ y el resultado es normalizado por la capa de softmax para obtener una distribución de probabilidad de K clases:

$$s(\mathbf{v})_k = \frac{\exp(v_k)}{\sum_{k'=1}^{K} \exp(v_{k'})}$$

Donde v_k es el k-esimo componente del vector $\mathbf{v} = Wp(z)$

Para el entrenamiento la función de error utiliza 4 términos:

- La precisión de la predicción utilizando *cross-entropy loss* [30].

$$E(h \circ f, D) = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} -1[y_i = k] \log\left((h \circ f)_k(x_i)\right)$$

  Donde el *cross-entropy loss* del conjunto de entrenamiento D es denotado por E, $(h \circ f)_k$ es el k-esimo componente para la observación x_i.

  Para la variante de clasificación binaria se utilizaría la siguiente ecuación de *binary cross-entropy* [31]:

$$E(h \circ f, D) = \frac{1}{n} \sum_{i=1}^{n} -1\big(\mathrm{y}_i \log\big((h \circ f)(x_i)\big) + (1 - y_i) \log\big(1 - (h \circ f)(x_i)\big)\big)$$

- El error de la distancia de cada prototipo a las entradas en el espacio latente es calculado con la media de la suma del cuadrado de la distancia mínima.

$$R_1(\mathbf{p}_1, \ldots, \mathbf{p}_m, D) = \frac{1}{m} \sum_{j=1}^{m} \min_{i \in [1,n]} \left\| \boldsymbol{p}_j - f(x_i) \right\|_2^2$$

- El error de la distancia de las entradas hacia los prototipos en el espacio latente, al igual que el punto anterior, es calculado usando la media de la suma del cuadrado de la distancia mínima.

$$R_2(\mathbf{p}_1, \ldots, \mathbf{p}_m, D) = \frac{1}{n} \sum_{i=1}^{n} \min_{j \in [1,m]} \left\| f(x_i) - \boldsymbol{p}_j \right\|_2^2$$

- El error de la reconstrucción del *autoencoder* es calculado con el promedio de la distancia cuadrada (*MSE - Mean Squared Error*) entre la observación original y la reconstrucción.

$$R(g \circ f, D) = \frac{1}{n} \sum_{i=1}^{n} \|(g \circ f)(x_i) - x_i\|_2^2$$

Siendo el error global L la suma ponderada de cada uno de los errores anteriores para el conjunto de datos D, dando igual importancia a todos cuando sus coeficientes λ son 1 para cada termino.

$$L\big((f,g,h),D\big) = \lambda_1 E(h \circ f, D) + \lambda_2 R(g \circ f, D) + \lambda_3 R_1(\mathbf{p}_1, \dots, \mathbf{p}_m, D) + \lambda_4 R_2(\mathbf{p}_1, \dots, \mathbf{p}_m, D)$$

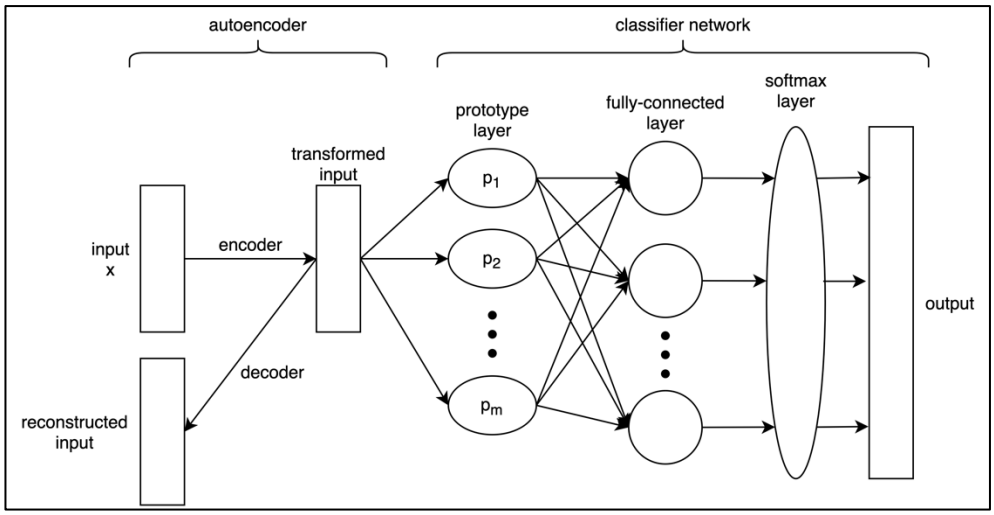

Figura 1 Arquitectura modelo de clasificación multiclase [21]

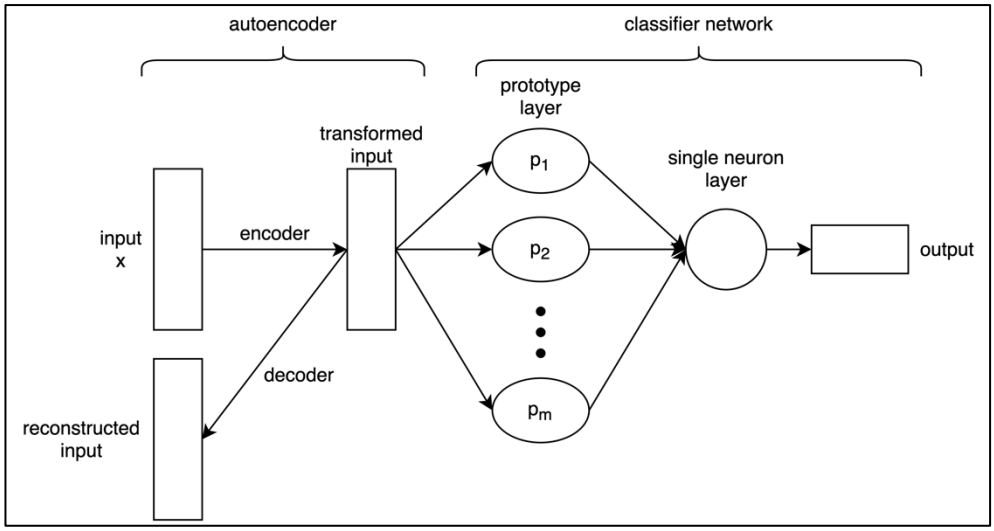

Figura 2 Arquitectura del modelo de clasificación binaria [21]

Los prototipos serían representativos de los clústeres de datos gracias a los términos de error R_1 y R_2 ya que ambos incentivarán al modelo a que los prototipos sean las mejores representaciones de los clústeres. La minimización del error R_1 entre los prototipos y la observación más cercana ayudará a que éstos estén lo más cerca posible de por lo menos una

observación del clúster, haciendo que su decodificación al espacio de las observaciones sea significativa para la interpretabilidad del modelo.

De igual forma, la minimización del error R_2 entre las observaciones y el prototipo más cercano será el incentivo que ayude a que la observación codificada sea lo más cercana posible a alguno de los prototipos en el espacio latente.

3.2.2. Interpretabilidad del modelo

El bloque de *autoencoder* del modelo *Deep Learning for Case-Based Reasoning through Prototypes* permitiría visualizar los prototipos obtenidos durante el entrenamiento, lo que le daría al usuario la habilidad de interpretar el comportamiento del modelo para llegar a cierta predicción.

Los prototipos son calculados durante la fase de entrenamiento, por lo tanto, plantean una explicación de cada predicción siendo fieles al comportamiento o razonamiento del modelo debido a que las observaciones son clasificadas con base en su proximidad con cada uno de los prototipos.

Así mismo, las explicaciones de las predicciones obtenidas del modelo predictivo se basan en su similitud con los prototipos y no tanto en resaltar las partes relevantes de la observación.

3.3. Clusterización

Las técnicas de clusterización de los datos ayudan a encontrar ciertos patrones en los que se agrupan las clases dentro del espacio de las observaciones. Técnicas como K-means necesitan definir previamente el numero de clústeres a buscar.

Estas técnicas uscan minimizar el error o la distancia euclidiana del centroide. El centro del clúster que no necesariamente será una de las observaciones del conjunto de datos, de cada clúster hacia cada una de las observaciones cercanas que puede ser calculada de la siguiente forma:

$$\sum_{i=0}^{n} \min_{\mu_j \in C} \left(\|x_i - \mu_j\|^2 \right)$$

Donde n es el numero de observaciones o muestras x divididas en C clusteres.

3.4. Desbalance de clases

En ciertos casos, los conjuntos de datos creados a partir de observaciones de la vida real pueden estar desequilibrados, esto es porque los datos “normales” suelen predominar a

aquellos que son “interesantes” y la diferencia puede ser de incluso 100,000 a 1 en algunas aplicaciones, dependiendo del caso se puede intentar hacer un sobremuestreo de la clase minoritaria o un submuestreo de la clase mayoritaria.

Se propone utilizar el método SMOTE [32] para el proceso de sobremuestreo por su enfoque de producir observaciones sintéticas de la clase minoritaria. Tomando como base 1 observación y a partir de las características de cierta cantidad de vecinos seleccionados al azar de los k más cercanos, en el caso de requerirse 200% de sobremuestreo se seleccionarían 2 vecinos de los k más cercanos y las observaciones son creadas en la dirección de los vecinos resaltados. Las observaciones sintéticas son creadas tomando la diferencia de los vectores de la observación siendo considerada y de su vecino más cercano, se multiplica esta diferencia por un numero aleatorio entre 0 y 1 y se agrega al vector de la observación siendo considerada, lo que genera un punto aleatorio en el segmento lineal entre ambas observaciones. Este proceso también ayuda al modelo de clasificación a generalizar mejor la clase minoritaria, el pseudo código de este método puede se concentrado en [32].

3.5. Validación y entrenamiento

Para la validación de los datos de prueba durante el entrenamiento se propone utilizar 4 de las métricas principales para clasificación: *Precision*, *Recall*, *Accuracy* y *F1-score*

Precision es la habilidad del modelo de no etiquetar como positiva una observación que es negativa y contestaría la pregunta ¿qué proporción de las observaciones positivas fueron identificadas correctamente?

$$Precision = \frac{TP}{TP+FP}.$$

Donde TP es la cantidad de predicciones positivas verdaderas y FP la cantidad de predicciones falsas positivas.

Recall es la habilidad del modelo de encontrar todas las observaciones positivas y respondería a la pregunta ¿qué proporción de las observaciones positivas fueron identificadas correctamente?

$$Recall = \frac{TP}{TP+FN}.$$

Donde TP es la cantidad de predicciones positivas verdaderas y FN la cantidad de predicciones falsas negativas.

Accuracy es la habilidad del modelo de clasificar correctamente las observaciones de todas las clases y puede ser representada como:

$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i)$$

Donde y es el resultado real para cada observación y $\hat{y}$ es el resultado de la predicción.

F1-score puede ser definida como la media harmónica ponderada de *precision* y *recall* de la prueba, donde la importancia de ambos es igual y es utilizada como una medida más realista del desempeño de la prueba.

$$F1 = \frac{2 * precision * recall}{precision + recall}.$$

4. DESARROLLO METODOLÓGICO

Resumen: *En este capítulo se presenta la metodología propuesta.*

4.1. Casos de Uso

La metodología propuesta se evaluó en dos casos de uso utilizando los conjuntos de datos:

1. "*Online Shoppers Purchasing Intention*" en [33] [34].
2. "*South German Credit*" en [35], [36].

Ambos conjuntos de datos disponibles en el repositorio de *UC Irvine Machine Learning Repository*.

En el primer conjunto de datos se pretende identificar a aquellos usuarios que realizarán una compra en un sitio de comercio electrónico, basándose en distintos datos relacionados con la navegación del usuario en el sitio e información relevante con respecto a la fecha de visita, así como estadísticas provistas por el servicio de "*Google Analytics*".

En el segundo conjunto de datos (cuya información es de los años 70) se intenta reconocer a aquellos usuarios que tengan más probabilidad de cumplir satisfactoriamente con el contrato del crédito que se está solicitando, para ello se apoya de datos que caracterizan su situación personal, económica y legal, así como de información relevante del crédito que se está tramitando.

Con los prototipos obtenidos y sus valores se busca entender que perfil tiene el prototipo en base a las diferentes variables. Por ejemplo: para el segundo conjunto de datos un prototipo puede ser uno que tenga una situación económica alta, situación personal: hombre, adulto mayor, casado, ciudad grande, ingresos altos, etc.

Estos conjuntos de datos están ligeramente desbalanceados, es decir, un conjunto de datos cuyas clasificaciones categóricas no están representadas en cantidades aproximadamente iguales. Esto es porque muchas veces en la realidad los casos normales suelen predominar en comparación con los casos anormales o interesantes [32].

Las problemáticas descritas de los dos conjuntos de datos mencionados anteriormente se pueden resolver con la metodología y el modelo *Deep Learning for Case-Based Reasoning through Prototypes* en este trabajo porque se abordan necesidades de clasificación donde se busca encontrar las observaciones de ciertas características. Por lo tanto, el modelo basado en prototipos sería de gran ayuda para poder identificar las características más representativas de cada uno de los clústeres de los conjuntos de datos que nos ayudarán a clasificar las observaciones.

El flujo que seguirían los datos de cada conjunto dentro de la metodología propuesta puede ser visualizado en la Figura 3, donde se puede observar cada una de las etapas de preprocesamiento previa al modelo predictivo y que el modelo predictivo esta conformado por las etapas: clusterización, autoencoder y clasificación DNN.

La metodología propuesta consiste principalmente en proponer métodos de balanceo de datos y clusterización para poder obtener una condición a priori para definir el rango de búsqueda de parámetros de la red neuronal y número de prototipos. Además, se busca balancear la interpretabilidad de la máquina de aprendizaje con el buen desempeño para la generación de estimaciones.

La metodología propuesta puede resumirse en el diagrama de bloques de la Figura 3. Donde se muestran las etapas:

- Codificación: Se cambian los valores de aquellas variables categóricas que sean del tipo “String” por una representación numérica.
- Balanceo de datos: Se realiza un sobremuestreo de la categoría menor para balancear las categorías del conjunto de datos.
- Estandarización de los datos: Se centran los datos para obtener una media de cero y una varianza unitaria.
- Clusterización: Se busca la cantidad de clústeres ideal para definir la cantidad de prototipos que buscará el modelo.
- Autoencoder: Se transforman los datos a un espacio latente donde se compararán con los prototipos.
- Clasificación: Los datos se introducen a la red de clasificación del modelo donde se calcularán las distancias hacia los prototipos.

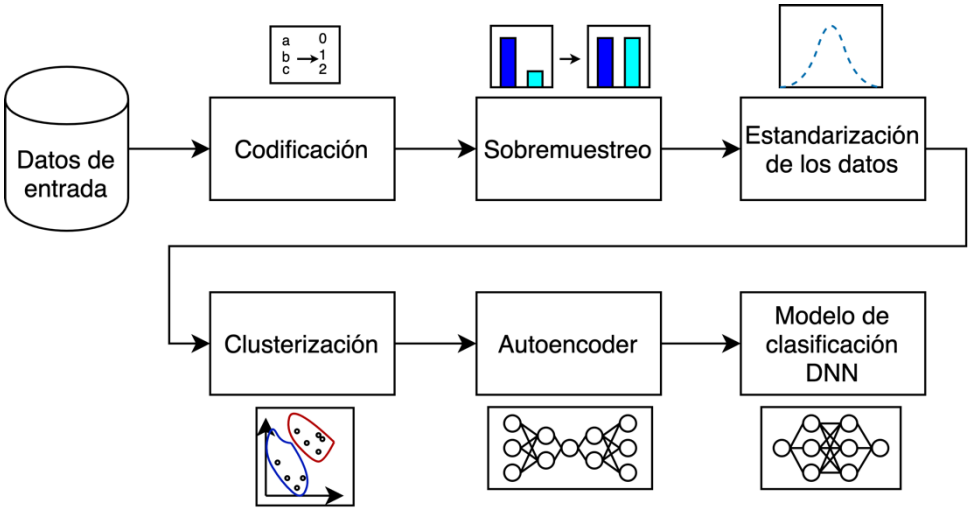

Figura 3 Diagrama a bloques de la metodología propuesta para la interpretabilidad de modelo predictivo usando prototipos.

En las siguientes secciones se profundiza en cada una de las etapas de la metodología propuesta a la luz de los casos de uso.

4.2. Datos

4.2.1. Descripción de los Datos

El primer conjunto de datos *Online Shoppers Purchasing Intention* está compuesto por 1 archivo en formato CSV de 1.1 MB que contiene 12,330 observaciones representadas por 18 variables, 10422 observaciones donde no se efectuó una compra y 1908 observaciones en las

que sí ocurrió el evento de interés. La salida puede ser representada como una variable de 2 categorías: cuando se efectuó una compra y cuando la compra no fue realizada.

En el segundo conjunto de datos *South German Credit* las observaciones son representadas por 21 variables. Un archivo en formato CSV de 4.8 KB contiene 1000 observaciones de las cuales en 700 el crédito no fue concluido satisfactoriamente y las otras 300 sí se concluyeron satisfactoriamente.

Estos conjuntos de datos están ligeramente desbalanceados, es decir, un conjunto de datos cuyas clasificaciones categóricas no están representadas en cantidades aproximadamente iguales. Esto es porque muchas veces en la realidad los casos normales suelen predominar en comparación con los casos anormales o interesantes [32].

4.2.2. Estructura de los Datos

En las tablas Tabla 2 y Tabla 3 se puede encontrar un ejemplo de algunas observaciones de los conjuntos de datos.

Tabla 2 Ejemplo de observaciones en el conjunto de datos - Online Shoppers Intention

	Sample1	Sample2	Sample3	Sample4	Sample5
Administrative	0	0	0	0	0
Administrative_Duration	0	0	0	0	0
Informational	0	0	0	0	0
Informational_Duration	0	0	0	0	0
ProductRelated	1	2	1	2	10
ProductRelated_Duration	0	64	0	2.66667	627.5
BounceRates	0.2	0	0.2	0.05	0.02
ExitRates	0.2	0.1	0.2	0.14	0.05
PageValues	0	0	0	0	0
SpecialDay	0	0	0	0	0
Month	Feb	Feb	Feb	Feb	Feb
OperatingSystems	1	2	4	3	3

Browser	1	2	1	2	3
Region	1	1	9	2	1
TrafficType	1	2	3	4	4
VisitorType	Returning_Visitor	Returning_Visitor	Returning_Visitor	Returning_Visitor	Returning_Visitor
Weekend	False	False	False	False	True
Revenue	False	False	False	False	False

Tabla 3 Ejemplo de observaciones en el conjunto de datos - South German Credit

	Sample1	Sample2	Sample3	Sample4	Sample5
status	1	1	2	1	1
duration	18	9	12	12	12
credit_history	4	4	2	4	4
purpose	2	0	9	0	0
amount	1049	2799	841	2122	2171
savings	1	1	2	1	1
employment_duration	2	3	4	3	3
installment_rate	4	2	2	3	4
personal_status_sex	2	3	2	3	3
other_debtors	1	1	1	1	1
present_residence	4	2	4	2	4
property	2	1	1	1	2
age	21	36	23	39	38
other_installment_plans	3	3	3	3	1
housing	1	1	1	1	2
number_credits	1	2	1	2	2
job	3	3	2	2	2

people_liable	2	1	2	1	2
telephone	1	1	1	1	1
foreign_worker	2	2	2	1	1
credit_risk	1	1	1	1	1

4.2.3. Características / Variables

En Tabla 4 y Tabla 5 podemos observar las variables que contienes las observaciones de los conjunto de datos, así como el tipo de dato dado por la librería *Pandas*, se cuenta con variables numéricas y categóricas, estas ultimas serán modificadas durante el pre-procesamiento para que su valor sea numérico.

Tabla 4 Variables del conjunto de datos - Online Shoppers Intention

Nombre	Tipo	Descripción
Administrative	Integer	Número de paginas visitadas de tipo administrativo.
Administrative_Duration	Float	Duración en paginas de tipo administrativo.
Informational	Integer	Número de paginas visitadas de tipo informativo.
Informational_Duration	Float	Duración en paginas de tipo informativo.
ProductRelated	Integer	Número de paginas visitadas relacionadas con el producto.
ProductRelated_Duration	Float	Duración en paginas relacionadas con el producto.
BounceRates	Float	Porcentaje de visitantes que entraron a una página y la abandonaron sin ninguna otra petición al servicio de "Google Analytics".
ExitRates	Float	Porcentaje de visitantes que entraron a una página y la abandonaron sin ninguna otra petición al servicio de "Google Analytics", siendo esta la última de la sesión.
PageValues	Float	Valor promedio de la página web visitada antes de completar una transacción de comercio electrónico.
SpecialDay	Float	Cercanía de un día especial (por ejemplo: día de las madres o San Valentín) a la fecha en que se visita la página.

Month	String Categórica	Mes de la visita.
OperatingSystems	Integer Categórica	Sistema operativo.
Browser	Integer Categórica	Navegador.
Region	Integer Categórica	Región.
TrafficType	Integer Categórica	Tipo de tráfico.
VisitorType	String Categórica	Tipo de visitante.
Weekend	String /Boolean	Si es fin de semana.
Revenue	String/Boolean	Si se efectuó una compra.

Tabla 5 Variables del conjunto de datos - South German Credit

Nombre	Tipo	Descripción
status	Integer Categórica	Estado de la cuenta de cheques del deudor en el banco.
duration	Integer	Duración del crédito en meses.
credit_history	Integer Categórica	Historial de cumplimiento de contratos de crédito anteriores o concurrentes.
purpose	Integer Categórica	Propósito del crédito.
amount	Integer	Cantidad del crédito (es el resultado de una transformación monotónica, no se sabe el valor real).
savings	Integer Categórica	Ahorros del deudor.
employment_duration	Integer	Duración del deudor con su empleador actual.

	Categórica	
installment_rate	Integer	Cuotas de crédito como porcentaje de la renta disponible del deudor.
personal_status_sex	Integer Categórica	Información categórica combinada de la información del sexo y estado civil del deudor (no se puede obtener el sexo del deudor con esta variable porque hombres y mujeres no solteros pertenecen a la misma categoría).
other_debtors	Integer Categórica	¿Existe otro deudor o garante del crédito?
present_residence	Integer	Tiempo en años del deudor viviendo en su actual vivienda.
property	Integer Categórica	La propiedad más valiosa del deudor.
age	Integer	Edad del deudor en años
other_installment_plans	Integer Categórica	Planes de pago a plazos de proveedores distintos al banco que otorga el crédito.
housing	Integer Categórica	Tipo de vivienda en la que vive el deudor.
number_credits	Integer	Numero de créditos que tiene o tuvo el deudor ,incluyendo el actual, en el banco.
job	Integer Categórica	Calidad del trabajo del deudor.
people_liable	Integer	Número de personas que dependen económicamente del deudor.
telephone	Integer Categórica	¿Hay un teléfono fijo registrado a nombre del deudor?
foreign_worker	Integer Categórica	¿El deudor es un trabajador extranjero?
credit_risk	Integer Categórica	¿Se ha cumplido (1) o no (0) el contrato de crédito?

4.3. Procesamiento de los Datos

Previo al entrenamiento del modelo, se procesaron los datos para garantizar que estos puedan ser utilizados por el modelo. El procesamiento de los datos, que se explicará en las siguientes subsecciones, incluye la codificación de variables categóricas [37], la estandarización de las variables [38], la clusterización de los datos [39] y el realizar el sobremuestreo [40].

4.3.1. Codificación

Se utilizó la función *OrdinalEncoder* [37] para codificar aquellas variables categóricas de nuestro conjunto de datos que sean originalmente cadenas de caracteres, la Tabla 6 y Tabla 7 demuestran un ejemplo de una observación previa y posterior al proceso de codificación.

Durante el proceso, se identifican las categorías de cada una de las variables categóricas y se transforman a una representación numérica, por ejemplo el conjunto: [Europa, América, Asia] sería transformado a [0,1,2]

Tabla 6 Observación previa a la codificación - Online Shoppers Intention

	Sample 1
Administrative	0
Administrative_Duration	0
Informational	0
Informational_Duration	0
ProductRelated	1
ProductRelated_Duration	0
BounceRates	0.2
ExitRates	0.2
PageValues	0
SpecialDay	0
Month	Feb
OperatingSystems	1
Browser	1

Region	1
TrafficType	1
VisitorType	Returning_Visitor
Weekend	False
Revenue	False

Tabla 7 Observación posterior a la codificación - Online Shoppers Intention

	Sample 1
Administrative	0.0
Administrative_Duration	0.0
Informational	0.0
Informational_Duration	0.0
ProductRelated	1.0
ProductRelated_Duration	0.0
BounceRates	0.2
ExitRates	0.2
PageValues	0.0
SpecialDay	0.0
Month	2.0
OperatingSystems	1.0
Browser	1.0
Region	1.0
TrafficType	1.0
VisitorType	2.0
Weekend	0.0

Revenue	0.0

4.3.2. Sobremuestreo

Como se explicó en la sección 3.4, para el sobremuestreo de la clase minoritaria se utilizará el método *SMOTE* [40] para aumentar el número de observaciones de esta clase con la generación de observaciones sintéticas.

En la Figura 4 y Figura 5 se muestran un ejemplo de la diferencia en la cantidad de observaciones de cada clase antes y después del proceso de sobremuestreo, en la Figura 4 se puede observar la diferencia en la cantidad de observaciones de cada clase del conjunto de datos "*Online Shoppers Intention*", siendo la clase minoritaria aquella que se considera interesante para el propósito de dicho conjunto de datos.

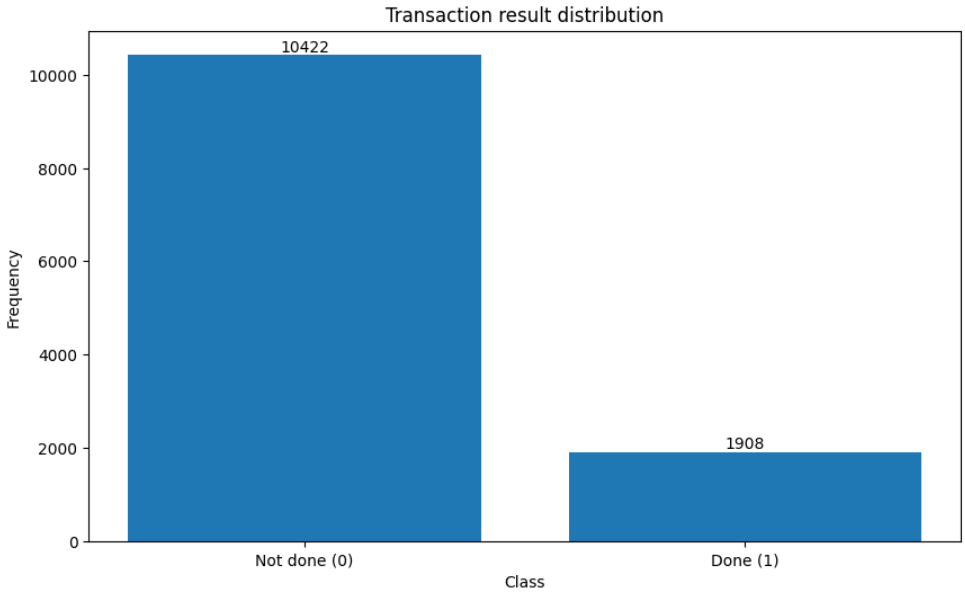

Figura 4 Distribución de clases previo al sobremuestreo - Online Shoppers Intention

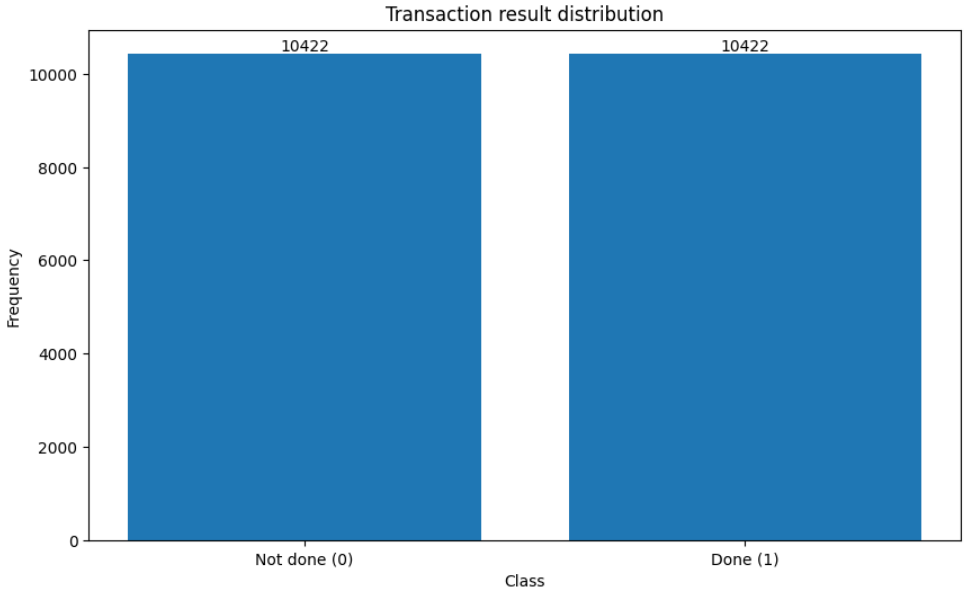

Figura 5 Distribución de clases posterior al sobremuestreo - Online Shoppers Intention

4.3.3. Estandarización de los datos

Se estandarizaron las variables del conjunto de datos haciendo uso de la función *StandarScaler* [38] que permite centrar los datos (media de 0) y escalarlos a una varianza unitaria, para lograr esto se resta la media y se divide entre la desviación estándar, esto se calcula para cada una de las características o variables del conjunto de datos.

$$z = \frac{x - u}{s}$$

Donde x es una observación del conjunto de datos, u es la media y s es la desviación estándar.

4.3.4. Clusterización

Usando el método de *K-means [39]*, se realizó la clusterización del conjunto de datos haciendo uso del método del codo [41]. El método del codo es una técnica heurística que pretende determinar el número de clústeres en un conjunto de datos, se utilizaron desde uno hasta diez clústeres para encontrar el número con mayor pendiente de error, esto es, el número de clústeres con la mayor diferencia de error con respecto al anterior.

Las Figura 6 y Figura 7 muestran la curva del error utilizando diferentes cantidades de clústeres, en ambos casos el número de clústeres sugerido por este método sería dos, ya que la recta con una mayor inclinación es la que va de 1 a 2 en el eje x.

En ambos casos la cantidad de clases reales que contienen los conjuntos de datos es dos, por lo que en este caso coincide que el número de prototipos para el modelo será el mismo. Sin embargo, esto es sólo un número de clústeres sugerido por este método para empezar a trabajar con nuestro modelo y no debería ser considerado como un valor definitivo ya que en la realidad podrían existir más perfiles de clientes que sólo aquellos clasificados en el conjunto de datos.

Esto permite identificar el número sugerido de prototipos para el modelo, aunque no necesariamente debe ser igual al número de clases definidos en el conjunto de datos.

El tamaño del error se debe principalmente a la cantidad de observaciones y características que hay en cada uno, considerando que el conjunto de datos "*Online Shoppers Intention*" contiene 12,330 y 18 variables, se puede observar en la Figura 6 un tamaño de error considerablemente grande, mientras que en la Figura 7 el tamaño del error es menor, ya que el conjunto de datos "*South German Credit*" contiene sólo 1000 observaciones con 21 variables.

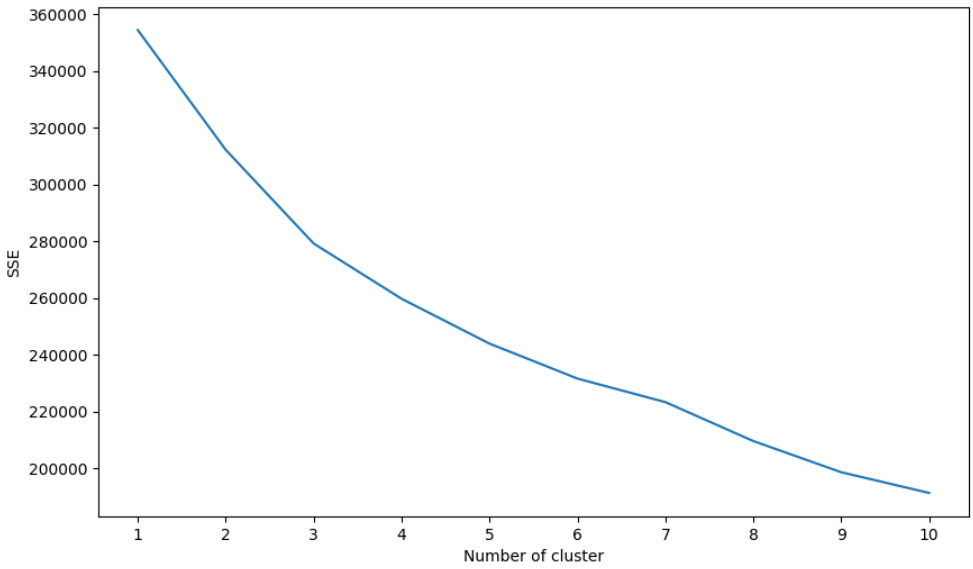

Figura 6 Búsqueda del número de prototipos ideal - Online Shoppers Intention

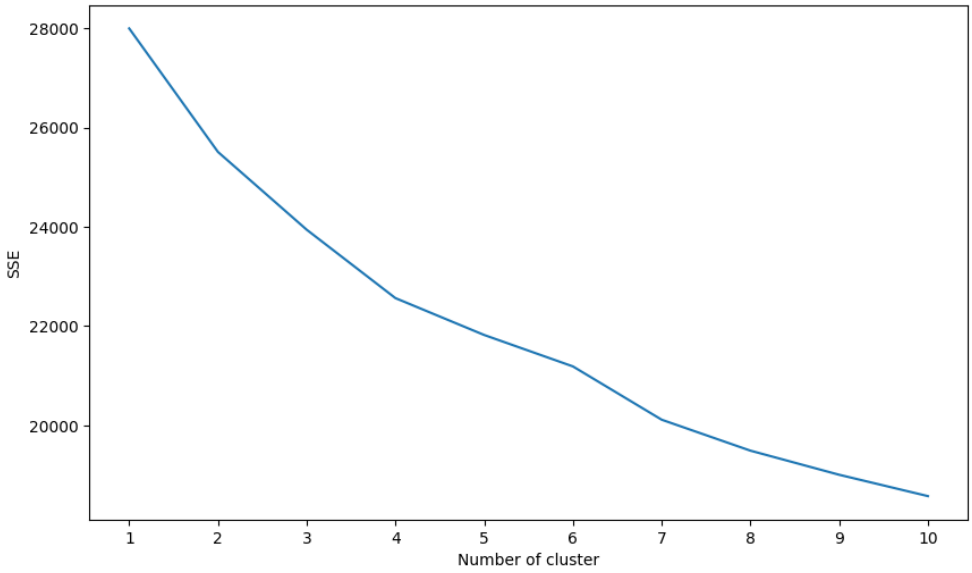

Figura 7 Búsqueda del número de prototipos ideal - South German Credit

4.4. Entrenamiento del modelo

Como se puede apreciar en la Figura 2, la arquitectura del modelo *Deep Learning for Case-Based Reasoning through Prototypes* para una clasificación binaria, posterior al pre-procesamiento de los datos, se introducen los mismos a la red del modelo donde serán codificados en la primera parte del *autoencoder* para ser utilizados en el espacio latente de éste y poder comparar las distancias de cada observación hacia los prototipos que habitan en el mismo espacio latente.

Como se mencionó en la sección de arquitectura del modelo adoptado, el *autoencoder* permite comparar las observaciones y los prototipos en un espacio latente y a su vez ayuda a encontrar las características importantes para la representación de los datos.

Con las distancias de cada observación hacia los prototipos se crea un vector, éste se utiliza para calcular la suma ponderada de las distancias con los pesos de la siguiente capa.

La siguiente capa en este caso contará sólo con 1 neurona, como se muestra en la Figura 2 ya que los casos de uso son de clasificación binaria, y se redondea el resultado obtenido de la neurona para obtener la predicción.

En el caso de una clasificación multiclase, el resultado de la capa densa se introduce a la siguiente capa de *softmax*, como se puede apreciar en la Figura 1, para obtener una distribución de probabilidades de las clases.

4.5. Validación y entrenamiento

Para la validación de los datos de prueba se utilizarán las 4 métricas de error propuestas anteriormente mencionadas en la sección 3.5: *Precision*, *Recall*, *Accuracy* y *F1-score*.

4.6. Interpretación / interpretabilidad de resultados

Para la interpretación de los prototipos, después de extraer los prototipos decodificados al espacio dimensional de las observaciones (utilizando el modulo de decoder del autoencoder), se seleccionan aquellas características de los prototipos que tienen valores diferentes ya que son importantes para poder identificar la cercanía de la observación hacia los prototipos, con esto se puede obtener un contexto global del comportamiento del modelo al obtener las características más representativas para cada clase.

Con lo anterior se puede realizar una comparación de la cantidad de características con más cercanía de una observación hacia cada uno de los prototipos, pudiendo así observar rápidamente las características que llevaron al modelo dar cierta predicción.

Tomando como ejemplo los prototipos y resultados obtenidos en el segundo conjunto de datos "*South German Credit*" mostrados más adelante en la Tabla 14, en la cuál se puede apreciar el valor de cada característica o variable de los prototipos resultantes del modelo y destacando las características con valores distintos entre los prototipos (como se mencionó en el primer párrafo de este inciso), así como el valor de 2 observaciones del conjunto de datos, se puede observar que las variables destacadas o relevantes para los prototipos son: "status", "employment duration", "installment rate", "present residence", "age", "other installment plans", "people liable", "telephone", "foreign worker".

Al observar los valores de las características de los prototipos se puede interpretar que el perfil del deudor que cumplirá satisfactoriamente el crédito debe tener: 4 años o más en su actual trabajo, que la mensualidad del crédito sea el 3.13% o más de sus ingresos, que tenga más de 2.9 años en su residencia actual, tener más de 35 años, no tener planes de pago a plazos con otros proveedores, tener 2 personas que dependan económicamente del deudor, tener un teléfono registrado a nombre del deudor y no ser un trabajador extranjero.

5. RESULTADOS Y DISCUSIÓN

Resumen: *En este capítulo se presentan los resultados obtenidos del desarrollo de este trabajo y una discusión sobre el desempeño predictivo y la interpretabilidad de la metodología propuesta*.

5.1. Resultados

Se pudo observar que la metodología propuesta basada en redes neuronales profundas con la incorporación de una capa de prototipos basado en el enfoque de *Deep Learning for Case-Based Reasoning through Prototypes* [21], mantuvo un buen desempeño predictivo y a su vez permitió la incorporación de la interpretabilidad en el mismo con el uso de los prototipos. Aunado a lo anterior, la metodología también abordó satisfactoriamente la problemática de los conjuntos de datos desequilibrados utilizando el método SMOTE para la creación de observaciones sintéticas.

5.2. Conjunto de datos 1: "Online Shoppers Intention"

En las figuras Figura 8 y Figura 9, se pudo observar que el entrenamiento del modelo utilizando el conjunto de datos "Online Shoppers Intention" con 1,000 épocas obtuvo un 89% de precisión, el cuál puede ser observado en el reporte de clasificación en la Tabla 8 donde también podemos observar que tiene un buen desempeño predictivo, similar para ambas clases con 89% y 88% de F1.

En la Figura 10 se pudo observar el histograma de la distribución de las predicciones hechas por el modelo para el conjunto de datos de prueba, cada columna representa un rango del resultado de la predicción que van desde 0 hasta 1 en este caso ya que es una clasificación binaria, tal que entre más cercano sea el valor de la predicción a uno de los extremos, más será la confianza del modelo sobre dicha predicción, por otro lado, mientras más cerca esté la predicción del umbral (.5 en este caso) habrá más incertidumbre sobre la misma.

En la misma figura fue posible apreciar que el modelo tiende a dar resultados correctos en los extremos de 0 a .1 y de .9 a 1, 40% y 37% respectivamente, con un porcentaje de error de 3% en conjunto.

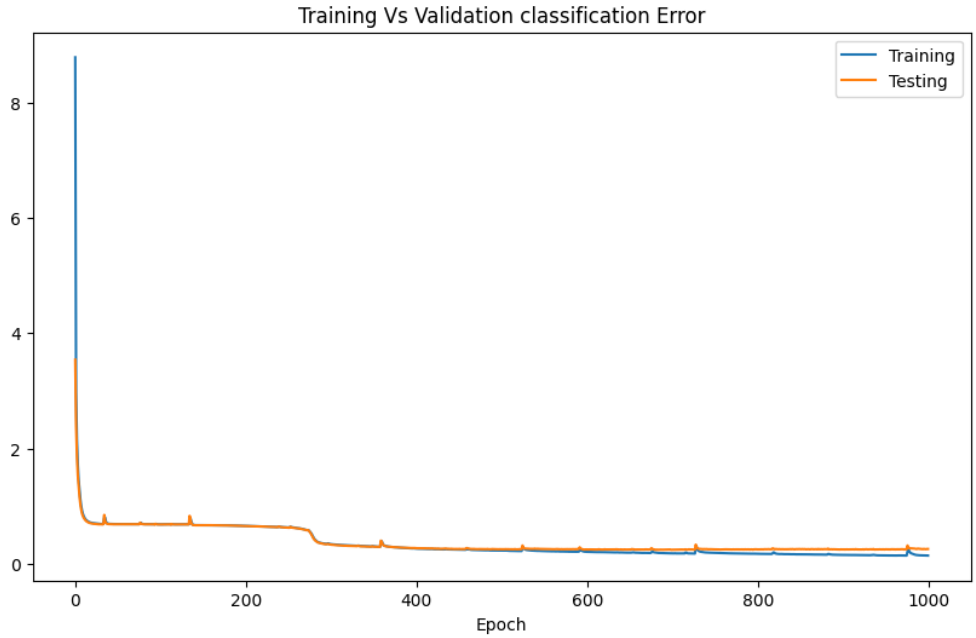

Figura 8 Error de clasificación por época - Online Shoppers Intention

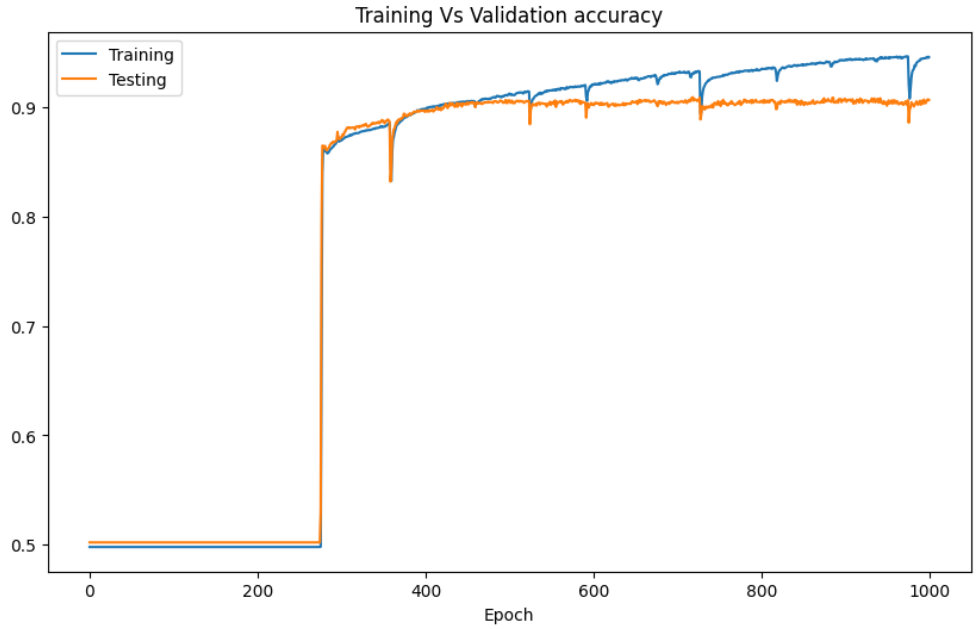

Figura 9 Precisión de entrenamiento y validación por época - Online Shoppers Intention

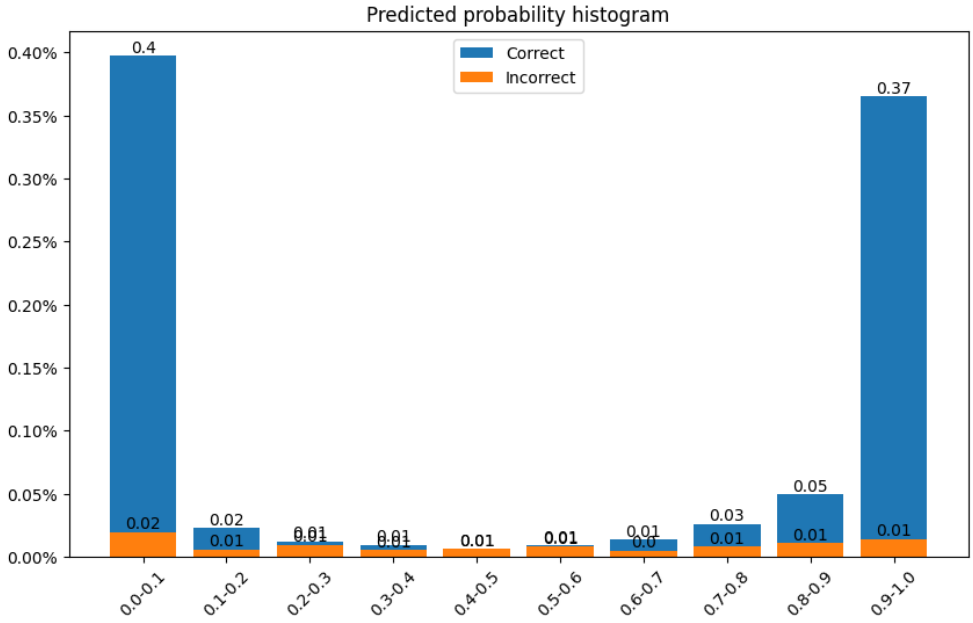

Figura 10 Histograma de predicciones - Online Shoppers Intention

5.2.1. Reporte de clasificación

	Precision	Recall	F1-score	Support
0	0.90	0.89	0.89	89
1	0.88	0.89	0.88	80
Accuracy			0.89	169
Macro avg	0.89	0.89	0.89	169
Weighted avg	0.89	0.89	0.89	169

Tabla 8 Reporte de clasificación - Online Shoppers Purchasing Intention

5.2.2. Prototipos

En la tabla Tabla 9 se puede apreciar el valor de cada característica o variable de los prototipos resultantes del modelo y destacando en azul las características con valores distintos entre los prototipos.

En las tabla Tabla 10, se pueden apreciar los valores de las características de los prototipos resultantes al igual que en la tabla Tabla 9, así como el valor de las caracteristicas para 2 observaciones, una de cada clase.

En la Tabla 11 se pueden observar la distancia euclidiana de las observaciones hacia cada prototipo y su clasificación real, estas distancias fueron calculadas en el espacio latente del *autoencoder*.

Esto es un ejemplo de cómo el enfoque basado en prototipos puede ayudar a implementar la interpretabilidad en un modelo de redes neuronales, con lo cuál es posible tener contexto de las predicciones resultantes del modelo y tener un mejor entendimiento del razonamiento o comportamiento del mismo al conocer los valores de las características de cada prototipo. Así mismo, recordando que las explicaciones de las predicciones obtenidas del modelo predictivo se basan en su similitud con los prototipos y no tanto en resaltar las partes relevantes de la observación.

Además, nos permite observar aquellas características que son importantes para la clasificación, ya que aquellas variables donde se tiene el mismo valor para todos los prototipos no representan un diferenciador que ayude a su clasificación.

En la Tabla 10 se puede observar al final de la tabla la cantidad de características de cada observación que son más cercanas a cada prototipo, excluyendo aquellos casos donde la variable en ambos prototipos tiene el mismo valor, así se puede determinar fácilmente a qué prototipo se parece más una observación, además de ayudar a obtener cierta certeza de qué variables son importantes para la clasificación debido a las variables cuyo valor es igual en ambos prototipos. De esta forma, el usuario podría tener un mejor entendimiento del comportamiento global del modelo.

	Prototype1	Prototype2
Administrative	3.500787	2.856980
Administrative_Duration	94.508987	94.508987
Informational	0.585032	0.585032
Informational_Duration	39.816223	39.816223
ProductRelated	41.173313	38.134892
ProductRelated_Duration	1725.785034	1463.158081
BounceRates	0.015140	0.018793
ExitRates	0.033478	0.044843
PageValues	32.576664	14.058432
SpecialDay	0.046092	0.065468
Month	6.440326	5.976314
OperatingSystems	2.105908	2.105908
Browser	2.387721	2.387721
Region	3.126048	3.296878
TrafficType	4.055720	4.389980
VisitorType	1.748806	1.977508
Weekend	0.296942	0.294041

Tabla 9 Prototipos del modelo entrenado - Online Shoppers Purchasing Intention

	Prototype1	Prototype2	Observation 1	Observation 2
Administrative	3.500787	2.856980	10.00	3.00
Administrative_Duration	94.508987	94.508987	261.87	25.57
Informational	0.585032	0.585032	0.80	1.00
Informational_Duration	39.816223	39.816223	44.59	0.00
ProductRelated	41.173313	38.134892	111.58	15.00
ProductRelated_Duration	1725.785034	1463.158081	3,412.81	2,026.48
BounceRates	0.015140	0.018793	0.01	0.01
ExitRates	0.033478	0.044843	0.03	0.07
PageValues	32.576664	14.058432	0.00	0.00
SpecialDay	0.046092	0.065468	0.00	0.00
Month	6.440326	5.976314	7.00	6.00
OperatingSystems	2.105908	2.105908	2.20	4.00
Browser	2.387721	2.387721	2.00	1.00
Region	3.126048	3.296878	1.00	1.00
TrafficType	4.055720	4.389980	10.84	1.00
VisitorType	1.748806	1.977508	2.00	2.00
Weekend	0.296942	0.294041	0.00	0.00
		Prototype1	8/17	5/17
		Prototype2	4/17	7/17
		Clasification	0	1

Tabla 10 Prototipos del modelo con 2 observaciones - Online Shoppers Purchasing Intention

	Prototipo1	Prototipo2	Clase real
Observación 1	0.16675067	0.07634607	0
Observación 2	0.03288352	0.25784305	1

Tabla 11 Distancia euclidiana de los prototipos a las 2 observaciones en el espacio latente - Online Shoppers Purchasing Intention

5.2.3. Interpretación de los resultados

Tomando como referencia los prototipos y resultados obtenidos de este conjunto de datos, se pueden observar que las variables más relevantes para los prototiposson: “Administrative”, “ProductRelated”, “ProductRelated_Duration”, “BounceRates”, “ExitRates”, “PageValues”, “SpecialDay”, “Month”, “Region”, “TrafficType”, “VisitorType”, “Weekend”.

Con las características anteriores se puede interpretar que el perfil de un usuario que concretó una compra en el sitio web debió haber visitado menos de 3 paginas de tipo administrativo, menos de 38 paginas de productos y haber pasado menos de 1463 minutos en ellas, salir del 1.8% o más de las páginas visitadas sin ninguna otra petición al servicio de *Google Analytics*, abandonar el 4.4% o más de las paginas visitadas sin ninguna otra petición al servicio de *Google Analytics* y siendo esta la última pagina visitada de la sesión, que el valor promedio de las páginas visitadas sea de 14.05, que el día de compra sea cercano a un día especial y que sea en la primera mitad del año y preferentemente en fin de semana, que esté ubicado de preferencia de la región 3, que el tipo de tráfico sea 4 y el usuario no deberá ser nuevo en el sitio.

5.3. Conjunto de datos 2: “South German Credit”

En las figuras Figura 11 y Figura 12, se pudo observar que el entrenamiento del modelo utilizando el conjunto de datos “South German Credit” con 10,000 épocas obtuvo un 76% de precisión, el cuál puede ser observado en el reporte de clasificación en la Tabla 12 donde también podemos observar que tiene un buen desempeño predictivo, aunque no tan alto como en el conjunto anterior posiblemente debido a la menor cantidad de muestras con las que cuenta el conjunto de datos, sin embargo, el desempeño es similar para ambas clases con 76% y 77% de F1.

Al igual que en los resultados del primer conjunto, en la Figura 13 se pudo observar el histograma de la distribución de las predicciones hechas por el modelo para este conjunto de datos.

En la misma figura se puede apreciar que el modelo tiende a dar resultados correctos en los extremos de 0 a .1 y de .9 a 1, 38% y 33% respectivamente, con un porcentaje de error de 21% en conjunto, mayor al porcentaje de error que se obtuve en el primer conjunto, posiblemente debido a que la cantidad de observaciones de este segundo conjunto de datos en menor.

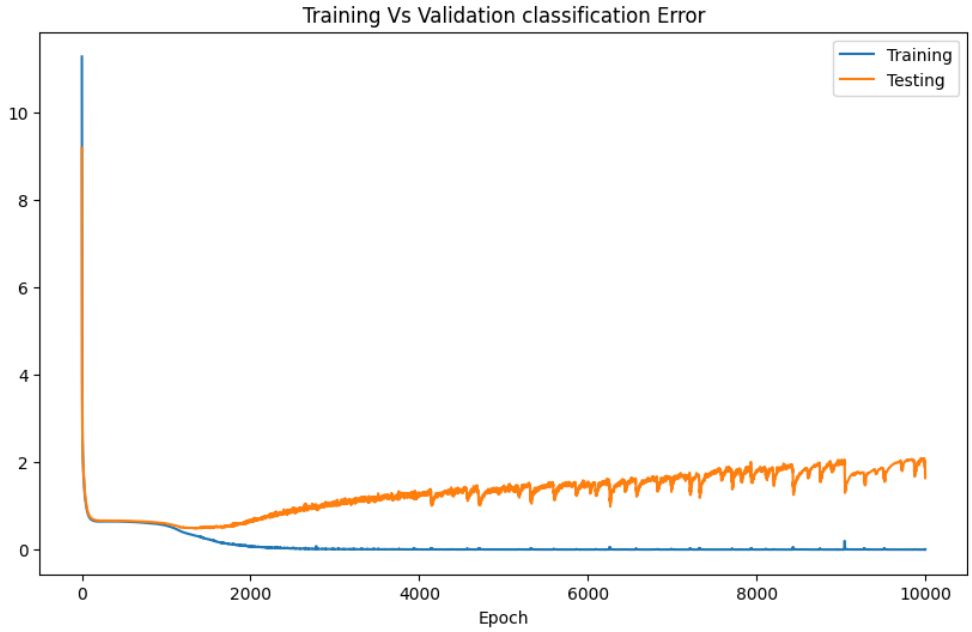

Figura 11 Error de clasificación por época - South German Credit

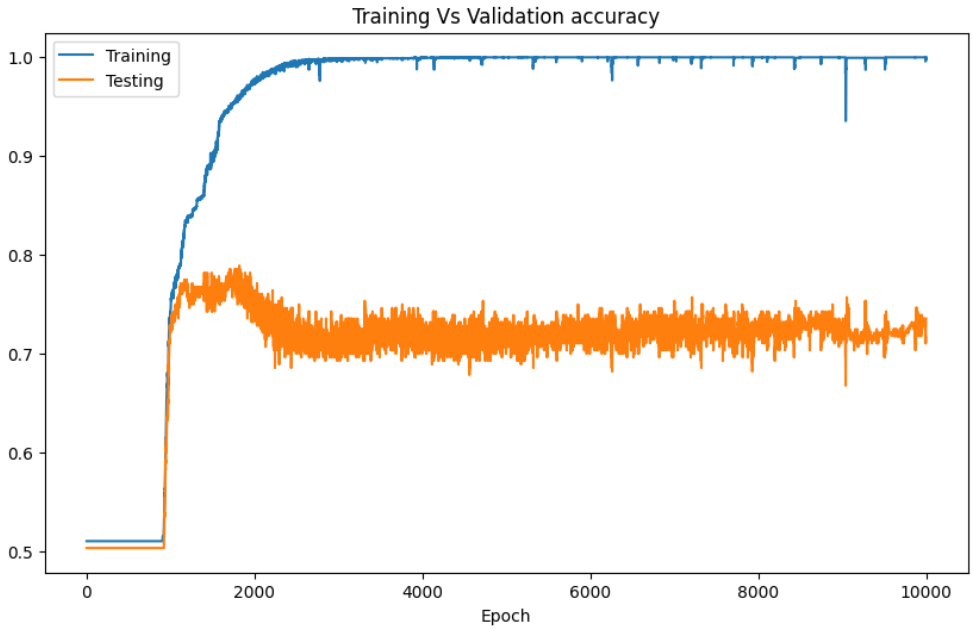

Figura 12 Precisión de entrenamiento y validación por época - South German Credit

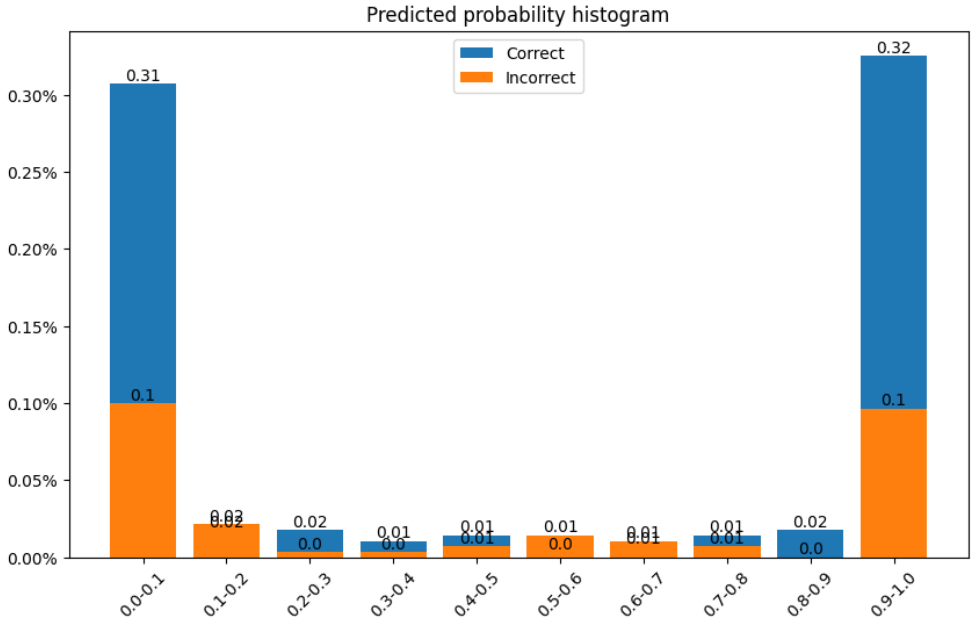

Figura 13 Histograma de predicciones - South German Credit

5.3.1. Reporte de clasificación

	Precision	Recall	F1-score	Support
0	0.73	0.74	0.74	139
1	0.74	0.73	0.74	141
Accuracy			0.74	280
Macro avg	0.74	0.74	0.74	280
Weighted avg	0.74	0.74	0.74	280

Tabla 12 Reporte de clasificación - South German Credit

5.3.2. Prototipos

En las tablas Tabla 13 y Tabla 14 se puede apreciar el valor de cada característica o variable de los prototipos resultantes del modelo y destacando las características con valores distintos entre los prototipos, así como el valor de las mismas para 2 observaciones, una de cada clase, además, en la Tabla 15 se pueden observar las distancias de las observaciones hacia cada

prototipo y su clasificación real, sin embargo, estas distancias fueron calculadas en el espacio latente del *autoencoder*.

Así como se mencionó en el caso de uso anterior, esto es otro ejemplo de cómo el enfoque basado en prototipos puede ayudar a implementar la interpretabilidad en un modelo de redes neuronales, con lo cuál es posible tener contexto de las predicciones resultantes del modelo y tener un mejor entendimiento del razonamiento o comportamiento del mismo al conocer los valores de las características de cada prototipo. Así mismo, recordando que las explicaciones de las predicciones obtenidas del modelo predictivo se basan en su similitud con los prototipos y no tanto en resaltar las partes relevantes de la observación

Además, nos permite observar aquellas características que son importantes para la clasificación, ya que aquellas variables donde se tiene el mismo valor para todos los prototipos no representan un diferenciador que ayude a su clasificación.

En la Tabla 14 se puede observar al final de la tabla la cantidad de características de cada observación que son más cercanas a cada prototipo, excluyendo aquellos casos donde la variable en ambos prototipos tiene el mismo valor, así se puede determinar fácilmente a qué prototipo se parece más una observación, además de ayudar a obtener cierta certeza de qué variables son importantes para la clasificación debido a las variables cuyo valor es igual en ambos prototipos. De esta forma, el usuario podría tener un mejor entendimiento del comportamiento global del modelo.

	Prototype1	Prototype2
status	2.447697	2.275714
duracion	22.046429	22.046429
credit_history	2.359286	2.359286
purpose	2.701429	2.701429
amount	3473.158691	3473.158691
savings	1.897143	1.897143
employment_duration	4.429590	3.251429
installment_rate	3.134840	2.987203
personal_status_sex	2.567143	2.567143
other_debtors	1.122857	1.122857
present_residence	2.915641	2.748571
property	2.324286	2.324286

age	35.121220	34.973572
other_installment_plans	2.960186	2.811593
housing	1.854286	1.854286
number_credits	1.330714	1.330714
job	2.832857	2.832857
people_liable	2.027346	2.019046
telephone	1.499078	1.332857
foreign_worker	1.998797	2.002094

Tabla 13 Prototipos del modelo entrenado - South German Credit

	Prototype1	Prototype2	3	5
status	2.447697	2.275714	2.0	4.0
duracion	22.046429	22.046429	29.0	24.0
credit_history	2.359286	2.359286	2.0	2.0
purpose	2.701429	2.701429	0.0	3.0
amount	3473.158691	3473.158691	2151.0	1311.0
savings	1.897143	1.897143	1.0	2.0
employment_duration	4.429590	3.251429	3.0	4.0
installment_rate	3.134840	2.987203	4.0	4.0
personal_status_sex	2.567143	2.567143	2.0	4.0
other_debtors	1.122857	1.122857	2.0	1.0
present_residence	2.915641	2.748571	2.0	3.0
property	2.324286	2.324286	3.0	2.0
age	35.121220	34.973572	24.0	26.0
other_installment_plans	2.960186	2.811593	1.0	3.0
housing	1.854286	1.854286	2.0	2.0
number_credits	1.330714	1.330714	1.0	1.0
job	2.832857	2.832857	3.0	3.0
people_liable	2.027346	2.019046	2.0	2.0
telephone	1.499078	1.332857	1.0	2.0
foreign_worker	1.998797	2.002094	2.0	2.0
		Prototype1	2/20	7/20
		Prototype2	7/20	2/20
		Clasification	0	1

Tabla 14 Prototipos del modelo con 2 observaciones - South German Credit

	Prototipo1	Prototipo2	Clase real
Observación 1	0.964665	0.079137	0
Observación 2	0.059708	0.803175	1

Tabla 15 Distancias de los prototipos a las 2 observaciones - South German Credit

5.3.1. Interpretación de los resultados

Como se mencionó anteriormente en la sección **¡Error! No se encuentra el origen de la referencia.**, utilizando los prototipos y resultados obtenidos en el segundo conjunto de datos "*South German Credit*" mostrados más adelante en la Tabla 14, se puede observar que las variables destacadas o relevantes para los prototipos son: "status", "employment duration", "installment rate", "present residence", "age", "other installment plans", "people liable", "telephone", "foreign worker".

Al observar los valores de las características de los prototipos se puede interpretar que el perfil del deudor que cumplirá satisfactoriamente el crédito debe tener: 4 años o más en su actual trabajo, que la mensualidad del crédito sea el 3.13% o más de sus ingresos, que tenga más de 2.9 años en su residencia actual, tener más de 35 años, no tener planes de pago a plazos con otros proveedores, tener 2 personas que dependan económicamente del deudor, tener un teléfono registrado a nombre del deudor y no ser un trabajador extranjero.

5.4. Discusión

Con los resultados obtenidos de los casos de uso, se pudo observar que efectivamente el uso de prototipos en modelos de redes neuronales aportan una capa que nos permite agregar la interpretabilidad al modelo de manera que se tiene un contexto global de las predicciones realizadas por éste al conocer las características de los prototipos utilizados para realizar dichas predicciones.

El espacio latente del *autoencoder* nos ayuda a encontrar las características más relevantes pero no ayudan a la interpretabilidad del modelo, sin embargo, el usuario podría hacer las comparaciones de las observaciones con los prototipos una vez que hayan sido traído al espacio dimensional de las observaciones utilizando la parte decodificadora del *autoencoder*.

6. CONCLUSIONES

Resumen: *En este capítulo se presentan las conclusiones y trabajo futuro en relación a la interpretabilidad en modelos de redes neuronales.*

6.1. Conclusiones

Se puede concluir que la interpretabilidad es una herramienta muy útil que permite entender las decisiones tomadas por los modelos predictivos, lo cual incrementa la confianza del usuario hacia éste al poder obtener un contexto de las predicciones, así como las características de los prototipos en las cuales el modelo se basa para realizar las predicciones, lo cuál también podría facilitar su inclusión en el flujo de trabajo, así mismo, podría ayudar a mejorar no sólo a los modelos al tener un mejor entendimiento del razonamiento del modelo, pero también los procesos de toma de decisiones basadas en datos [2].

6.2. Trabajo Futuro

Se podría establecer un marco de trabajo a través de la metodología planteada en este trabajo, así como la implementación de un algoritmo genético para modificar los híper parámetros y el tamaño de las capas del *autoencoder* con lo cual se intentaría buscar una mejor precisión del modelo predictivo.

Además, también se podría investigar formas alternativas y más complejas de dar explicaciones en torno a cada predicción hecha por el modelo, de manera que el usuario obtenga una retroalimentación más humana, lo cuál podría ayudar aún más en su aceptación e inclusión en un flujo de trabajo, al poder entender con mayor facilidad los motivos de dicha predicción.

BIBLIOGRAFÍA

[1] Z. C. Lipton, «The Mythos of Model Interpretability», *ArXiv160603490 Cs Stat*, jun. 2016, Accedido: sep. 09, 2019. [En línea]. Disponible en: http://arxiv.org/abs/1606.03490.

[2] A. Adadi y M. Berrada, «Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)», *IEEE Access*, vol. 6, pp. 52138-52160, 2018, doi: 10.1109/ACCESS.2018.2870052.

[3] Matt Turek, «Explainable Artificial Intelligence», *Explainable Artificial Intelligence (XAI)*. https://www.darpa.mil/program/explainable-artificial-intelligence (accedido jul. 09, 2020).

[4] B. Kim, «Interpretable Machine Learning: The fuss, the concrete and the questions», p. 125.

[5] S. Lundberg y S.-I. Lee, «A Unified Approach to Interpreting Model Predictions», p. 10, may 2017.

[6] N. Dhanachandra, K. Manglem, y Y. J. Chanu, «Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm», *Procedia Comput. Sci.*, vol. 54, pp. 764-771, ene. 2015, doi: 10.1016/j.procs.2015.06.090.

[7] G. J. Székely, M. L. Rizzo, y N. K. Bakirov, «Measuring and testing dependence by correlation of distances», *Ann. Stat.*, vol. 35, n.º 6, pp. 2769-2794, dic. 2007, doi: 10.1214/009053607000000505.

[8] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.* 2018.

[9] C. Rudin, «Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead», *Nat. Mach. Intell.*, vol. 1, n.º 5, pp. 206-215, may 2019, doi: 10.1038/s42256-019-0048-x.

[10] M. T. Ribeiro, S. Singh, y C. Guestrin, «"Why Should I Trust You?": Explaining the Predictions of Any Classifier», *ArXiv160204938 Cs Stat*, ago. 2016, Accedido: jul. 09, 2020. [En línea]. Disponible en: http://arxiv.org/abs/1602.04938.

[11] T. Hailesilassie, «Rule Extraction Algorithm for Deep Neural Networks: A Review», *ArXiv161005267 Cs*, sep. 2016, Accedido: jul. 10, 2020. [En línea]. Disponible en: http://arxiv.org/abs/1610.05267.

[12] T. Miller, «Explanation in Artificial Intelligence: Insights from the Social Sciences», *ArXiv170607269 Cs*, ago. 2018, Accedido: jul. 09, 2020. [En línea]. Disponible en: http://arxiv.org/abs/1706.07269.

[13] A. A. Freitas, «Comprehensible classification models: a position paper», *ACM SIGKDD Explor. Newsl.*, vol. 15, n.º 1, pp. 1–10, mar. 2014, doi: 10.1145/2594473.2594475.

[14] B. Kim, «Interactive and interpretable machine learning models for human machine collaboration», Thesis, Massachusetts Institute of Technology, 2015.

[15] T. Kulesza, M. Burnett, W.-K. Wong, y S. Stumpf, «Principles of Explanatory

Debugging to Personalize Interactive Machine Learning», en *Proceedings of the 20th International Conference on Intelligent User Interfaces*, Atlanta, Georgia, USA, mar. 2015, pp. 126–137, doi: 10.1145/2678025.2701399.

[16] V. Schetinin *et al.*, «Confident interpretation of Bayesian decision tree ensembles for clinical applications», 2007, doi: 10.1109/TITB.2006.880553.

[17] B. Kim, R. Khanna, y O. O. Koyejo, «Examples are not enough, learn to criticize! Criticism for Interpretability», en *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, y R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2280–2288.

[18] «Global AI software market size 2018-2025», *Statista*. https://www.statista.com/statistics/607716/worldwide-artificial-intelligence-market-revenues/ (accedido jul. 11, 2020).

[19] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, y N. Elhadad, «Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission», en *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, NSW, Australia, ago. 2015, pp. 1721–1730, doi: 10.1145/2783258.2788613.

[20] «Equifax Launches NeuroDecision® Technology». https://investor.equifax.com/news-and-events/press-releases/2018/03-26-2018-143044126 (accedido jul. 09, 2020).

[21] O. Li, H. Liu, C. Chen, y C. Rudin, «Deep Learning for Case-Based Reasoning through Prototypes: A Neural Network that Explains Its Predictions», *ArXiv171004806 Cs Stat*, nov. 2017, Accedido: nov. 25, 2019. [En línea]. Disponible en: http://arxiv.org/abs/1710.04806.

[22] F. Doshi-Velez y B. Kim, «Towards A Rigorous Science of Interpretable Machine Learning», p. 13.

[23] D. Martens, J. Vanthienen, W. Verbeke, y B. Baesens, «Performance of classification models from a user perspective», *Decis. Support Syst.*, vol. 51, n.º 4, pp. 782-793, nov. 2011, doi: 10.1016/j.dss.2011.01.013.

[24] «Interpretability of Machine Learning Models and Representations an Introduction.pdf». Accedido: jul. 19, 2020. [En línea]. Disponible en: https://www.researchgate.net/profile/Adrien_Bibal/publication/326839249_Interpretability_of_Machine_Learning_Models_and_Representations_an_Introduction/links/5b6861caa6fdcc87df6d58e4/Interpretability-of-Machine-Learning-Models-and-Representations-an-Introduction.pdf.

[25] A. Andrzejak, F. Langner, y S. Zabala, «Interpretable models from distributed data via merging of decision trees», en *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, abr. 2013, pp. 1-9, doi: 10.1109/CIDM.2013.6597210.

[26] M. Du, N. Liu, y X. Hu, «Techniques for interpretable machine learning», *Commun. ACM*, vol. 63, n.º 1, pp. 68–77, dic. 2019, doi: 10.1145/3359786.

[27] A. Rai, «Explainable AI: from black box to glass box», *J. Acad. Mark. Sci.*, vol. 48,

n.º 1, pp. 137-141, ene. 2020, doi: 10.1007/s11747-019-00710-5.

[28] M. T. Ribeiro, S. Singh, y C. Guestrin, «Model-Agnostic Interpretability of Machine Learning», *ArXiv160605386 Cs Stat*, jun. 2016, Accedido: sep. 09, 2019. [En línea]. Disponible en: http://arxiv.org/abs/1606.05386.

[29] B. Kim, C. Rudin, y J. Shah, «The Bayesian Case Model: A Generative Approach for Case-Based Reasoning and Prototype Classification», *ArXiv150301161 Cs Stat*, mar. 2015, Accedido: sep. 29, 2019. [En línea]. Disponible en: http://arxiv.org/abs/1503.01161.

[30] «tf.losses.softmax_cross_entropy | TensorFlow Core r1.15», *TensorFlow*. https://www.tensorflow.org/versions/r1.15/api_docs/python/tf/losses/softmax_cross_entropy?hl=zh-cn (accedido jul. 24, 2020).

[31] «tf.losses.sigmoid_cross_entropy | TensorFlow Core r1.15», *TensorFlow*. https://www.tensorflow.org/versions/r1.15/api_docs/python/tf/losses/sigmoid_cross_entropy?hl=zh-cn (accedido jul. 24, 2020).

[32] N. V. Chawla, K. W. Bowyer, L. O. Hall, y W. P. Kegelmeyer, «SMOTE: Synthetic Minority Over-sampling Technique», *J. Artif. Intell. Res.*, vol. 16, pp. 321-357, jun. 2002, doi: 10.1613/jair.953.

[33] C. O. Sakar, «Online Shoppers Purchasing Intention Dataset Data Set», *UCI Machine Learning Repository*, jun. 28, 2020. https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset (accedido jun. 28, 2020).

[34] C. O. Sakar, S. O. Polat, M. Katircioglu, y Y. Kastro, «Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks», *Neural Comput. Appl.*, vol. 31, n.º 10, pp. 6893-6908, oct. 2019, doi: 10.1007/s00521-018-3523-0.

[35] Anja Kipke, «South German Credit Data Set», *UCI Machine Learning Repository*, jun. 28, 2020. https://archive.ics.uci.edu/ml/datasets/South+German+Credit (accedido jun. 28, 2020).

[36] N. Supplied, «Kreditscoring zur Klassifikation von Kreditnehmern», 2010, doi: 10.5282/UBM/DATA.23.

[37] «sklearn.preprocessing.OrdinalEncoder — scikit-learn 0.23.1 documentation». https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html (accedido jul. 13, 2020).

[38] «sklearn.preprocessing.StandardScaler — scikit-learn 0.23.1 documentation». https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html (accedido jul. 13, 2020).

[39] «sklearn.cluster.KMeans — scikit-learn 0.23.1 documentation». https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html (accedido jul. 13, 2020).

[40] «imblearn.over_sampling.SMOTE — imbalanced-learn 0.5.0 documentation». https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTE.html (accedido

jul. 14, 2020).

[41] «THE APPLICATION OF CLUSTER ANALYSIS IN STRATEGIC MANAGEMENT RESEARCH: AN ANALYSIS AND CRITIQUE», p. 18.