Universidad Católica de Santa María

Facultad de Ciencias e Ingenierías Físicas y Formales

Escuela Profesional de Ingeniería de Sistemas

PREDICCIÓN DE RENUNCIA DE SOCIOS DE UNA COOPERATIVA UTILIZANDO TÉCNICAS SUPERVISADAS DE APRENDIZAJE AUTOMÁTICO

Tesis presentada por el Bachiller:

Linarez Gonzales, Alvaro Abraham

para optar por el Título Profesional:

Ingeniero de Sistemas: Especialidad en Ingeniería de Software

Asesor:

Dr. Sulla Torres, José

Arequipa – Perú
2019

FACULTAD DE CIENCIAS E INGENIERIAS FISICAS Y FORMALES

ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS

DICTAMEN DE BORRADOR DE TESIS

VISTO

El Borrador de Tesis titulado:

Prediccion de Renuncia de Socios de una cooperativa utilizando técnicas supervisadas de Aprendizaje Automático

Presentado por (el) (la) (los) Bachilleres

Alvaro Abraham Linarez Gonzales

Nuestro dictamen es:

Procedente

OBSERVACIONES: Ninguna

Arequipa, 21 de Noviembre de 2019

1748
Gillermo Caldas R

Jose Sulla T.

PRESENTACIÓN

Sr. Director de la Escuela Profesional de Ingeniería de Sistemas.

Sres. Miembros del Jurado.

De conformidad con las disposiciones del Reglamento de Grados y Títulos de la Escuela Profesional de Ingeniería de Sistemas, pongo a vuestra consideración el presente trabajo de investigación titulado: "PREDICCIÓN DE RENUNCIA DE SOCIOS DE UNA COOPERATIVA UTILIZANDO TÉCNICAS SUPERVISADAS DE APRENDIZAJE AUTOMÁTICO", el mismo que de ser aprobado me permitirá optar por el Título Profesional de Ingeniero de Sistemas.

LINAREZ GONZALES ALVARO ABRAHAM.

Dedicatoria

El presente trabajo de investigación va dedicado a mis padres, por sus consejos, apoyo, motivación y, sobre todo, por su amor incondicional.

Agradecimientos

Le agradezco al Dr. José Sulla Torres y al Dr. Guillermo Calderón Ruiz, por su asesoría y los conocimientos que me brindaron, al Mg. Jorge Córdova Ponce, presidente de la cooperativa COOSUNAT, por su apoyo y confianza brindada, y a todas las personas involucradas que hicieron posible la realización del presente trabajo de investigación.

RESUMEN

El presente trabajo de investigación tiene como objetivo la predicción de renuncia de socios de una cooperativa ubicada en la ciudad de Arequipa, mediante técnicas supervisadas de aprendizaje automático siguiendo una metodología personalizada. Se realizó el preprocesamiento de datos, se eligieron las técnicas idóneas para este caso de estudio y se aplicaron dichas técnicas con las librerías del lenguaje de programación Python. Como la cooperativa no tiene muchos datos y las técnicas requieren bastantes datos para una mejor precisión, se optó por utilizar datos generados sintéticamente correlacionados a los datos originales.

Se hizo un análisis de los resultados de las técnicas con los datos reales y los datos sintéticos en el que se determinó que la mejor técnica para este caso es de potenciación de gradiente con un 90% de precisión. Finalmente, para la validación de las técnicas se hizo una prueba con dos casos reales, el primero de un socio que renunció a la cooperativa y el segundo con un socio que se mantuvo en la cooperativa, la técnica que obtuvo el resultado correcto fue la entrenada con los datos sintéticos.

Palabras Clave

Aprendizaje automático, renuncia de socios, cooperativa, aprendizaje supervisado, datos sintéticos.

ABSTRACT

This research work aims to predict the customer churn of a credit union located in the city of Arequipa, through supervised automatic learning techniques following a personalized methodology. Data preprocessing was performed, the appropriate techniques were chosen for this case study and these techniques were applied with the Python programming language libraries. As the credit union does not have much data and the techniques require enough data for better accuracy, it was decided to use data generated synthetically correlated to the original data.

An analysis of the results of the techniques was made with the real data and the synthetic data in which it was determined that the best technique for this case is gradient enhancement with 90% accuracy. Finally, for the validation of the techniques, a test was executed with two real cases, the first of a member who resigned from the credit union and the second with a partner who remained in the credit union, the technique that obtained the correct result was the trained with the synthetic data.

Keywords

Machine learning, customer churn, credit union, supervised learning, synthetic data.

INTRODUCCIÓN

Desde Alexa de Amazon, hasta autos que conducen solos, el desarrollo tecnológico en el campo de la inteligencia artificial está progresando cada vez más, ya no es cada diez años como sucedió en el siglo XX, sino año tras año e incluso en períodos más pequeños de doce meses como lo indican Karppi y Granata (2019). Estos avances afectan en gran medida la vida cotidiana de las personas, a través de la publicidad específica que se muestra en Internet, predicción del valor de las acciones de la bolsa de valores en un futuro próximo, entre otros. Demostrando que el campo de aplicación de la inteligencia artificial es muy amplio e importante porque también puede usarse para aumentar el desarrollo de un país, ya que según Iqbal, Saleem y Naseer (2018), puede usarse en el campo académico, social y financiero.

Según SAS (2017), el aprendizaje automático es una rama de la inteligencia artificial basado en la idea de que los sistemas pueden aprender de datos, identificar patrones y tomar decisiones con intervención humana mínima. Existen técnicas de aprendizaje automático que pueden ser muy útiles para el ámbito financiero, porque proporcionan probabilidades y porcentajes a través de una matriz de confusión que encuentra precisión, sensibilidad y especificidad que contribuyen a obtener mejores resultados en el análisis de la información como lo indican Boutaba, Salahuddin, Limam, Ayoubi, Shahriar, Estrada-solano y Caicedo (2018).

Por lo tanto, puede ser muy útil encontrar mejores resultados cuando se aplica en la vida real, para probarlo, se tomó un caso de estudio de predicción de renuncia de clientes, para este caso, la predicción de renuncia de socios de una cooperativa de ahorro y crédito utilizando técnicas supervisadas de aprendizaje automático, aplicando una metodología personalizada para este caso.

La verificación de estas técnicas fue realizada al comparar los resultados de los datos reales con los resultados de datos generados sintéticamente, es importante indicar que estos datos sintéticos son correlacionados a los datos originales.

Finalmente, para la validación de las técnicas se realizó una prueba con datos de un socio real que se retiró de la cooperativa y un socio que se mantuvo en ella y observar el resultado de las técnicas entrenadas con datos reales y datos sintéticos en cada uno de estos casos, y así comprobar la exactitud de la predicción realizada.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Capítulo 1: Desarrollo del Trabajo de Investigación

1.1. Estado del Arte

Para ampliar y profundizar el estado del arte presentando en el Plan de Tesis (ver Apéndice A), se presentan los antecedentes científicos del trabajo de investigación.

Las cooperativas en el Perú son muy importantes para el desarrollo económico, además de la inclusión financiera que logran porque llegan a los sectores más lejanos y vulnerables del país.

Según el Ministerio de Producción (2019), una cooperativa es una organización que agrupa a varias personas con la finalidad de realizar una actividad empresarial. Su modo de operar se basa en la cooperación de todos los socios. Todos "cooperan" para poder beneficiarse en forma directa para obtener un bien, un servicio o trabajo en mejores condiciones.

Partiendo de este concepto de cooperación, un grupo de trabajadores de la SUNAT fundaron la "Cooperativa de Ahorro y Crédito de Oficiales de la SUNAT" con nombre abreviado "COOSUNAT", donde cualquier trabajador de la SUNAT ya sea cesante, jubilado, sin distinción de régimen laboral o tipo de contrato puede inscribirse como socio.

La cooperativa a pesar de tener pocos años de haber sido fundada, ha crecido a pasos agigantados, ha logrado llegar a la categoría “Nivel II A” por tener un capital de más de dos millones y medio de soles.

Los importes de los socios a la cooperativa son descontados de sus sueldos brutos de la SUNAT, los socios eligen la cantidad que desean que se les descuente mediante una solicitud a mesa de partes de cualquier sucursal de la SUNAT.

Pero a pesar de todos los beneficios que la cooperativa ofrece, existen socios que no se adecúan o que no desean seguir aportando, entre otros motivos, derivando en la renuncia de éstos a la cooperativa, generando una tasa de pérdida de clientes.

Bernazzani (2018), indica que la tasa de pérdida de clientes es el porcentaje de los clientes o suscriptores de una empresa que cancelan o no renuevan su suscripción durante un periodo de tiempo dado. La tasa de pérdida es una medición critica para empresas cuyos clientes pagan de manera frecuente o aquellas basadas en suscripciones.

Hay diversos métodos para reducir la tasa de pérdida de clientes como: comenzar con el pie derecho con el cliente, solicitar feedback en momentos clave, comunicarse activamente con el cliente y analizar la tasa de pérdida cuando ocurra para mejorar el servicio al cliente.

Aparte de los métodos señalados por Bernazzani, en la actualidad existen técnicas para predecir si un cliente dejará o no una empresa en un tiempo determinado mediante información histórica de éstos utilizando técnicas de aprendizaje automático como es el caso de las redes neuronales artificiales, todo con el objetivo de lograr un mejor análisis y reducir la tasa de pérdida de clientes.

Según Brownlee (2016), aprendizaje automático es un subcampo de aprendizaje automático relacionado con algoritmos inspirados en la estructura y funcionamiento del cerebro llamados redes neuronales artificiales.

La cooperativa tiene la necesidad de saber cuándo y debido a qué motivo un socio va a dejar la cooperativa, para satisfacer esta necesidad, se empleará un sistema inteligente que mediante redes neuronales artificiales se determinará el porcentaje de que un socio dejaré o no la cooperativa.

López y Pastor (2014) desarrollaron un modelo de redes neuronales para estudiar la quiebra de los bancos estadounidenses. Gracias al modelo los inversores, depositantes y otros participantes en los mercados de capital pueden evaluar el perfil de riesgo de su inversión. Aunque este modelo tiene sus limitaciones como la necesidad de muchos cálculos para la salida, visualización completa y el no poder controlar los factores macroeconómicos que puedan afectar la predisposición de los bancos a quebrar.

Un ejemplo de uso de redes neuronales artificiales para la predicción de datos es el de Funahashi y Horiuchi (2017), ellos necesitaban predecir el contenido de agua de la mantequilla y modelar las características del proceso de batido a partir de una red neuronal artificial. Gracias al uso de este modelo, se pudo realizar un gran análisis, concluyendo que el control de la temperatura de alimentación de la crema es muy importante en la fabricación de mantequilla.

Vafeiadis, Diamantaras, Sarigiannidis, Chatzisavvas (2015), realizaron una comparación de técnicas de aprendizaje automático para predecir la renuncia de clientes en la industria de telecomunicaciones. Los métodos con mejor rendimiento fueron la Red de Propagación hacia atrás y el Árbol de Decisión, ambos métodos lograron una precisión de 94% y 77% respectivamente.

Faris (2018), realizó un modelo híbrido basado en optimización de enjambre de partículas y en una red neuronal de la industria de telecomunicaciones para predecir la tasa de renuncia de clientes. La optimización de enjambre de partículas fue utilizada para mejorar los pesos de las variables de entrada y optimizar la estructura de la red neuronal simultáneamente para incrementar el poder de la precisión. Se basaron en dos conjuntos de datos de dos compañías de telecomunicaciones obteniendo como resultado que el modelo propuesto puede incrementar significativamente la tasa de cobertura de la renuncia de clientes en comparación a otros clasificadores, además indica que la automática optimización de la red neuronal elimino el esfuerzo que se necesitaba para obtener el mejor número de capas ocultas de la red neuronal.

Khalid, Ridwan, Makhtar, Nordin, y Rasid (2018), realizaron una comparación de algoritmos de redes neuronales para la predicción de renuncia de clientes para una compañía de telecomunicaciones de Malasia. Los algoritmos comparados fueron Propagación posterior de Levenberg Marquardt, retro propagación BFGS Quasi-Newton, propagación posterior de gradiente conjugado con actualizaciones Fletcher-Reeves. Su análisis mostró que la red neuronal entrenada con el algoritmo Levenberg Marquart obtuvo la mayor precisión con un 94.82%, además concluyeron que todas los predictores comparados son aceptables para la predicción de tasa renuncia de clientes. El modelo óptimo de red neuronal para los autores consiste de catorce variables de entrada, un nodo oculto y una variable de salida con el algoritmo de Levenberg Marquart.

Kim, Lee y Mun (2018), desarrollaron un modelo para predecir las mareas de tormenta en Corea del Sur utilizando una Red Neuronal Artificial, para ello se emplearon datos históricos de 59 tormentas que sucedieron entre 1978 y 2014 en dicho país. Las variables de entrada fueron longitud, velocidad de movimiento, latitud, dirección de rumbo, presión central, radio de velocidad del viento y velocidad máxima del viento. Para medir el rendimiento de este modelo se expresó como el coeficiente de correlación, los coeficientes máximos y mínimos fueron 0.861 y 0.979 respectivamente.

Kumar S. y Kumar M. (2019), mediante redes neuronales artificiales y con diferentes funciones de activación realizaron la predicción de la tasa de renuncia de clientes de un conjunto de datos de una empresa de telecomunicaciones para determinar los factores que influyen en los clientes para su renuncia, logrando una alta precisión de más del 80%.

Maleki, Sorooshian, Goudarzi, Baboli, Birgani y Rahmati (2019), realizaron un estudio para evaluar la efectividad de una Red Neuronal Artificial para predecir las concentraciones de contaminantes atmosféricos por hora y dos índices de calidad del aire que son Índice de Calidad del Aire (AQI) e Índice de Salud de la Calidad del Aire (AQHI) en la ciudad de Ahvaz, Irán. Determinaron que los valores de coeficiente de correlación y el error cuadrático medio fueron 0.87 y 59.9 respectivamente. Además, concluyeron que la aplicación de una Red Neuronal Artificial es factible para ciudades como Ahvaz para pronosticar la calidad del aire con la finalidad de prevenir los efectos en la salud.

Gentiluomo, Roessner, Augustijn, Svilenov, Kulakova, Mahapatra, Winter, Streicher, Rinnanc, Peters, Harris y Frieß (2019), utilizaron Redes Neuronales Artificiales para predecir las propiedades biofísicas de los anticuerpos monoclonales terapéuticos (temperatura de fusión, temperatura de inicio de agregación, parámetro de interacción y la concentración de sal de la composición de aminoácidos). Al solo usar la composición de aminoácidos mantuvieron Redes Neuronales Artificiales simples, permitiendo una alta aplicabilidad general, robustez e interpretabilidad. Los autores obtuvieron 0.94% como resultados de coeficiente de correlación y alrededor de 20% fue el error cuadrático obtenido.

Un ejemplo de uso de redes neuronales recurrentes en la predicción de renuncia de clientes fue el realizado por Zolidah, Zaidah y Syahir (2014), ellos implementaron una red neuronal recurrente de Elman y una red neuronal recurrente de Jordan con aprendizaje de refuerzo para predecir la tasa de renuncia de los usuarios de celulares. El proyecto pudo

demostrar que la red neuronal recurrente de Jordan proporciona una mejor precisión que la red neuronal recurrente de Elman.

En el ámbito de cooperativas, Vasnconcellos, Arte, Ayres y Fonseca (2019), analizaron la puntuación de créditos de una cooperativa en Brasil utilizando el método de regresión logística y bosques aleatorios que son técnicas de aprendizaje automático, concluyendo que la técnica de bosque aleatorio funciona mejor que el método de regresión logística para la puntuación crediticia.

A nivel nacional, Sulla (2015), utilizó técnicas supervisadas de minería de datos para poder predecir la deserción de estudiantes de una universidad, para ello utilizo varias técnicas supervisadas de minería de datos árboles de decisión, redes neuronales, redes bayesianas, entre otras. Llegando a la conclusión de que los estudiantes que abandonan la universidad tienen solo 11 cursos aprobados y un promedio final de tan solo 7.84%.

Córdova (2017), realizó un sistema de predicción que tenía el objetivo de predecir la muerte y sobrevida de pacientes del hospital Honorio Delgado de Arequipa mediante técnicas de redes neuronales de tipo propagación hacia atrás, clasificadores bayesianos y máquinas de vectores de soporte.

Dada la investigación realizada se puede concluir que la predicción de datos facilita mucho la toma de decisiones por parte de las organizaciones, gracias al uso de las redes neuronales artificiales se pueden llegar a obtener resultados precisos y confiables, y no solo

esto, también se puede hallar patrones complejos en los datos que a simple vista es muy difícil de detectar, logrando un análisis más completo.

Este proyecto estará enfocado en analizar técnicas supervisadas de aprendizaje automático y mediante una serie de verificaciones y validaciones, seleccionar la que logre la mayor precisión en la predicción de renuncia un socio de la cooperativa COOSUNAT. Se logrará facilitar la toma de decisiones de la gerencia y mejorar el servicio que tiene la cooperativa con los socios.

1.2. Bases Teóricas de la Investigación

Para el presente trabajo de investigación se requiere el desarrollo de unos conceptos teóricos en alusión a la inteligencia artificial, cooperativas, entre otros temas para una mejor comprensión.

1.2.1. Inteligencia Artificial

Desde SIRI a autos que se manejan solos, la inteligencia artificial está evolucionando rápidamente. Mientras que las películas presentan a la inteligencia artificial como robots con características humanas, en la realidad esta abarca desde cualquier algoritmo de Google hasta armas autónomas.

Según Future of Life (2016), Inteligencia Artificial es la aplicación rápida de procesamiento de datos, análisis predictivo y aprendizaje automático para simular el comportamiento y la capacidad de resolver problemas con máquinas y software.

Se podría decir que es la "inteligencia" de las máquinas y programas computacionales contra la inteligencia humana y de los animales. Las máquinas y los programas que usan inteligencia artificial están diseñadas para leer e interpretar la entrada de datos y responder a esta usando análisis predictivo o aprendizaje automático.

Aplicaciones	Modelos	Software/ Hardware	Lenguajes de Programación
• Reconocimiento de imágenes • Reconocimiento de voz • Chatbots • Generación de lenguaje natural	• Aprendizaje Automático • Aprendizaje Profundo • Redes Neuronales	• GPUs • Almacenamiento de datos en la nube	• Python • R

Figura 1. Componentes de la Inteligencia Artificial

A. Tipos de Inteligencia Artificial

Según Rouse (2018), la IA se clasifica en cuatro tipos, desde los sistemas de IA que existen hoy en día, hasta sistemas inteligentes que todavía no existen.

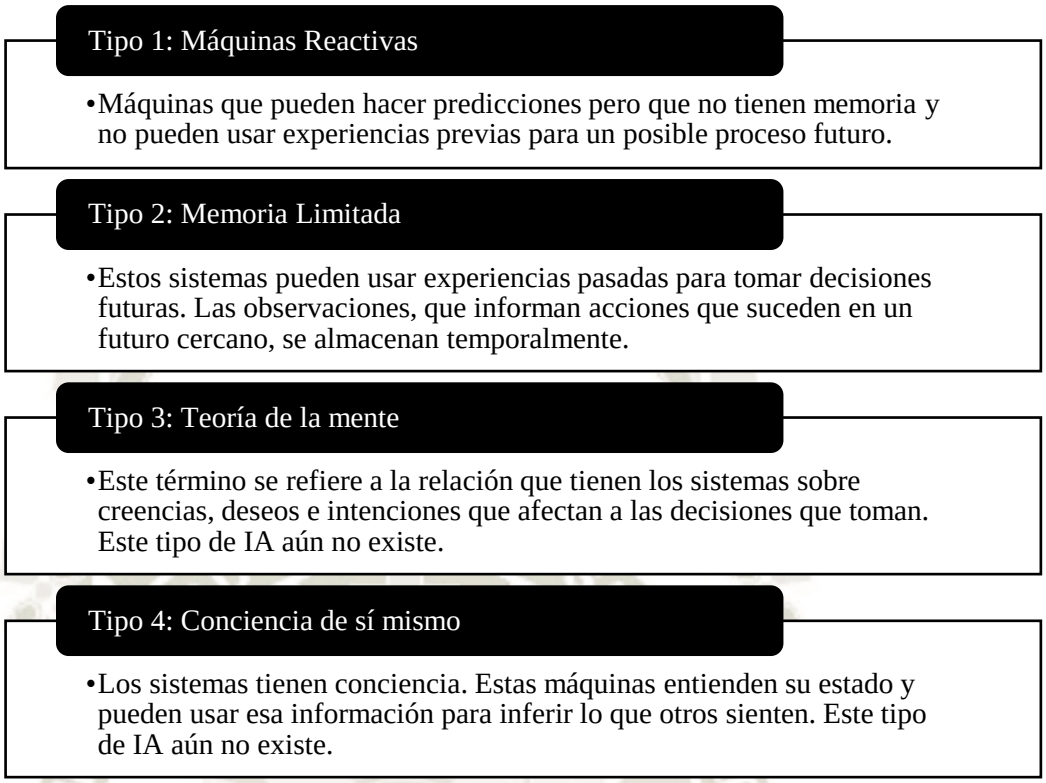


Figura 2. Tipos de Inteligencia Artificial

B. ¿Cómo "funciona" la IA?

DeepAi (2017), indica que la IA "funciona" al combinar varios enfoques para resolver problemas como las matemáticas, estadísticas computacionales, aprendizaje automático y análisis predictivo.

Un sistema de IA común toma un conjunto de datos como entrada y lo procesará rápidamente usando algoritmos inteligentes que aprenden y mejoran cada vez que un nuevo conjunto de datos es procesado. Después de que se completa el entrenamiento, el modelo producido, si es entrenado satisfactoriamente, será capaz de predecir o revelar información específica de nuevos datos.

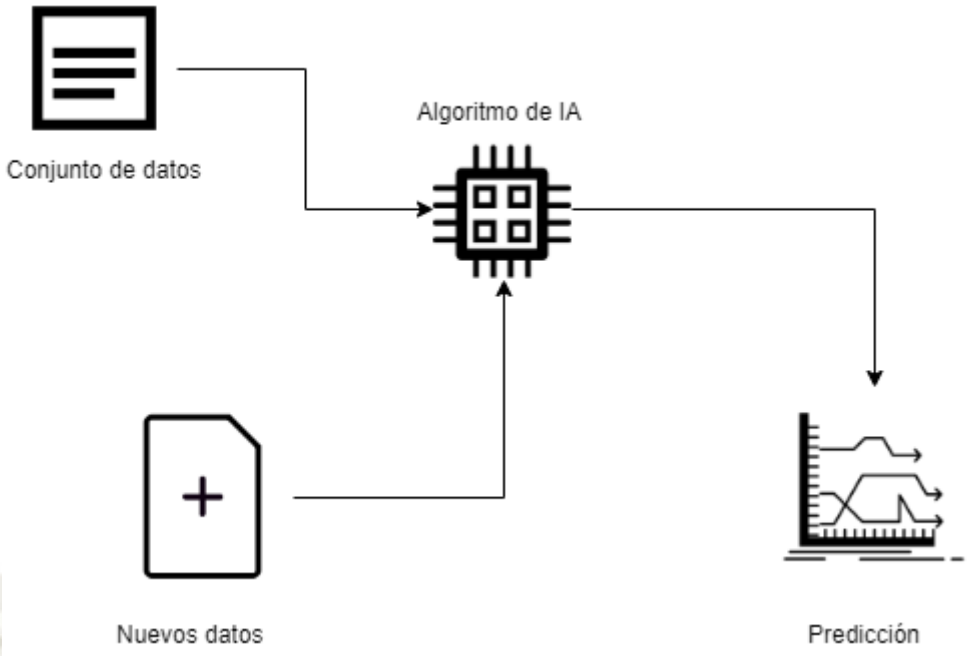


Figura 3. Funcionamiento de la Inteligencia Artificial

1.2.2. Aprendizaje Automático

Según SAS (2017), el aprendizaje automático es un método de análisis de datos que automatiza la construcción de modelos analíticos. Es una rama de la IA basado en la idea de que los sistemas pueden aprender de datos, identificar patrones y tomar decisiones con intervención humana mínima.

A. ¿Por qué es necesario el aprendizaje automático?

El aprendizaje automático es necesario para tareas que son muy complicadas de programar directamente, por ejemplo, tareas que resultan imprácticas por no decir imposibles, de resolver y codificar; así que, para resolver esa situación, se utiliza un algoritmo de aprendizaje automático con una gran cantidad de datos, que se le permite explorar y buscar un modelo que pueda satisfacer la necesidad de los programadores.

Algunos ejemplos de tareas que son resueltas y, con resultados más precisos gracias a las técnicas de aprendizaje automático son:

- Reconocimiento de patrones: expresiones faciales, reconocimiento de rostros, objetos y palabras, entre otros.
- Reconocimiento de anomalías: secuencias inusuales de transacciones bancarias, patrones inusuales que derivan a enfermedades, entre otros.
- Predicción: futuros precios de acciones o cambios de moneda, qué película le gustará más a una persona, entre otros.

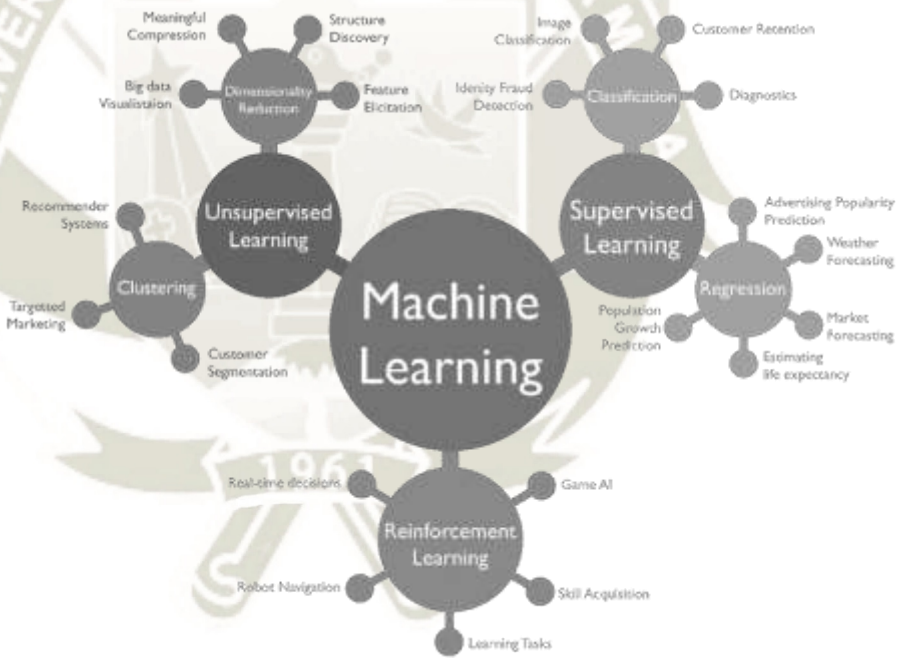


Figura 4. Ramas del Aprendizaje Automático según SAS (2017)

B. ¿Quiénes utilizan aprendizaje automático?

Según SAS (2017), el aprendizaje automático lo utilizan:

Servicios Financieros

- Los bancos y otras empresas del sector financiero usan el aprendizaje automático por dos razones: identificar datos importantes y prevenir fraudes.

Gobierno

- La enorme cantidad de datos que maneja el gobierno puede utilizarse para aumentar eficiencia, ahorrar dinero, minimizar el robo de identidad, todo gracias a la ayuda del aprendizaje automático.

Cuidado de la salud

- Gracias al aprendizaje automático, se pueden analizar datos para evaluar la salud de un paciente en tiempo real.

Ventas

- Las empresas de ventas utilizan aprendizaje automático para capturar datos, analizarlos y usarlos para personalizar la experiencia de compra, implementar campañas de márketing, optimizar precios, entre otros.

Aceite y Gas

- La cantidad de usos del aprendizaje automático para estas industrias es vasto y sigue expandiéndose, como por ejemplo: encontrar nuevos recursos de energía, analizar minerales, predicción de fallas de sensores de refinería, entre otros.

Transporte

- Analizar datos para identificar patrones y tendencias es la clave para la industria del transporte, basado en realizar rutas más eficientes y predecir posibles problemas para aumentar la rentabilidad.

Figura 5. Aplicaciones del Aprendizaje Automático

C. Datos Etiquetados

Según Techopedia (2016), los datos etiquetados son una designación para datos que han sido etiquetados con una o más etiquetas que identifican ciertas propiedades o características, clasificaciones u objetos contenidos. Por ejemplo, las etiquetas pueden indicar si una imagen contiene un perro o un gato, qué palabras se pronunciaron en una grabación de audio, qué tipo de acción se realiza en un video, cuál es el tema de un artículo de noticias, entre otros.

D. Aprendizaje Supervisado

Soni (2018), indica que los algoritmos de aprendizaje supervisado son entrenados usando datos etiquetados como entrada donde la salida deseada ya es conocida. Por ejemplo, una herramienta de entrenamiento puede tener datos etiquetados como “F” (Fallado) u “O” (Operativo). El algoritmo de aprendizaje recibe un conjunto de entradas junto con sus respectivas salidas y el algoritmo aprende al comparar la salida actual con la salida correcta para encontrar errores modificando el modelo.

Mediante métodos como clasificación, regresión, predicción y potenciación del gradiente, el aprendizaje supervisado utiliza patrones para predecir eventos futuros. Por ejemplo, puede anticipar cuando las transacciones con tarjeta de crédito son fraudulentas o cuando un cliente puede presentar un reclamo.

El aprendizaje supervisado generalmente es utilizado en el contexto de clasificación para asignar entradas a etiquetas de salida o en el contexto de regresión, cuando se requiere asignar entradas a una salida continua.

Los algoritmos más utilizados en el aprendizaje supervisado son: regresión logística, clasificador bayesiano, máquinas de vector de soporte, redes neuronales artificiales y bosques aleatorios.

Tanto en regresión como en clasificación, el objetivo es encontrar relaciones específicas o estructuras en los datos de entrada que permitan producir datos de salida correctos de forma efectiva. Es importante mencionar que los datos de salida correctos son determinados netamente de los datos entrenados, a pesar de que se tenga una verdad fundamental que el modelo asumirá como verdadera, no quiere decir que las etiquetas de datos siempre sean correctas en situaciones del mundo real. Las etiquetas de datos ruidosas o incorrectas claramente harán una reducción de la efectividad del modelo.

Clasificación Regresión

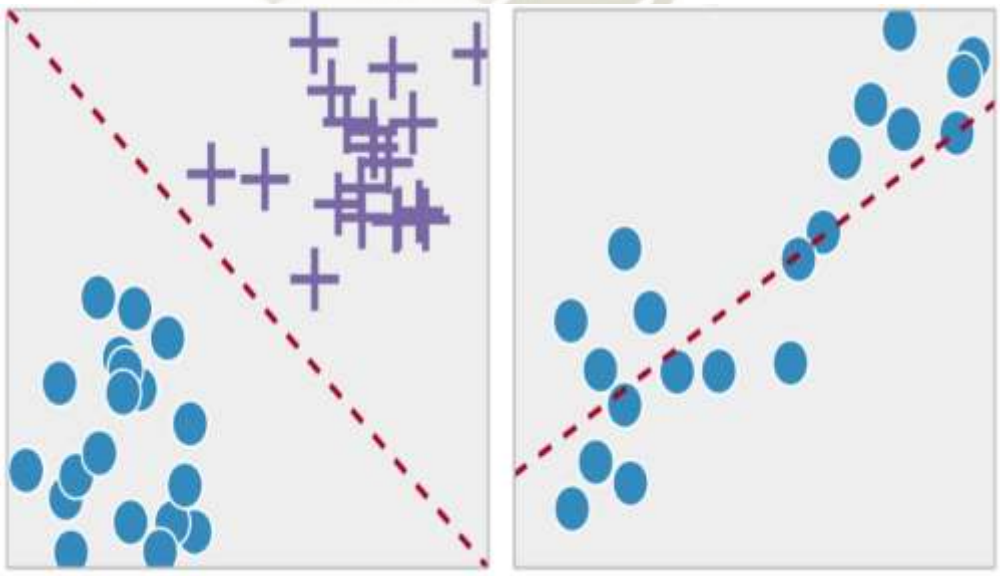


Figura 6. Clasificación y Regresión

E. Aprendizaje No Supervisado

El aprendizaje no supervisado es utilizado con datos que no tienen etiquetas históricas, el sistema no recibe la "respuesta correcta". Los algoritmos no supervisados deben "averiguar" qué es lo que se muestra.

Según Soni (2018), el objetivo del aprendizaje no supervisado es explorar los datos y encontrar alguna estructura dentro de ellos, además es importante puntualizar que este aprendizaje funciona bien con datos transaccionales. Por ejemplo, se puede identificar algunos clientes con atributos similares para hacerles una campaña publicitaria específicamente para ellos o también encontrar los patrones entre ellos.

Las técnicas más populares de aprendizaje no supervisado son: mapas autoorganizados, k vecinos más cercanos y k-means. Estos algoritmos son también utilizados para segmentar textos, recomendar artículos e identificar datos atípicos.

Las tareas más comunes dentro del aprendizaje no supervisado son la agrupación, el aprendizaje de representación y la estimación de densidad. En todos estos casos, es necesario conocer la estructura de los datos sin utilizar etiquetas que han sido proporcionadas de manera explícita. Algunos algoritmos comunes incluyen la agrupación de k-means, análisis de componentes principales y autoencoders. Como no se proporcionan etiquetas, no existe una manera específica de comparar el rendimiento del modelo en la mayoría de los métodos de aprendizaje no supervisados.

Dos casos de uso de aprendizaje no supervisado son el análisis exploratorio y la reducción de la dimensionalidad.

El aprendizaje no supervisado es muy útil en el análisis exploratorio ya que puede identificar de forma automática la estructura en los datos. Por ejemplo, si un analista intentara clasificar a los clientes, los métodos de agrupamiento no supervisados serian un gran punto de partida para el análisis. En situaciones donde es prácticamente imposible o no práctico para las personas procesar tendencias en los datos, el aprendizaje no supervisado provee información inicial que luego puede utilizarse para probar hipótesis individuales.

La reducción de la dimensionalidad, que se refiere a los métodos utilizados para representar datos con menor cantidad de columnas o características, se puede lograr con métodos de aprendizaje no supervisados. En el aprendizaje de representación, se requiere aprender las relaciones entre las características individuales, lo que permite representar los datos usando las características que se interrelacionan con las características iniciales. Esta estructura es a menudo representada usando menos características que con las que se empezó, por lo que puede hacer que el procesamiento de datos adicionales sea menos intenso y pueda eliminar características redundantes.

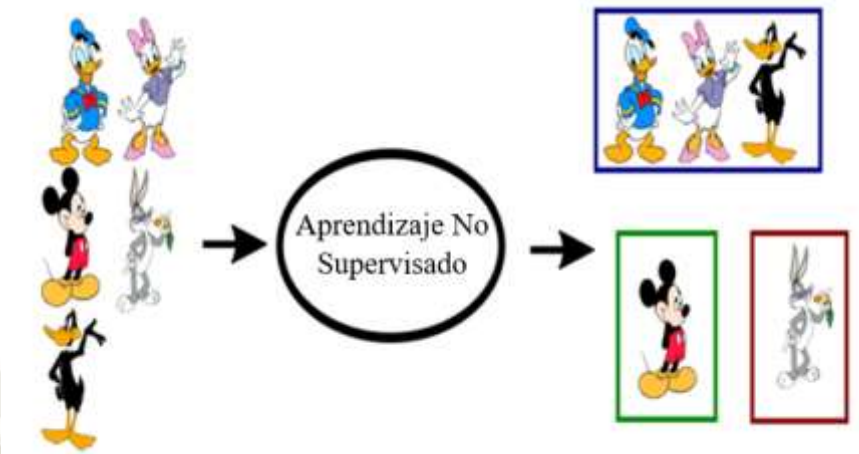


Figura 7. Ejemplo de agrupamiento del aprendizaje no supervisado como indica Soni (2018)

F. Aprendizaje por Refuerzo

El aprendizaje por refuerzo usualmente es utilizado para la robótica, desarrollo de videojuegos y navegación. Con el aprendizaje por refuerzo, los algoritmos descubren a través de pruebas y errores qué acciones producen las mayores recompensas.

SAS (2017), indica que este tipo de aprendizaje tiene tres componentes principales: el agente (el aprendiz o quien toma decisiones), el entorno (todo con lo que el agente interactúa) y acciones (lo que el agente puede hacer). El objetivo es que el agente pueda elegir acciones que maximicen la recompensa esperada durante un período de tiempo determinado. El agente alcanzará el objetivo más rápido siguiendo una buena política, así que el objetivo del aprendizaje por refuerzo es aprender la mejor política.

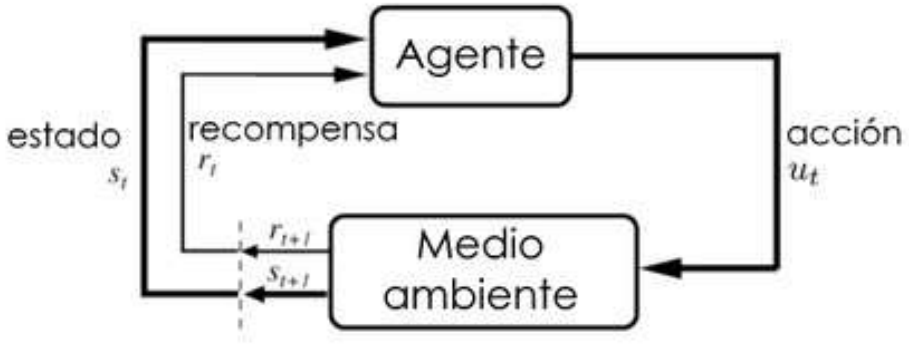


Figura 8. Funcionamiento del aprendizaje por refuerzo

1.2.3. Redes Neuronales Artificiales

Las redes neuronales artificiales son una de las herramientas más utilizadas en aprendizaje automático. Dormehl (2019), indica que las redes neuronales artificiales son sistemas inspirados en cómo funciona el cerebro, pretendiendo replicar la manera en que los humanos aprenden. Las redes neuronales artificiales consisten en capas de entrada, capas oculta y capas de salida, son herramientas muy buenas para encontrar patrones muy grandes o muy complejos para que un humano pueda extraerlos y hacer aprender a la máquina a reconocerlos.

Las redes neuronales (también llamadas “perceptrones”), han existido desde los años 40, ha sido recién en las últimas décadas donde éstas han tomado mayor relevancia para la inteligencia artificial. Esto es gracias a la técnica llamada “propagación hacia atrás”, que permite a las redes neuronales ajustar sus capas ocultas a situaciones donde la salida no coincide con lo que el creador espera, como una red diseñada para reconocer perros que identifica de manera errada un gato, por ejemplo.

Otro avance importante ha sido la llegada de las redes neuronales de aprendizaje profundo, en donde las diferentes capas de una red multicapa extraen diferentes características hasta que se pueda reconocer lo que se está buscando.

Un ejemplo sencillo para explicar el proceso de aprendizaje de una red neuronal de aprendizaje profundo sería el de una fábrica. Una vez que se ingresan las materias primas (conjunto de datos), se pasan por la cinta transportadora, con cada parada o capa posterior, se extrae un conjunto diferente de características de alto nivel. Si el objetivo de la red es reconocer un objeto, la primera capa podría analizar el brillo de los pixeles de la imagen.

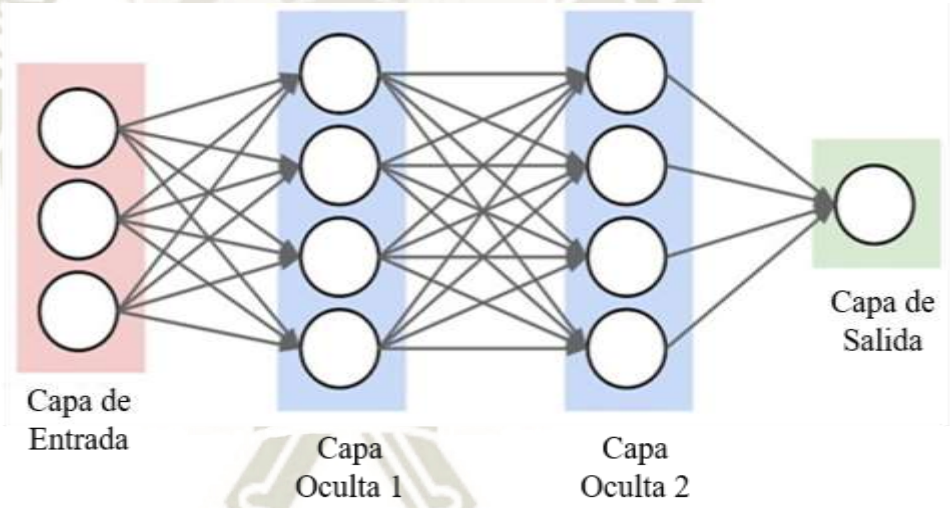


Figura 9. Estructura de una Red Neuronal Artificial según Dormehl (2019)

La siguiente capa podría identificar los bordes de la imagen, basado en los pixeles similares. Luego, otra capa pueda reconocer texturas y formas, y así sucesivamente. Al momento en que la cuarta o quinta capa es alcanzada, la red de aprendizaje profundo habrá creado detectores de características complejas. Esta puede descubrir que ciertos elementos de la imagen (como los ojos, nariz y boca) se encuentran comúnmente juntos.

Una vez terminado, los programadores que han entrenado la red pueden otorgar etiquetas a las salidas y luego utilizar la propagación hacia atrás para corregir cualquier error que haya podido ocurrir. A continuación, la red neuronal puede llevar a cabo sus propias tareas de clasificación sin necesidad de que los programadores intervengan en todo momento.

A. ¿Cuántas redes neuronales existen?

Según Dormehl (2019), hay varios tipos de redes neuronales, cada una para un caso de uso específico y nivel de complejidad. La red neuronal más básica es la red neuronal prealimentada, en donde la información se mueve en una única dirección: adelante; desde los nodos de entrada, a través de los nodos escondidos hasta los nodos de salida.

Otro tipo de red neuronal que es muy utilizada es la red neuronal recurrente, donde los datos se mueven en direcciones múltiples. Estas redes neuronales poseen grandes habilidades para aprender y son empleadas para tareas complicadas como la escritura a mano o reconocimiento de idioma.

También existen las redes neuronales convolucionales, redes de máquinas de Boltzmann, redes de Hopfield, entre otros. Elegir la red ideal para cada situación depende de los datos con lo que se tiene que entrenar y la aplicación que se quiere realizar. En varios casos, puede ser conveniente emplear diversos enfoques, como sería el caso de una tarea difícil como el reconocimiento de voz.

B. ¿Qué clases de tareas puede realizar una red neuronal?

Desde hacer que los autos puedan manejarse solos, hasta generar rostros muy realistas por computadoras, traducción automática, detección de fraude, leer mentes; las redes neuronales están detrás de muchos de los mayores avances de la inteligencia artificial.

Sin embargo, las redes neuronales han sido diseñadas para detectar patrones en los datos. Las tareas específicas incluyen clasificación (clasificación de conjuntos de datos en clases predefinidas), agrupación (clasificación de datos en diferentes categorías indefinidas) y predicción (uso de eventos pasados para adivinar los futuros como el mercado de valores o la taquilla de películas).

C. Pero, ¿cómo exactamente "aprenden" cosas?

De la misma manera que las personas aprenden de sus experiencias de vida, las redes neuronales necesitan datos para aprender. En la mayoría de los casos, mientras más datos procese la red neuronal, más precisa se volverá, mientras más entrene, gradualmente con el tiempo será más eficiente y cometerá menos errores.

Según Le (2018), cuando los investigadores o los científicos de datos entrenan una red neuronal, generalmente dividen los datos en tres conjuntos. Primero es un conjunto de entrenamiento que ayuda a la red a establecer diversos pesos entre los nodos. Luego, se refinan estos con un conjunto de datos de validación. Finalmente, se usará un conjunto de prueba para determinar si se puede convertir con éxito la entrada en la salida deseada.

D. ¿Las redes neuronales tienen limitaciones?

Uno de los grandes desafíos que afrontan los programadores es el tiempo que toma entrenar las redes neuronales, que requiere una cantidad considerable de poder computacional para tareas muy complicadas. Sin embargo, según Dormehl (2019), el mayor problema es que las redes neuronales son “cajas negras” en las que el usuario introduce datos y recibe respuestas, pero no tiene acceso al proceso de toma de decisiones.

E. Ventajas de las redes neuronales artificiales

Rouse (2019), indica que las ventajas de las redes neuronales artificiales son las siguientes:

Las capacidades de procesamiento en paralelo hacen que la red puede realizar más de un trabajo a la vez.

La información es almacenada en una red completa, no sólo en una base de datos.

La capacidad de aprender y modelar relaciones complejas no lineales ayuda a modelar las relaciones de la vida real entre entrada y salida.

La tolerancia a fallas ya que la corrupción de uno o más nodos de la red neuronal artificial no detendrá la generación de salida.

La corrupción gradual significa que la red se degradará lentamente con el tiempo, en lugar de que la red se destruya al instante.

La capacidad de producir resultados con conocimiento incompleto y la pérdida de rendimiento se basa en cuán importante es la información faltante.

No hay restricciones a las variables de entrada, así como tampoco a la forma en que son distribuídas.

El aprendizaje automático hace que la red neuronal artificial pueda aprender de los eventos y tomar decisiones basadas en las observaciones.

La capacidad de aprender relaciones ocultas en los datos sin ordenar ninguna relación fija ya que una red neuronal artificial puede modelar mejor los datos altamente volátiles y la varianza no constante.

La capacidad para generalizar e inferir relaciones no visibles en datos no vistos ya que las redes neuronales artificiales pueden predecir la salida de datos no vistos.

Figura 10. Ventajas de una Red Neuronal Artificial

F. Desventajas de las redes neuronales artificiales

Según Rouse (2019), las desventajas de las redes neuronales artificiales son las siguientes:

- Falta de reglas para determinar la estructura de la red ya que la arquitectura idónea de una red neuronal artificial sólo se puede determinar por pruebas, errores y experiencia.
- El requisito de procesadores con capacidades de procesamiento en paralelo hace que las redes neuronales artificiales dependan del hardware.
- La red funciona con información numérica, por lo tanto, todos los problemas deben traducirse en valores numéricos antes de que puedan ser ingresados a la red neuronal.
- La falta de explicación de las soluciones es una de las mayores desventajas de las redes neuronales artificiales. La incapacidad de explicar el por qué o cómo se genera la solución genera una cierta desconfianza en la red.

Figura 11. Desventajas de una Red Neuronal Artificial

G. Aplicaciones de las redes neuronales artificiales

Según Le (2018), las aplicaciones de las redes neuronales artificiales son las siguientes:

Figura 12. Aplicaciones de una Red Neuronal Artificial

1.2.4. Construcción de una Red Neuronal

Según Zhou (2019), para construir una red neuronal artificial se realiza lo siguiente:

A. Construcción de bloques: Neuronas

En primer lugar, se debe enfocarse en las neuronas, la unidad básica de una red neuronal. Una neurona toma entradas, realiza operaciones con ella y produce una salida.

Un ejemplo de una neurona con dos entradas es la siguiente:

Entradas Salida

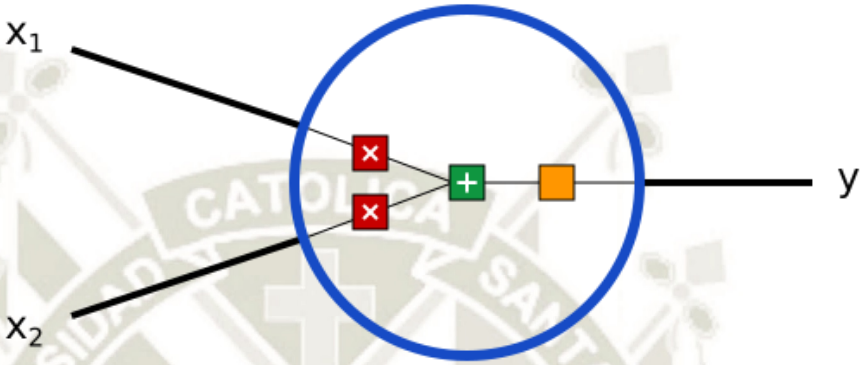


Figura 13. Construcción de una Red Neuronal Artificial según Zhou (2019)

En este ejemplo suceden tres cosas:

Primero, cada entrada es multiplicada por un peso, como se ve en la ecuación (1):

$$x_1 = x_1 * w_1 \quad (1)$$

$$x_2 = x_2 * w_2$$

A continuación, en la ecuación (2) todas las entradas con pesos son añadidas junto con un sesgo b:

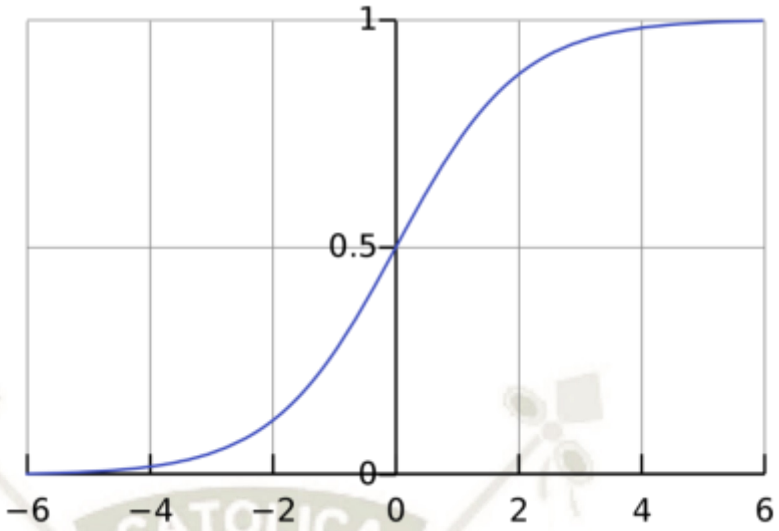


Figura 14. Función Sigmoidea

B. Combinando neuronas en una red neuronal

Una red neuronal no es más que un conjunto de neuronas conectadas. Un ejemplo de red neuronal simple es la siguiente:

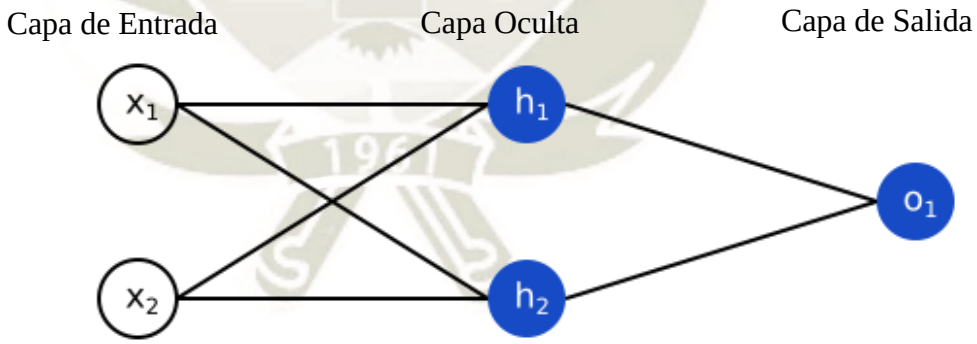


Figura 15. Capas de una Red Neuronal Artificial

Esta red neuronal tiene dos entradas, una capa oculta con dos neuronas (h_1 y h_2) y una capa oculta con una neurona (o_1). Las entradas para o_1 son las salidas de h_1 y h_2.

Una capa oculta es cualquier capa entre la capa de entrada (primera) y la capa de salida (última), connotando que pueden existir varias capas ocultas.

C. Entrenando una red neuronal

Para Zhou (2019), para entrenar una red neuronal se requiere:

Pérdida: antes de entrenar la red neuronal, se necesita una forma de cuantificar qué tan "buena" es la red para que pueda "mejorarse". Eso es la pérdida.

Para ello, el error cuadrático medio (MSE) es comúnmente utilizado para medir la pérdida, su fórmula se muestra en la ecuación (4):

$$\text{MSE} = \frac{1}{n} + \sum_{i=1}^{n} (y_{true} - y_{pred})^2 \quad (4)$$

n: es el número de muestras

y: representa la variable que predice

y_true: es el valor verdadero de la variable (la "respuesta correcta")

y_pred: es el valor predicho de la variable. Es lo que produce la red

(y_true−y_pred)²: error al cuadrado. La función de pérdida toma el promedio de todos los errores al cuadrado. Cuan mejor sean las predicciones, menor será la pérdida.

Mejores predicciones: menor pérdida

Entrenar una red neuronal: tratar de minimizar la pérdida

D. Entrenamiento: Descenso de gradiente estocástico

El descenso de gradiente estocástico (SGD) es un algoritmo de optimización que indica como cambiar los pesos y sesgos para minimizar la perdida. La ecuación (5) es la siguiente:

$$w_1 \leftarrow w_1 - n\frac{\partial L}{\partial w_1} \tag{5}$$

1.2.5. Función de Activación

Singh (2017), indica que las funciones de activación son muy importantes para que las redes neuronales artificiales puedan aprender y se pueda dar más sentido a cosas muy complicadas, además de introducir propiedades no lineales a la red neuronal artificial. El objetivo principal de una función de activación es convertir una señal de una neurona de entrada a una señal de salida.

1.2.6. Descenso de Gradiente

Hong (2016), indica que el descenso de gradiente es un algoritmo de optimización que funciona mediante la búsqueda eficiente de parámetros, la intersección y la pendiente para la regresión lineal.

1.2.7. Propagación hacia atrás

Hong (2016), indica que Backpropagation es un algoritmo utilizado para entrenar las redes neuronales artificiales, pudiendo actualizar los pesos eficientemente. Usualmente es usado con método de optimización de descenso de gradiente. Eremenko

(2018), indica que básicamente, Backpropagation ocurre cuando se retroalimentan los datos finales a través de la red neuronal y luego se ajustan las sinapsis ponderadas entre el valor de entrada y la neurona, al repetir este ciclo y ajustar los pesos, se reduce la función de costo.

1.2.8. Razonamiento basado en casos

Lozano y Fernández (2016), indican que el razonamiento basado en caso es un paradigma de resolución de problemas capaz de usar conocimiento de experiencias previas en concreto, además, el razonamiento basado en casos provee un acercamiento al aprendizaje incremental porque almacena una experiencia nueva cada vez que un problema es resuelto.

1.2.9. Algoritmos Genéticos

Mathworks (2018), indica que un algoritmo genético es un método para resolver problemas de optimización restringidos y no restringidos basados en la selección natural, el proceso que impulsa la evolución biológica. El algoritmo genético modifica repetidamente una población de soluciones individuales.

1.2.10. Bosque Aleatorio

Según Yiu (2019), el bosque aleatorio, como el nombre lo indica, consiste en un amplio número de árboles de decisiones individuales que operan como un conjunto. Cada árbol individual en el bosque aleatorio arroja una predicción de clase y la clase con más votos se convierte en la predicción del modelo como se muestra en la siguiente figura:

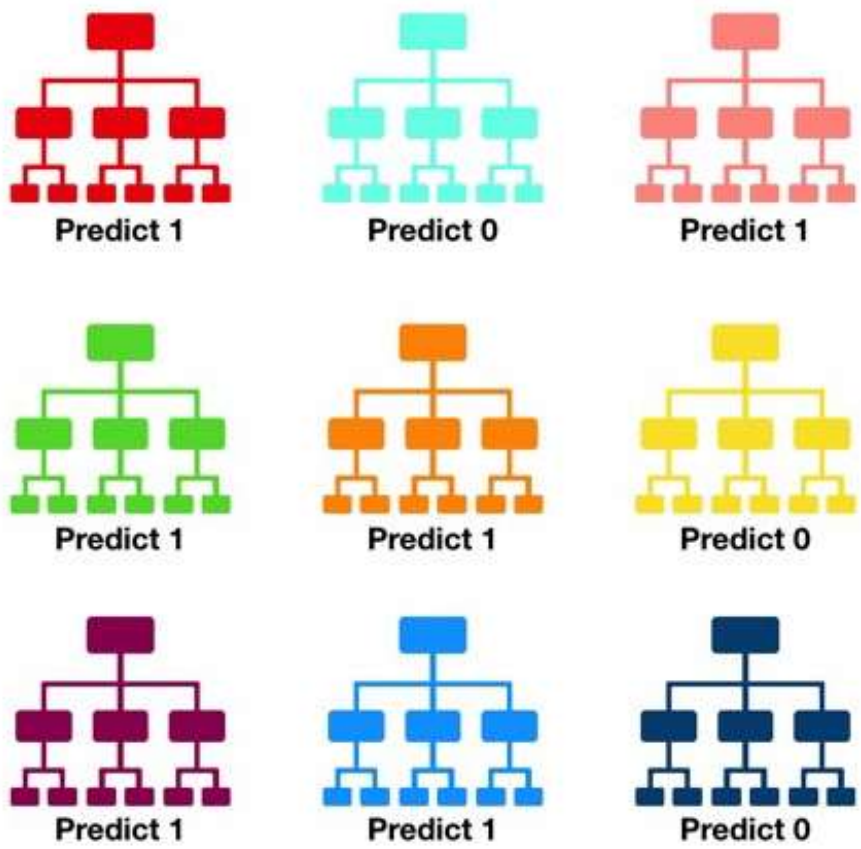


Figura 16. Ejemplo de Bosque Aleatorio según Yiu (2019)

A. Visualización de un modelo de bosque aleatorio mediante una predicción

Yiu (2019), indica que el concepto fundamental detrás del bosque aleatorio es sencillo pero poderoso: sabiduría de las multitudes. En la ciencia de los datos, la razón por la cual el modelo de bosque aleatorio funciona tan bien es que una gran cantidad de modelos (árboles) relativamente no correlacionados que operan juntos superará a cualquiera de los modelos individuales.

La baja correlación entre los modelos es la clave. Al igual que las inversiones con bajas correlaciones (como acciones y bonos) se unen para formar una cartera que es mayor que la suma de sus partes, los modelos no

correlacionados pueden producir predicciones que son más precisas que cualquiera de las predicciones individuales. La razón de este efecto es que los árboles se protegen entre sí de sus errores individuales (mientras no se equivoquen constantemente en la misma dirección), pero mientras varios árboles estén equivocados, muchos otros árboles estarán en lo correcto, por lo que, como grupo, los árboles pueden moverse en la dirección correcta.

Un requisito importante para que los bosques aleatorios funcionen bien es que las predicciones (errores) hechas por los árboles individuales deben tener bajas correlaciones entre sí.

1.2.11. Regresión Lineal

En términos sencillos, la regresión lineal es una manera de analizar la fortaleza de la relación entre una variable (variable de resultado) y uno o más variables (variables independientes).

Lorberfeld (2019), indica que una característica distintiva de la regresión lineal es que la relación entre las variables independientes y la variable de resultado es lineal. Esto quiere decir que, al momento de graficar las variables independientes con la variable de resultado, se observara que los puntos comienzan a tomar una forma de línea.

Según Saishruthi (2018), la regresión lineal tiene varios usos prácticos, la mayoría de las aplicaciones se dividen en las siguientes categorías:

Si el objetivo es la predicción, pronóstico o reducción de errores, se puede utilizar la regresión lineal para ajustar un modelo predictivo a un conjunto de datos de valores de respuestas y variables explicativas. Después de desarrollar dicho modelo, si se recopilan valores adicionales de las variables explicativas sin un valor de respuesta que lo acompañe, el modelo ajustado puede usarse para hacer una predicción de la respuesta.

Si el objetivo es explicar la variación de la variable de respuesta que puede ser atribuido a la variación en las variables explicativas, se puede aplicar un análisis de regresión lineal para cuantificar la fuerza de la relación entre la respuesta y las variables explicativas y para determinar si alguna variable explicativa pueda no tener una relación lineal con la respuesta, o identificar que subconjuntos de variables explicativas pueden contener información redundante sobre la respuesta.

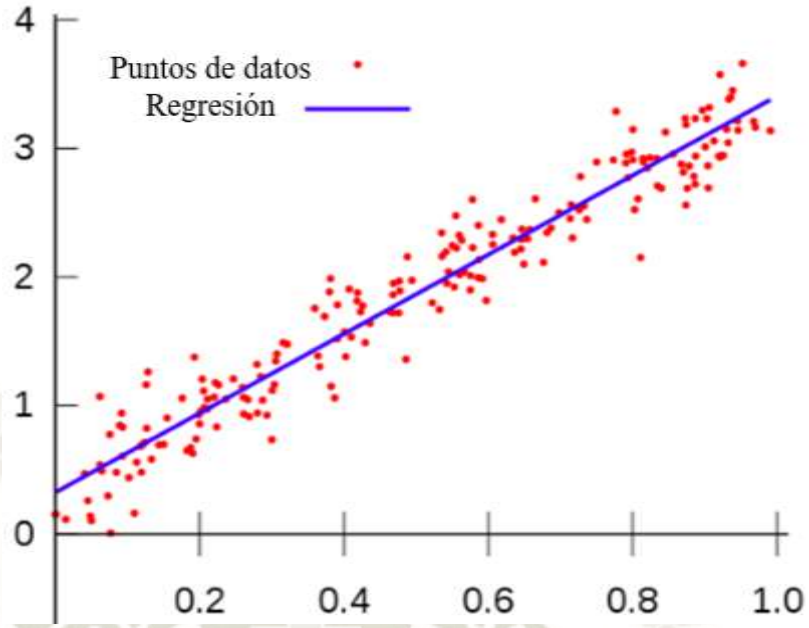


Figura 17. Gráfico de Regresión Lineal de Lorberfeld (2019)

1.2.12. Regresión Logística

En la sección anterior se puntualizó que la regresión lineal muestra los efectos de algunas variables tienen sobre otra variable, asumiendo que la variable de resultado es continua y que la relación entre la variable y la variable de resultado es lineal. Pero si la variable de salida es categórica, entonces entra en acción la regresión logística.

Chanakya (2018), indica que las variables categóricas son variables que solo pueden tiene una sola categoría.

Un ejemplo de variables categóricas podrían ser los días de la semana: si se tiene información sobre acontecimientos ocurridos en ciertos días de la semana, no hay posibilidad de que se presente información entre los lunes y jueves. Si algo ocurre el lunes, siempre será ese lunes.

Los modelos de regresión logística generan una probabilidad de que los datos estén en una categoría u otra, en lugar de un valor numérico regular, por esta razón los modelos de regresión logística son usados principalmente para la clasificación.

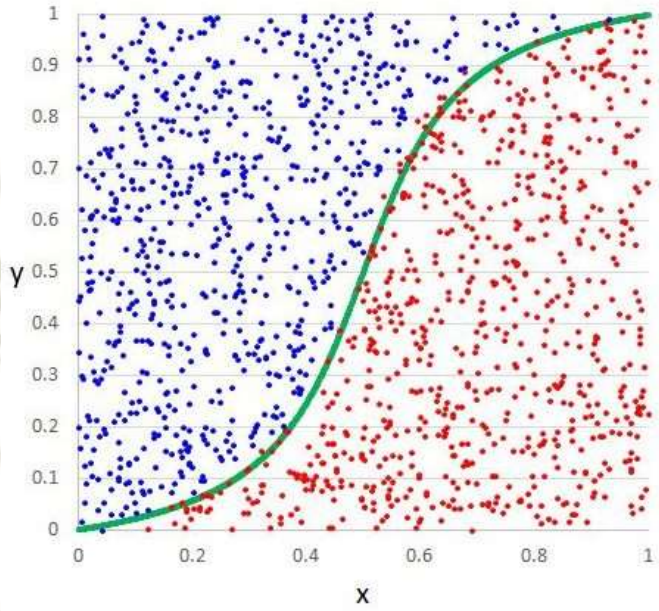


Figura 18. Uso de la Regresión Logística

1.2.13. Potenciación de Gradiente

Singh (2018), indica que la potenciación es un método para convertir el aprendizaje débil en aprendizaje fuerte. En la potenciación, cada nuevo árbol se ajusta a una versión modificada del conjunto de datos original. El algoritmo de potenciación de gradiente puede explicarse introduciendo el algoritmo AdaBoost. Este algoritmo empieza por entrenar un árbol de decisión y a cada observación se le asigna el mismo peso.

Después de evaluar el primer árbol, se incrementan los pesos de esas observaciones que son difíciles de clasificar y disminuir los pesos para las

observaciones que son fáciles de clasificar. El segundo árbol se genera con estos pesos en los datos con el objetivo de mejorar las predicciones del primer árbol, entonces el nuevo modelo sería la mezcla del primer y segundo árbol.

A continuación, se calcula el error de clasificación del nuevo modelo de dos árboles y se genera un tercer árbol para predecir residuos revisados. Se repite este proceso para un numero especifico de iteraciones. Los árboles posteriores ayudan a clasificar las observaciones que no son bien clasificadas por los árboles anteriores. Las predicciones del modelo final es la suma de los pesos de las predicciones realizadas por los árboles anteriores.

La potenciación de gradiente entrena muchos modelos de una manera gradual, aditiva y secuencial. La mayor diferencia entre los algoritmos AdaBoost y la potenciación de gradiente es como estos algoritmos identifican las deficiencias del aprendizaje débil (árboles de decisión). Mientras que el modelo AdaBoost identifica las deficiencias usando datos con alto peso, la potenciación de gradiente realiza lo mismo mediante el uso de gradientes en la función pérdida.

Según Singh (2018), la función de pérdida es la medida que indica cuan buenos son los coeficientes de los modelos para ajustar los datos subyacentes. Una compresión lógica de la función de pérdida dependerá de lo que se trata de optimizar. Por ejemplo, si se trata de predecir los precios de venta de viviendas mediante el uso de regresión, la función de pérdida se basará en el error entre los precios reales y predichos de la vivienda. Del mismo modo, si el objetivo es clasificar los impagos de crédito, la función

de pérdida se basará en medir cuan bueno es el modelo predictivo para clasificar los préstamos incobrables.

Una de las principales razones para utilizar la potenciación de gradiente es que permite optimizar una función de costo especificada por el usuario, en lugar de una función de pérdida que generalmente ofrece menos control y no corresponde esencialmente con las aplicaciones del mundo real.

1.2.14. Máquinas de Vector de Soporte

Según Yadav (2018), el objetivo del algoritmo de máquinas de vectores soporte es encontrar un hiperplano en un espacio n dimensional para clasificar los datos de manera clara.

Para separar las dos clases de datos hay muchos hiperplanos posibles que podrían elegirse, esto con el objetivo de encontrar un plano que tenga el margen máximo, es decir, la distancia máxima entre los datos de ambas clases. Maximizar el margen de distancia proporciona cierto refuerzo a los datos futuros para que puedan ser clasificados con mayor confianza.

Yadav (2018), indica que las ventajas y desventajas de las máquinas de vector de soporte son las siguientes:

Ventajas	Desventajas
• Es efectivo en dimensiones altas. • Es efectivo cuando la cantidad de características son más que los ejemplos de entrenamiento. • Es adecuado para la clasificación binaria en casos extremos.	• Para conjuntos de datos más grandes se requiere una gran cantidad de tiempo para su procesamiento. • No funciona bien en caso de clases superpuestas. • Seleccionar la función de kernel adecuada puede ser complicado.

Figura 19. Ventajas y Desventajas de Máquinas de Vector de Soporte

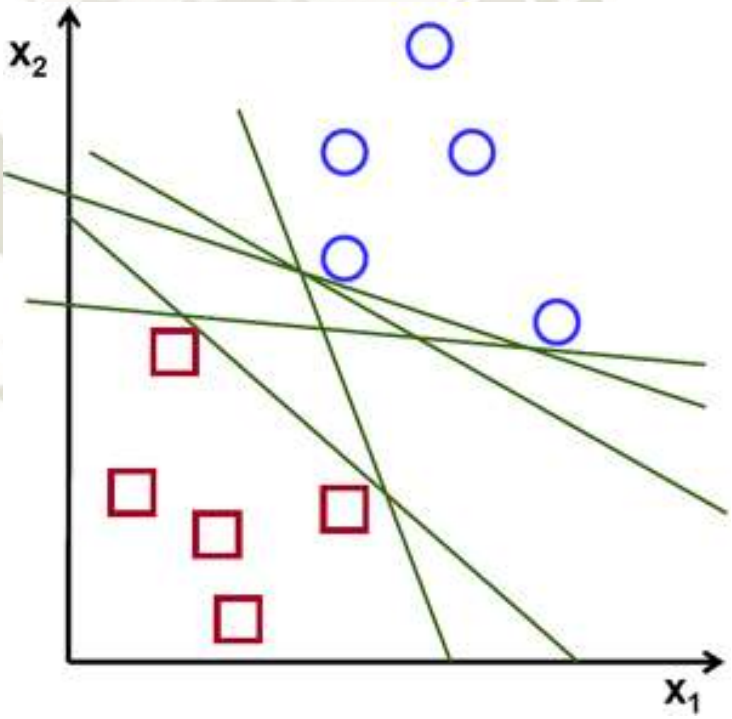


Figura 20. Dispersión con Maquinas de Vector de Soporte

1.2.15. Lenguaje de Programación Python

Según Python Contributors (2019), Python es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semántica dinámica. Sus estructuras de datos integradas de alto nivel combinadas con tipeo y enlace dinámico lo hacen atractivo para el desarrollo rápido de aplicaciones, así como para utilizarlo como lenguaje de scripting para conectar componentes existentes.

Las sintaxis simple y fácil de aprender de Python enfatiza la lectura y por lo cual, reduce el costo de mantenimiento de los programas.

Python admite módulos y paquetes, lo que fomenta la modularidad de los programas y la reutilización de código. El intérprete de Python y su extensa librería están disponibles en formato de código fuente o binario de forma gratuita para todas las plataformas y son distribuidas libremente.

A. Ventajas de Python

Según Mindfire Solutions (2017), las ventajas de Python son:

- Amplias librerías de soporte

  Proporciona grandes librerías que incluyen operaciones de cadenas, herramientas de servicio web, interfaces y protocolos del sistema operativo. La mayoría de las tareas de programación ya están escritas por lo que se reduce la longitud de los códigos que se escribirán en Python.

- Integración

  Python integra aplicaciones empresariales facilitando el desarrollo de servicios web al invocar componentes COM o COBRA. Además, tiene una poderosa capacidad de control ya que llama directamente a través de C y C++. Python también procesa XML y otros lenguajes de marcado, por lo que puede ejecutarse en todos los sistemas operativos modernos a través del mismo código de bytes.

- Mejora la productividad del programador

  El lenguaje tiene varias librerías de soporte y son orientados a objetos que aumentan de dos a 10 veces la productividad del programador al usar lenguajes como Java, VB, Perl, C, C++, C#.

- Productividad

  Con su sólida característica de integración de procesos, framework de pruebas unitarias y las capacidades de control se aumenta la velocidad para la mayoría de las aplicaciones y la productividad de las aplicaciones. Es una gran opción para la creación de aplicaciones de red multiprotocolo escalables.

B. Desventajas de Python

Mindifire Solutions (2017), indica que las desventajas de Python son las siguientes:

- Dificultad para usar en otros lenguajes

  Los usuarios de Python se acostumbran tanto a sus características y a sus extensas librerías que enfrentan problemas para aprender a trabajar en otros lenguajes de programación. Además, pueden ver la declaración de variables, requisitos sintácticos de agregar llaves o puntos y comas como una tarea molesta.

- Deficiencias en programación móvil

  Python tiene una gran presencia en muchas plataformas de escritorio y de servidor, pero es visto como un lenguaje deficiente para la programación móvil. Esta es la razón por la que se construyen muy pocas aplicaciones móviles como Carbonnelle.

- Se ralentiza al aplicar velocidad

  Python se ejecuta con la ayuda de un intérprete en lugar de un compilador, lo que hace que se ralentice, ya que la compilación y la ejecución ayuda a que funcione normalmente.

- Errores de tiempo de ejecución

  Python se escribe dinámicamente por lo que tiene muchas restricciones de diseño, incluso se ve que requiere más tiempo de pruebas, también es

menester indicar que los errores pueden aparecer cuando las aplicaciones ya se están ejecutando.

- Acceso a base de datos subdesarrolladas

  Comparada con tecnologías populares como JDBC y ODBC, el acceso a la base de datos de Python está poco desarrollada y es primitiva.

1.2.16. IDE

Según Codecademy (2018), un entorno de desarrollo integrado (IDE) es una aplicación de software que proporciona facilidades a los desarrolladores de software. Un IDE consiste normalmente de al menos un editor de código fuente, herramientas de automatización de compilación y un depurador. Algunos IDEs como Netbeans y Eclipse contienen un compilador, un intérprete o ambos; otros como SharpDevelop y Lazarus no lo tienen.

A. Pycharm

Según Pycharm (2019), PyCharm es un entorno de desarrollo integrado (IDE) utilizado en la programación de software, específicamente para el lenguaje de programación Python. Fue desarrollado por la compañía checa Jetbrains, provee análisis de código, un depurador gráfico, una unidad de pruebas integradas, integración de versión de controles y soporte desarrollo web con Django como también soporte para la ciencia de datos con Anaconda.

B. Anaconda

Anaconda (2018), indica que Anaconda es una distribución gratuita y de código abierto del lenguaje de programación Python para el desarrollo de software

científico (ciencia de datos, aplicaciones de aprendizaje automático, procesamiento de datos a gran escala, análisis predictivo, entre otros), con el objetivo de simplificar la administración y el despliegue de paquetes.

C. Navegador Anaconda

Anaconda Cloud (2016), indica que el navegador anaconda es una interfaz gráfica de usuario de escritorio incluida en Anaconda que permite la ejecución de aplicaciones y el fácil manejo de paquetes conda, entornos y canales sin la necesidad de utilizar la línea de comandos.

D. Jupyter

Según Jupyter (2014), Júpiter es un proyecto de código abierto sin fines de lucro nacido del proyecto IPython en 2014 a razón de admitir la ciencia de datos y la programación científica. Importante indicar que es un software 100% de código abierto, gratuito y lanzado bajo los términos liberales de la licencia BSD.

1.2.17. DataSynthetizer

DataSynthetizer es una herramienta que genera un conjunto de datos sintéticos a partir de datos reales, estos son estructurados y estadísticamente similares a los datos reales. Es una herramienta muy útil para el campo de la inteligencia artificial porque hay técnicas que necesitan una gran cantidad de datos para poder detectar patrones, diferentes comportamientos en los datos. Es muy improbable que cualquier persona pueda tener tal cantidad de información, por ello con esta herramienta se puede generar una cantidad masiva de datos a partir de una cantidad de datos pequeña siendo muy útil para investigaciones de aprendizaje automático. Está realizado en Python y fue creado por Ping, Stoyanovich y Howe (2017).

1.2.18. FENACREP

Según FENACREP (2017), la Federación Nacional de Cooperativas de Ahorro y Crédito del Perú (FENACREP), es una organización de integración que se fundó el 10 de abril de 1959 con el objetivo de realizar actividades de representación, educación, defensa, asistencia y supervisión de las cooperativas del Perú.

Desde el año 2019 la FENACREP está supervisada por la Superintendencia de Banca, Seguros y AFP (SBS). El movimiento cooperativo del Perú está conformado de alrededor de 150 instituciones que sirven a más de un millón y medio de socios a nivel nacional.

1.2.19. SBS

SBS (2018), indica que la Superintendencia de Banca, Seguros y AFP (SBS), es el organismo encargado de regular y supervisar los sistemas de seguros, financiero y privado de pensiones (PPS), así como la prevención y detección de lavado de activos y financiación del terrorismo.

La SBS tiene como propósito velar por los intereses de los asegurados, afiliados y depositantes al PPS.

La autonomía funcional de la SBS es reconocida en la Constitución Política del Perú y sus objetivos, atribuciones y funciones, se encuentran establecidos en la Ley General del Sistema Financiero y del Sistema de Seguros y Orgánica de la SBS.

A. SBS y las Cooperativas

Las cooperativas de ahorro y crédito son los instrumentos de cooperación social más antiguos del país y su importancia es vital para la inclusión financiera

y el desarrollo económico del Perú, en especial en las zonas más alejadas y vulnerables del país.

Según la SBS (2018), gracias a la Ley N° 30822, que entró en vigor el 1 de enero de 2019, la SBS está encargada de supervisar a las cooperativas, por lo cual, ha creado nuevas bases para este marco el regulatorio que respete los principios y la naturaleza del cooperativismo, además de implementar acciones para poder acompañar en este proceso a las cooperativas.

1.2.20. Cooperativa

Según FENACREP (2017), una cooperativa de ahorro y crédito es una institución financiera que es propiedad de sus socios, quiénes la guían y gozan de sus beneficios.

Es una entidad sin fines de lucro que tiene como fin poder servir a sus socios brindando un ambiente seguro y conveniente para depositar sus ahorros y poder acceder a créditos a tasas preferenciales, además de otros servicios financieros.

A. Principios Cooperativos

FENACREP (2017), indica que los principios corporativos son:

- Membresía abierta y voluntaria

  Las cooperativas son organizaciones públicas y son para todas aquellas personas que deseen utilizar sus servicios, además deben aceptar miembros sin ningún tipo de discriminación ya sea de género, raza, clase social, posición política y religión.

- Control democrático de los socios

  Las cooperativas son controladas por sus socios de forma democrática, éstos participan activamente en las políticas y en las tomas decisiones de la cooperativa. Las personas elegidas para representar a la cooperativa responden ante los socios.

- Participación económica de los socios

  Los socios contribuyen de forma igualitaria y controlan democráticamente el capital de la cooperativa. Usualmente los socios reciben una compensación (si es que la hay) sobre el capital como condición de membresía. Los excedentes son asignados por los socios para los siguientes fines: creación de reservas para el desarrollo la cooperativa, beneficios de los socios proporcionales a las transacciones de la cooperativa y a otras actividades aprobadas por los socios.

- Autonomía e independencia

  Las cooperativas son controladas por sus socios de manera autónoma. Si tienen algún acuerdo con otras organizaciones o tienen parte de su capital en otras fuentes, éstas son realizadas en términos que aseguren el control de sus socios democráticamente y manteniendo la autonomía de la cooperativa

- Educación, formación e información

  Las cooperativas brindan entretenimiento y educación a sus socios, a sus gerentes y empleados, con el objetivo de desarrollarse más. Además,

informan al público en general acerca de los beneficios de ser socio de la cooperativa.

- Cooperación entre cooperativas

  Al trabajar en equipo por medio de instalación a nivel local, regional y nacional, las cooperativas sirven a sus socios de manera más eficaz y a la vez fortalecen el movimiento cooperativo.

- Compromiso con la comunidad

  Con las políticas aprobadas por los socios, la cooperativa trabaja para el desarrollo sostenible de la comunidad.

1.2.21. COOSUNAT

Según Córdova (2016), la Cooperativa de Ahorro y Crédito de Trabajadores de la SUNAT (COOSUNAT), fue fundada en febrero del año 2015 por un grupo de trabajadores de la SUNAT, con el objetivo de la cooperación e integración entre trabajadores.

Para inscribirse como socio en la cooperativa, el requisito primordial es ser trabajador de la SUNAT, sin importar cual es el régimen laboral o tipo de contrato de trabajo. Además, si se es cesado o desvinculado de la SUNAT, el socio puede seguir aportando a la cooperativa y así gozar de los mismos derechos y beneficios de un socio que aún labore en la SUNAT.

COOSUNAT tiene el reconocimiento e inscripción como cooperativa con Partida Registral Nº 11301485 por parte de la Superintendencia Nacional de Registros Públicos, asimismo, cuenta con el RUC 20600279484.

Es supervisada por la Superintendencia Banca, Seguros y AFP (SBS) y está inscrita en la Federación de Cooperativas de Ahorro y Crédito del Perú (FENACREP), recalcando de que solo se acepta como socio a trabajadores de la SUNAT y no al público general ya que esta cooperativa es de tipo cerrada.

La Superintendencia de Banca, Seguros y AFP (SBS), reconoció a la cooperativa con el nivel dos, ya que al finalizar el año 2018, se contó con activos superiores a las 600 UIT (S/ 2520000). Con esta distinción, la cooperativa puede realizar operaciones financieras conforme a lo indicado en la Ley N° 30822.

COOSUNAT tiene como objetivo social la promoción del ahorro entre sus socios y la utilización de los fondos comunales en la inversión de actividades empresariales a nivel nacional, donde cada socio será parte de estos éxitos.

1.2.22. SUNAT

Según Guerra (2018), la Superintendencia Nacional de Aduanas y de Administración Tributaria (SUNAT), es un organismo recaudador de los tributos internos del país permitiendo la financiación de los gastos públicos del Estado, así como también la regularización de pago de los sueldos de los empleados públicos.

Desde el año 2002 la Superintendencia Nacional de Aduanas y la Superintendencia Nacional de Administración Tributaria se fusionaron y así estas dos entidades independientes trabajaron como una sola para mejorar sus labores y aumentar la productividad del Estado.

La función principal de la SUNAT es administrar los tributos internos del país, además de implementar programas para nutrir la cultura de los contribuyentes del estado, proponer reglas de normas tributarias, entre otros.

1.2.23. Relación Laboral 728

Toda persona que trabaje de manera subordinada, que reciba órdenes, tenga un jefe y ocupe un puesto tiene una relación laboral.

Según Pizarro (2018), las relaciones laborales del sector privado se clasifican de acuerdo con el Texto Único Ordenado del Decreto Legislativo 728, el cual regula la siguiente modalidad de relación laboral:

Las personas cuyo contrato es de plazo indeterminado, la relación no tiene un contrato escrito de manera obligatoria basta que el trabajador se pueda registrar en la planilla de la empresa ya que es un contrato laboral. No se necesita saber el tiempo en el cual el contrato va a terminar, pero se ha diseñado para que pueda durar un largo tiempo.

Para dar por concluida la relación laboral con la empresa, la persona debe renunciar o por el mutuo acuerdo de ambas partes, también como el fallecimiento del trabajador o el despido.

1.2.24. CAS

Según Conduce tu Empresa (2019), el Contrato Administrativo de Servicios (CAS), es una modalidad especial del Estado que se da entre una persona y éste, para realizar un servicio subordinado y dependiente dentro de una institución proporcionando recursos, bienes, equipos, insumos para así realizar las tareas para el cual ha sido contratado.

1.2.25. Renuncia de clientes

Galleto (2016), indica que la renuncia de clientes sucede cuando los clientes dejan de hacer negocios con una compañía o un servicio. La renuncia de clientes es una medida crítica porque es menos costoso retener a los clientes actuales que adquirir nuevos clientes ya que para ganar nuevos clientes se tiene que utilizar bastante

comercialización, mucha publicidad, recursos de ventas para realizar todo este proceso. En cambio, retener a los clientes generalmente es más rentable porque ya se ha ganado la confianza y la lealtad de los clientes actuales.

La renuncia de clientes impide el crecimiento de las empresas, así que tienen que definir un método para calcularla en un periodo de tiempo determinado. Al conocer y poder monitorear la tasa de renuncia de clientes, las empresas estarán equipadas para determinar las tasas de éxito de retención de clientes e implementar mejores estrategias con sus clientes.

Varias organizaciones calculan la tasa de renuncia de clientes de diferentes maneras ya que la tasa de renuncia puede representar el número total de clientes perdidos, porcentaje de clientes perdidos comparado con el total de clientes o el valor del negocio perdido. Además, otras organizaciones calculan la tasa de renuncia durante un periodo de tiempo como periodos trimestrales o años fiscales. Uno de los métodos más utilizados para calcular la renuncia de clientes es dividir el número total de clientes que tiene una empresa al comienzo de un periodo de tiempo entre el número de clientes perdidos durante el mismo periodo.

A. Causas de la renuncia de clientes

Según Galleto (2016), existen varias causas que llevan a los clientes dejar la empresa, las cuales son las siguientes:

Mal servicio al cliente. Actualmente se vive y se trabaja en la era del cliente, ellos exigen un servicio y experiencia excepcional, cuando no lo reciben acuden a los competidores, incluso, comparten las experiencias negativas en redes

sociales. El mal servicio al cliente genera mucha más renuncia de clientes que solo un cliente que tuvo una mala experiencia en el servicio.

Otras causas de la renuncia de clientes son: el proceso de corporación deficiente de la empresa, falta de éxito por parte del cliente de manera continua, causas naturales que suceden en las empresas de vez en cuando, comunicaciones de baja calidad y falta de lealtad a la empresa.

B. Desventajas de la renuncia de clientes

Las desventajas de la renuncia de clientes según Galleto (2016), son:

Existe una relación directa entre el valor de por vida del cliente y la capacidad de hacer crecer la empresa, como tal, cuanto mayor sea la tasa de renuncia de clientes, menores serán las probabilidades de hacer crecer la empresa. Incluso si se tiene la mejor campaña de marketing en el mercado, el resultado final se verá afectado si se pierden clientes a un ritmo elevado, ya que el costo de adquirir nuevos clientes es muy alto.

Existe información variada sobre el tema del costo de retener a los clientes contra la adquisición de nuevos clientes, estos estudios muestran que los costos de adquisición de clientes superan con creces los costos de retenerlos. En general, las empresas gastan 7 veces más en adquirir clientes que retenerlos y el valor global promedio de un cliente perdido es de 243 dólares.

1.3. Desarrollo de la Metodología

Se desarrolló la metodología expuesta en el Plan de Tesis (ver Apéndice A), paso a paso, cada uno con sus respectivos resultados esperados, la descripción de cada uno se detalla a continuación:

Tabla 1

Metodología empleada para la investigación

Paso	**Descripción**	**Resultados Esperados**
Recolección de datos para el caso de estudio	Reunir los datos de los socios que se encuentran distribuidos de manera desordenada en la base de datos de la cooperativa.	Datos relevantes de todos los socios establecidos en hojas de cálculo.
Preprocesamiento de datos	Limpiar, integrar, transformar y reducir los datos para mejorar la calidad de estos.	Datos de los socios consolidados en un archivo de tipo csv (valores separados por coma).
Elección de técnicas supervisadas	Elegir técnicas supervisadas de aprendizaje automático para el análisis.	Cuadro de un conjunto de técnicas supervisadas de aprendizaje automático a partir de la realización de un análisis exhaustivo de rendimiento y precisión de cada técnica.

Aplicación de técnicas supervisadas	Aplicar las técnicas de aprendizaje automático a la cooperativa.	Listado de resultados de la precisión de las técnicas.
Verificación de resultados	Verificar los resultados obtenidos a partir de datos reales con un conjunto de datos sintéticos para hallar la máxima precisión.	Cuadro de resultados en donde se tiene la comparación de las precisiones de los datos reales con el conjunto de datos sintéticos.
Validación de las técnicas supervisadas	Validar las técnicas utilizadas para comprobar su efectividad.	Resultados de la predicción de las técnicas con dos casos reales de socios, uno renuncio y el otro se mantuvo. Prototipo de una aplicación web en donde se puedan ingresar datos de los socios y mediante el uso de las técnicas, muestre si el socio renunciará o no.

1.3.1. Recolección de datos para el caso de estudio

El conjunto de datos con los que se va a trabajar y analizar emana de la información real de la cooperativa COOSUNAT ubicada en el distrito de Cerro Colorado en la ciudad de Arequipa. Esta información está conformada por datos de los socios y por una clasificación propia de la cooperativa según él régimen laboral.

Estos datos de los socios son obtenidos al momento de que un trabajador de la SUNAT realiza el proceso de inscripción de socio que se detalla a continuación:

Se rellena el formulario de inscripción de socio (ver Apéndice B) desde cualquiera sucursal de la SUNAT y se envía por Olva Courier a la dirección de la cooperativa en Arequipa.

El inconveniente que tiene este proceso manual es que no existe una validación de los datos del formulario, por lo que, si hay campos vacíos, estos son aceptados de la misma manera que si hubieran llenado dicho campo.

Del conjunto de datos de los socios, se tienen los siguientes atributos con mayor relevancia:

Tabla 2.

Atributos más relevantes de los socios con su descripción

Variables	**Descripción**
Código del socio	Es el código interno de la cooperativa con el cuál se distingue a cada socio
Apellidos y nombres	Los apellidos y nombres del socio
Sucursal	La sucursal de la SUNAT en donde se encuentra laborando el socio
Contrato	El tipo de contrato del socio si es CAS o 728
Sexo	El sexo del socio, masculino o femenino

Estado civil	El estado civil que tiene el socio
Edad	La edad actual del socio
Tiempo en la cooperativa	El tiempo en meses en que el socio está en la cooperativa
Importe total	Hasta el mes de junio cuánto de aportes tiene el socio
Préstamo con la cooperativa	Hasta el mes de junio de 2019 cuánto de aportes tiene el socio
Deudas con la cooperativa	Si el socio ha tenido o no un préstamo con la cooperativa
Salario estimado	El salario estimado del socio
Renuncia	Si el socio ha renunciado o no a la cooperativa

La cooperativa tiene almacenados estos datos de los socios en hojas de cálculo de Excel porque esta herramienta facilita el manejo de estos datos para su personal, pese a ello, hay datos faltantes, erróneos, fuera de lugar, por ejemplo: existen varios atributos vacíos como estado civil, la edad, el sexo.

Un atributo muy importante para el análisis es el salario estimado, el cual, la cooperativa no puede tener acceso a esta información, sólo se puede saber el salario de un socio si éste solicitó un préstamo. Para solicitar un préstamo, el socio tiene que realizar un trámite con adjuntando varios documentos, uno de ellos son las dos últimas boletas de con este el analista de crédito puede aprobar la solicitud o no. Por ello es de

vital importancia la concretar la reducción de datos faltantes y poder normalizar la información para realizar el posterior análisis.

Del conjunto de datos reunido, se observó la siguiente distribución en los diferentes atributos a través de la herramienta Jupyter:

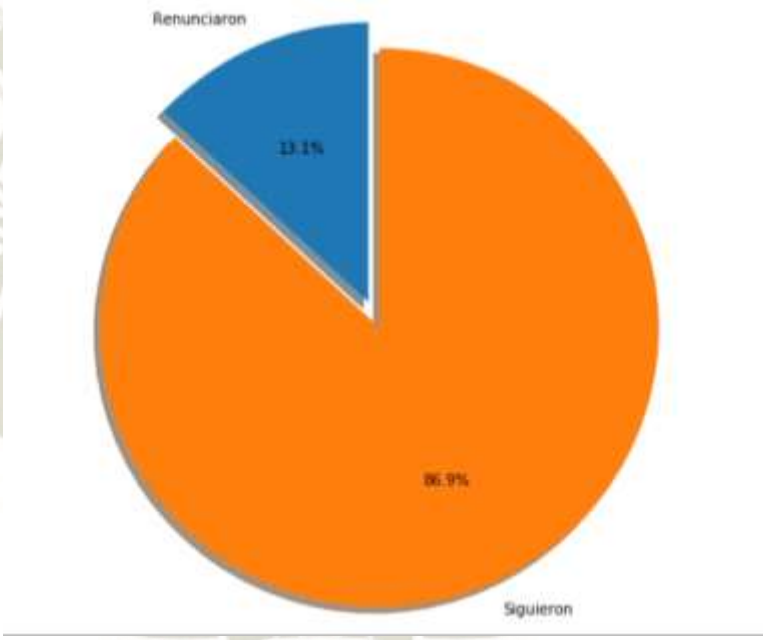


Figura 21. Proporción de socios que renunciaron y siguieron en la cooperativa

Se observa mediante el gráfico circular que el 13.1% de socios han renunciado a la cooperativa desde su fundación.

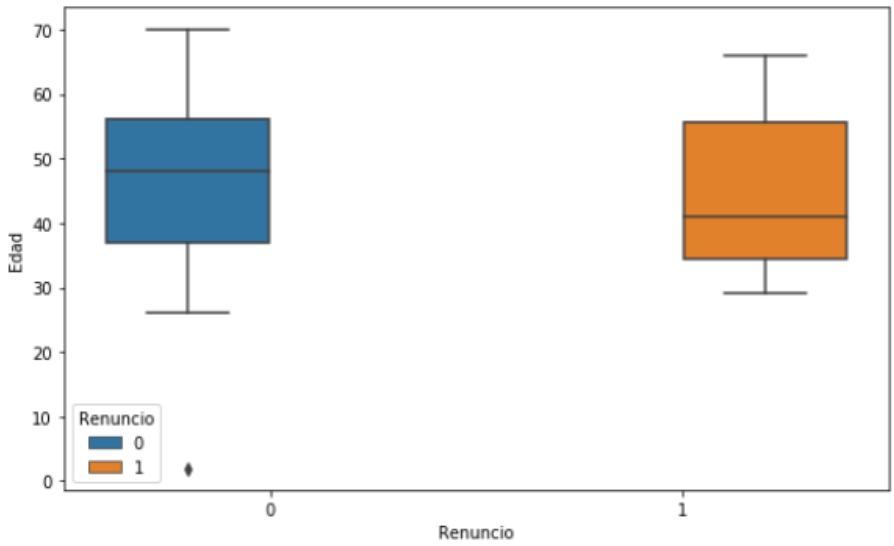


Figura 22. Renuncia de socios por edad

Mediante el diagrama de caja y bigote se observa que la mitad de los socios que han renunciado son mayores de 40 años, además indica que no hay socios menores de 30 años que hayan renunciado a la cooperativa. El 75% de los socios que han renunciado a la cooperativa son mayores de 35 años.

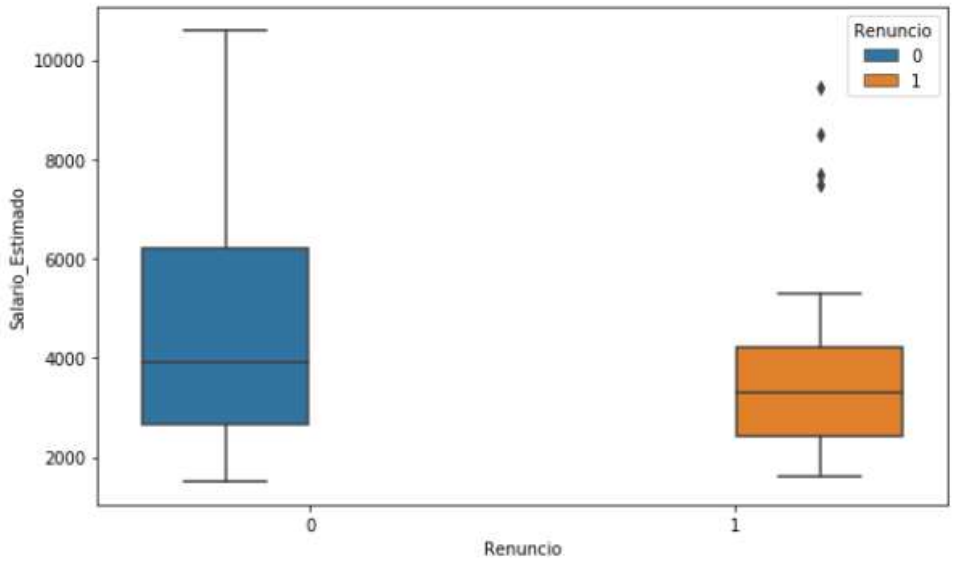


Figura 23. Renuncia de socios por salario estimado

Con el gráfico de caja y bigote se observa que los socios que se mantienen en la cooperativa el 75% tienen un salario estimado de entre 1500 a 6000 soles y los socios que renuncian tienen un salario estimado de entre 1500 a 5800 soles.

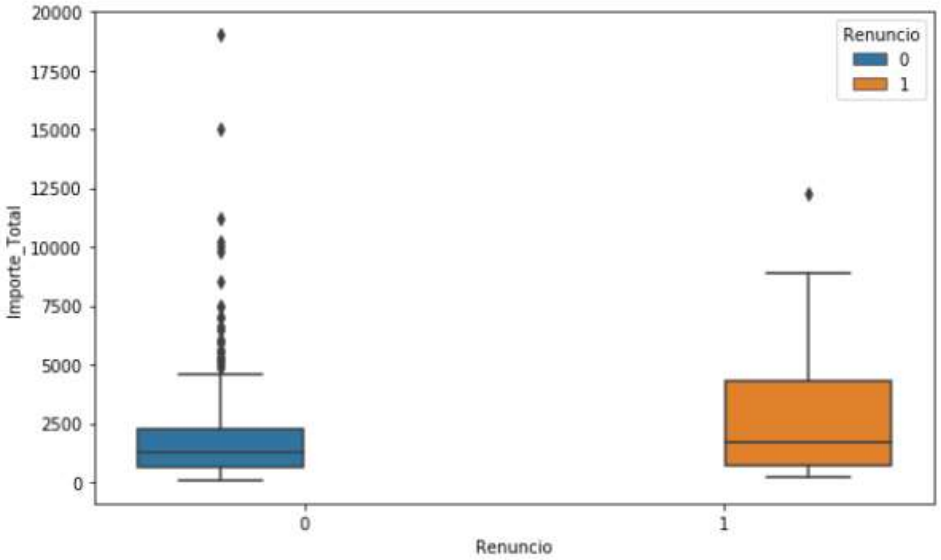


Figura 24. Renuncia de socios por total de importes

Con el grafico de caja y bigote se observa que más de la mitad de los socios que han aportado un monto superior a los 2500 soles e inferior a los 5000 se han retirado.

Los socios que se han retirado tienen un máximo de aportes de aproximadamente 8000 soles salvo un socio que llegó a aportar 12500 soles. Es importante indicar que los socios que se han mantenido en la cooperativa tienen aportes muy dispersos, es por eso que se observa la gran cantidad de valores atípicos.

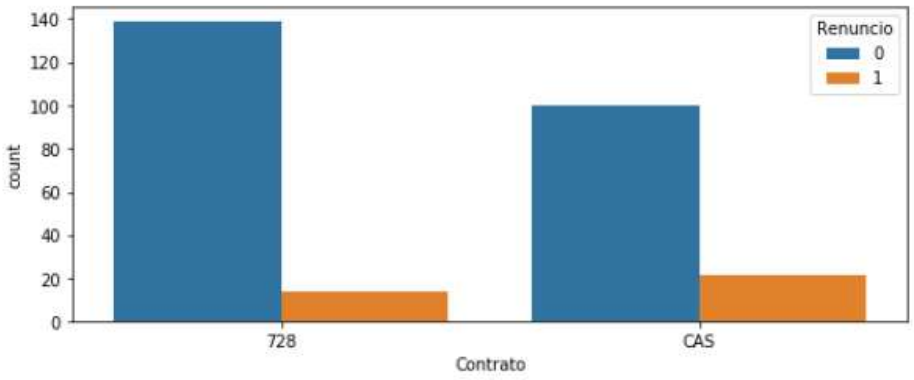


Figura 25. Proporción de socios que renunciaron por tipo de contrato

Del gráfico de barras se observa que los socios CAS son los que han renunciado en mayor proporción que los socios con contrato 728 a pesar de que hay más cantidad de socios 728 en la cooperativa.

1.3.2. Preprocesamiento

Para realizar el preprocesamiento se identificaron los atributos qué tienen que ser categorizados que son sucursal, sexo, contrato y estado civil.

En sucursal hay en total diecisiete diferentes valores, por lo cual se redujo este número a tres, clasificándolos en nuevos atributos que son Norte, Centro y Sur. El norte está constituido por los departamentos de La Libertad, Piura, Tumbes, Lambayeque, Loreto. El centro por Lima, Áncash, Junín, Pasco, Ucayali y el sur por Ica, Arequipa, Apurímac, Cusco, Puno, Moquegua y Tacna, esto se realizó con la fórmula BUSCARV de la hoja de cálculo Excel.

Para el atributo sexo, se colocó 1 para el sexo masculino y 0 para el sexo femenino, para el atributo contrato, se colocó 1 para el contrato 728 y 0 para CAS y para el atributo estado civil, se colocó 1 para soltero y 0 para casado, todo ello a través de la opción de buscar y reemplazar de la hoja de cálculo Excel.

A. Relleno de datos faltantes

Como se indicó en el punto anterior, no existe una validación de datos al momento de realizar la inscripción de un socio, por lo que se siguiere que se realice la validación para los atributos para así evitar los campos faltantes e inconsistentes como se detalla en la siguiente tabla:

Tabla 3.

Validación de los atributos proponiendo el tipo y longitud para cada uno

Variables	**Tipo**	**Longitud**	**Descripción**
Código del socio	Caracter	5	Es el código interno de la cooperativa con el cuál se distingue a cada socio
Apellidos y nombres	Caracter	50	Los apellidos y nombres del socio
Sucursal	Caracter	10	La sucursal de la SUNAT en donde se encuentra laborando el socio
Contrato	Caracter	3	El tipo de contrato del socio si es CAS o 728

Sexo	Caracter	1	El sexo del socio, masculino o femenino
Estado civil	Caracter	1	El estado civil que tiene el socio
Edad	Numérico	3	La edad actual del socio
Tiempo en la cooperativa	Numérico	3	El tiempo en meses en que el socio está en la cooperativa
Importe total	Flotante	7	Hasta el mes de junio cuánto de aportes tiene el socio
Préstamo con la cooperativa	Booleano	1	Hasta el mes de junio de 2019 cuánto de aportes tiene el socio
Deudas con la cooperativa	Booleano	1	Si el socio ha tenido o no un préstamo con la cooperativa
Salario estimado	Flotante	7	El salario estimado del socio

Renuncia	Booleano	1	Si el socio ha renunciado o no a la cooperativa

B. Filtro de datos

Para saber cuál es el salario estimado de un socio es indispensable que el socio haya tenido un préstamo con la cooperativa porque si no, no hay otra manera de saber eso porque esa información es confidencial. Por lo tanto, sólo se obtuvieron los salarios del 49.28% del total de socios de la cooperativa que son concretamente 275 socios, por lo que los socios que no tienen el salario estimado son filtrados.

C. Escalamiento de datos

Como se puntualizó en la sección anterior del marco metodológico, una red neuronal necesita que los datos estén escalados, por lo cual, se realizó un escalamiento para los atributos edad, tiempo en la cooperativa, importe total y salario estimado. Este proceso se realizó a través del módulo StandardScaler de la librería Sklearn.preprocessing de Python. El módulo funciona de la siguiente manera:

Estandariza los valores eliminando la media y escalando a la varianza de la unidad.

La escala de una muestra x se calcula con la siguiente ecuación (6):

$$z = (x - u) / s \quad (6)$$

Donde u es la media de las muestras de entrenamiento y s es la desviación estándar de las muestras de entrenamiento.

```
# Feature Scaling - Se escalan los datos para que
# tengan una proporción entre 0 y 1
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Figura 26. Uso de la librería Sklearn para escalar datos

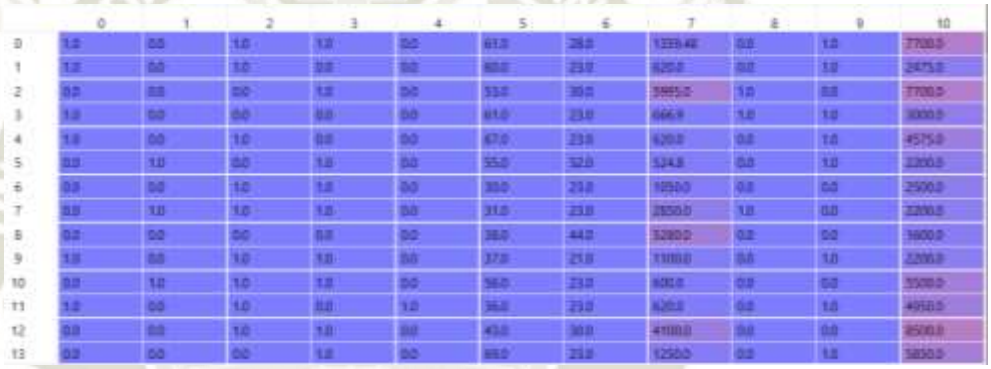


Figura 27. Estado de datos sin escalar

D.

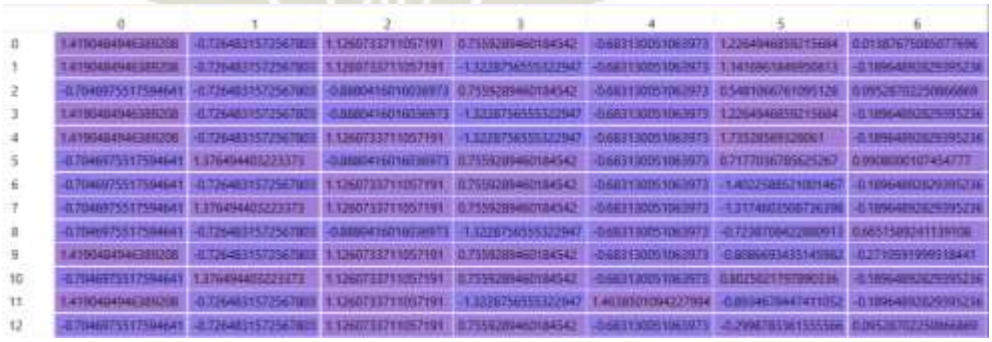


Figura 28. Datos escalados

D. Consolidación de los datos

Los datos de los socios de la cooperativa se encuentran en cuatro diferentes hojas de cálculo de Excel.

- Una hoja de cálculo llamada Padrón de Socios, en donde se encuentran el código de socio, los nombres y apellidos de cada socio, sexo, estado civil, tipo de contrato, edad y sucursal.
- Una hoja de cálculo llamado Aportes, en donde se encuentran los datos de aportes totales de todos los socios, su tiempo en la cooperativa y si renunciaron o no.
- Una hoja de cálculo llamado Créditos, en donde se encuentran todos los préstamos de los socios que han realizado con la cooperativa, así como los socios deudores.
- Para el salario estimado se creó una hoja Excel con los nombres de los socios que han tenido algún crédito con la cooperativa y su remuneración básica obtenida de su boleta de pago.

Para la consolidación de datos se integró todos estos datos de las 4 hojas de cálculo en un solo archivo de tipo CSV, todo esto se realizó con la herramienta Excel.

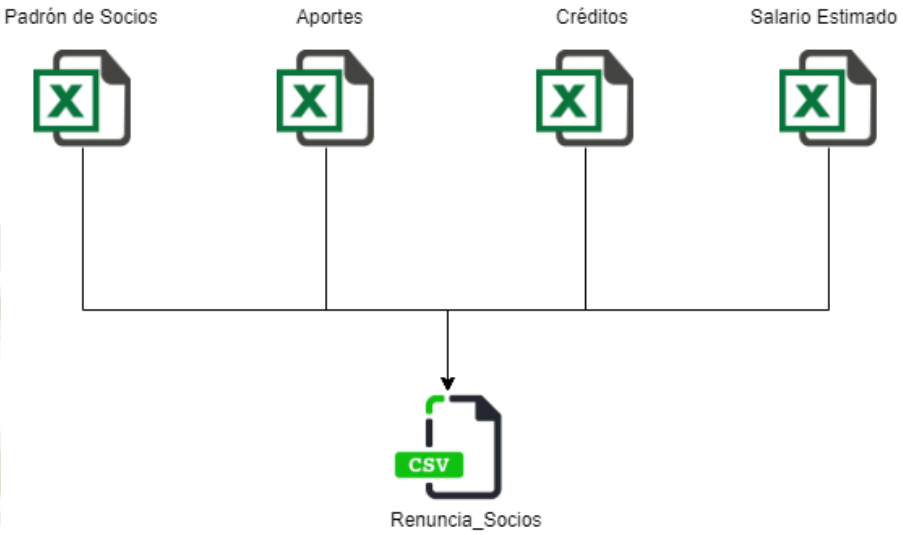


Figura 29. Proceso de consolidación de datos

Tabla 4.

Primeros 10 datos de los socios en el archivo CSV

Numero_Fila	Id_Socio	Apellidos_y_Nombres	Sucursal	Contrato	Sexo	Estado_Civil	Edad	Tiempo_en_Cooperativa	Importe_Total	Tiene_Prestamo	Tiene_Deudas	Salario_Estimado	Renuncio
1	1	Socio0	SUR	728	M	C	42	54	19052.36	1	1	7700	0
2	2	Socio1	NORTE	728	F	S	42	52	4400	0	0	2650	0
3	4	Socio2	SUR	728	M	C	55	52	5200	1	1	9450	0
4	6	Socio3	CENTRO	728	M	C	53	52	5600	1	1	7700	0
5	7	Socio4	SUR	728	M	C	53	52	5295	0	0	8975	0
6	10	Socio5	SUR	728	M	C	55	52	524.8	0	1	2200	0
7	13	Socio6	SUR	728	M	C	55	52	7495	0	0	7700	0
8	16	Socio7	SUR	728	M	C	70	52	6500	0	1	7700	0
9	17	Socio8	SUR	728	M	C	48	52	5300	0	1	2200	0
10	14	Socio9	NORTE	728	F	C	54	52	7050	1	1	7700	0

1.3.3. Elección de técnicas supervisadas

A continuación, se presentan las técnicas elegidas de acuerdo a las características que cada uno posee y se adecúan de la mejor manera para realizar el trabajo de investigación.

A. Redes Neuronales Artificiales

Esta técnica fue seleccionada porque tiene la habilidad de aprender de sí misma y producir una salida. Además, si una neurona no responde o si la información se ha perdido la red neuronal puede seguir funcionando y producir la salida. Y por último, a diferencia de muchas otras técnicas de predicción, las redes neuronales artificiales no imponen restricción alguna a las variables de entrada, por ejemplo cómo deben distribuirse.

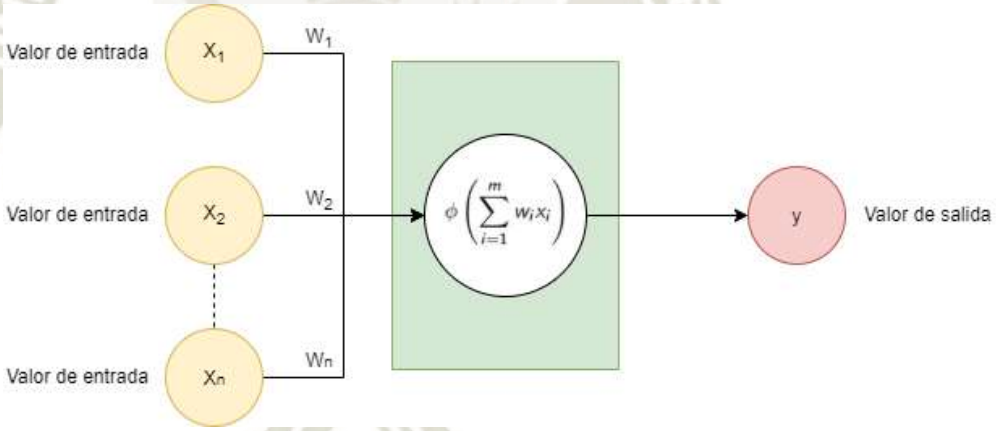


Figura 30. Funcionamiento de una Red Neuronal Artificial

B. Regresión Logística

La técnica de regresión logística ha sido seleccionada porque es muy eficiente, no requiere muchos recursos computacionales, se puede interpretar con facilidad, no se necesita que los valores de entrada sean escalados, realiza buenas predicciones de probabilidad y su implementación no es muy complicada.

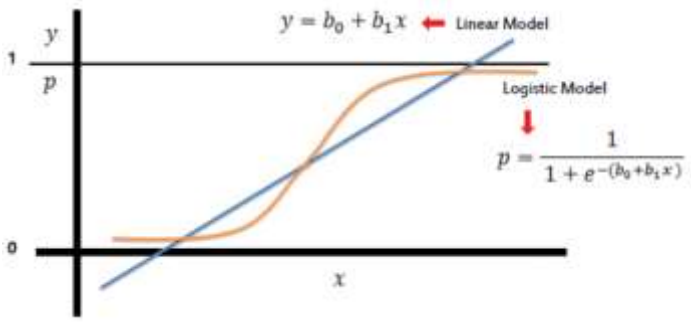


Figura 31. Fórmula de la Regresión Logística según Chanakya (2018)

En la regresión logística, la constante (b_0) mueve la curva hacia la izquierda y hacia la derecha y la pendiente (b_1) define la inclinación de la curva.

C. Máquinas de Vector de Soporte

La técnica de máquinas de vectores soporte ha sido seleccionada porque no necesita grandes conjuntos de datos para trabajar, tiene una un rendimiento muy bueno cuando no hay mucho ruido en la información, no necesitan mucha memoria y tiene varias variantes para elegir y que funcionen bien con el conjunto de datos.

Para la predicción de una nueva entrada utilizando el producto de punto entre la entrada (x) y cada vector de soporte (x_i) se calcula de la siguiente manera con la ecuación (7):

$$f(x) = B_0 + \text{Suma}(a_i * (x, x_i)) \quad (7)$$

Esta es una ecuación que implica calcular los productos internos de un nuevo vector de entrada (x) con todos los vectores de soporte en los datos de entrenamiento. El coeficiente de aprendizaje debe estimar los coeficientes B_0 y a_i (para cada entrada) a partir de los datos de entrenamiento.

D. Bosque Aleatorio

La técnica del bosque aleatorio ha sido seleccionada porque es muy flexible y bastante preciso, naturalmente asigna puntaje de importancia a los atributos por lo que puede manejar mejor los atributos redundantes, es robusto para el sobreajuste y los datos no necesitan ser escalados.

La fórmula de la implementación del bosque aleatorio se representa con la ecuación (8):

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \quad (8)$$

ni_j = la importancia del nodo j

w_j = número ponderado de muestras que alcanzan el nodo j

C_j = el valor de impureza del nodo j

$left_j$ = nodo hijo de la división izquierda en el nodo j

$right_j$ = nodo hijo de la división derecha en el nodo j

E. Potenciación de Gradiente

La técnica de potenciación de gradiente ha sido seleccionada porque es robusto a los datos faltantes, a la carga los datos faltantes e irrelevantes; esta técnica asigna naturalmente puntaje importancia en los atributos, tiene un excelente rendimiento y los datos no necesitan ser escalados.

Para realizar predicciones, la función de pérdida (MSE) debe ser mínima. Al usar el descenso de gradiente y actualizar las predicciones basadas en una tasa de aprendizaje, los valores de MSE es mínimo, obteniendo las siguientes ecuaciones (9) y (10).

$$y_i^p = y_i^p + \alpha * \delta \sum (y_i - y_i^p)^2 / \delta y_i^p \quad (9)$$

$$y_i^p = y_i^p - \alpha * 2 * \Sigma (y_i - y_i^p) \quad (10)$$

Donde, α es la tasa de aprendizaje y es la suma de los residuos

Por lo tanto, se está actualizando las predicciones de modo que la suma de los residuos es cercana a 0 (o mínimo) y los valores pronosticados estarán lo suficientemente cerca de los valores reales.

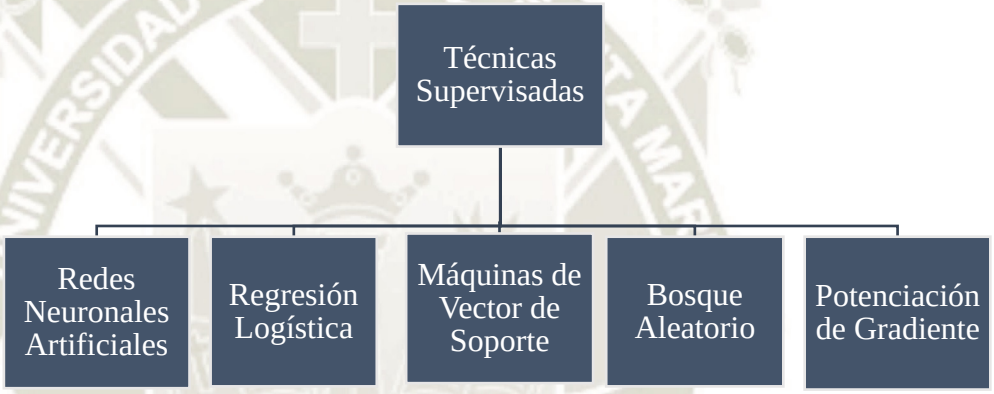


Figura 32. Cuadro de técnicas supervisadas seleccionadas

1.3.4. Aplicación de técnicas supervisadas

Para aplicar estas técnicas se ha utilizado el lenguaje de programación Python, la herramienta Júpiter y el IDE Pycharm. Algunas de estas técnicas se pueden compilar de forma paralela utilizando el GPU de la computadora para que el proceso sea más rápido, por ello se ha utilizado una laptop con un procesador Core I7 de 8ª generación y una tarjeta gráfica NViDIA GeForce GTX 1060.

Para encontrar la mayor precisión de cada una de las técnicas se realizaron una serie de pruebas cambiando los parámetros de estas. Al inicio se utilizaron los parámetros por defecto como indica la documentación de la librería Scikit-learn (2019) y también se utilizaron valores superiores e inferiores a estos. Los parámetros son los siguientes:

Tabla 5.

Parámetros de las técnicas con los que se realizaron las pruebas

Parámetros	**Descripción**
cv	Determina la estrategia de división de validación cruzada. Los posibles valores para cv son: ninguno, para usar la validación cruzada predeterminada que es la triple, para especificar el número de divisiones.
C	Determina la regularización, mientras los valores sean más pequeños, la regularización será más fuerte. Su valor por defecto es 1.
tol	Tolerancia para los criterios, valor por defecto es 0.0001
max_iter	Número máximo de iteraciones para la convergencia. Su valor por defecto es 100.
epochs	Separa el entrenamiento en distintas fases, lo cual es útil para el registro y la evaluación periódica.
batch_size:	El “batch” es un conjunto de n elementos. Los elementos en un “batch” se procesan de forma independiente, en paralelo. Cuanto más grande sea el “batch”, mejor será la aproximación;

	sin embargo, tardará más en procesarse y solo dará como resultado una actualización.
gamma	Determina la regularización. Su valor por defecto es 0.
max_depth	Determina la profundidad máxima de los estimadores de regresión individual. La profundidad máxima limita el número de nodos en el árbol. Su valor por defecto es 3.
n_estimators	El número de etapas de potenciaciones para realizar. Su valor por defecto es 100.
min_samples_split	El número mínimo de elementos necesarios para dividir un nodo interno. Su valor por defecto es 2.
learning_rate	Reduce la contribución de cada árbol mediante el aprendizaje de velocidad. Su valor por defecto es 0.1.

Para todas las técnicas se aplicó la librería GridSearchCV, esta realiza una búsqueda exhaustiva sobre los parámetros especificados para estimar cual es el que otorga la mejor precisión. No obstante, se realizaron pruebas con cada parámetro para saber cuál es el resultado que otorga.

A. Redes Neuronales Artificiales

Se utilizó la librería GridSearchCV para encontrar el mejor resultado de los parámetros:

```
classifier = KerasClassifier(build_fn=build_classifier)
parameters = {'batch_size': [25, 32], 'epochs': [100, 250],
              'optimizer': ['rmsprop', 'SGD'], 'h1_size': [6], 'h2_size': [6]}
grid_search = GridSearchCV(estimator=classifier, param_grid=parameters,
                           scoring='accuracy', cv=10, n_jobs=1)
grid_search = grid_search.fit(X=X_train, y=y_train)
best_parameters = grid_search.best_params_
best_accuracy = grid_search.best_score_
print('Best Parameters: %s' % best_parameters)
print('Best Accuracy: %s' % best_accuracy)
```

```
220/220 [==============================] - 0s 127us/step - loss: 0.0957 - acc: 0.9727
Epoch 243/250
220/220 [==============================] - 0s 122us/step - loss: 0.0953 - acc: 0.9727
Epoch 244/250
220/220 [==============================] - 0s 141us/step - loss: 0.0960 - acc: 0.9727
Epoch 245/250
220/220 [==============================] - 0s 141us/step - loss: 0.0952 - acc: 0.9727
Epoch 246/250
220/220 [==============================] - 0s 122us/step - loss: 0.0948 - acc: 0.9727
Epoch 247/250
220/220 [==============================] - 0s 122us/step - loss: 0.0944 - acc: 0.9727
Epoch 248/250
220/220 [==============================] - 0s 122us/step - loss: 0.0943 - acc: 0.9727
Epoch 249/250
220/220 [==============================] - 0s 127us/step - loss: 0.0938 - acc: 0.9727
Epoch 250/250
220/220 [==============================] - 0s 127us/step - loss: 0.0932 - acc: 0.9727
Best Parameters: {'batch_size': 25, 'epochs': 250, 'h1_size': 6, 'h2_size': 6, 'optimizer': 'rmsprop'}
Best Accuracy: 0.8772727272727273
```

Figura 33. Resultado de los mejores parámetros de la Red Neuronal Artificial

Se comprobó el resultado de la mejor técnica utilizando esos parámetros:

```
classifier = KerasClassifier(build_fn=build_classifier)
parameters = {'batch_size': [8], 'epochs': [250], 'optimizer': ['rmsprop'],
              'h1_size': [6], 'h2_size': [6]}
grid_search = GridSearchCV(estimator=classifier, param_grid=parameters, scoring='accuracy', cv=10, n_jobs=1)
grid_search = grid_search.fit(X=X_train, y=y_train)
best_parameters = grid_search.best_params_
best_accuracy = grid_search.best_score_
print('Best Parameters: %s' % best_parameters)
print('Best Accuracy: %s' % best_accuracy)
```

```
220/220 [==============================] - 0s 349us/step - loss: 0.0583 - acc: 0.9773
Epoch 243/250
220/220 [==============================] - 0s 331us/step - loss: 0.0584 - acc: 0.9773
Epoch 244/250
220/220 [==============================] - 0s 304us/step - loss: 0.0567 - acc: 0.9727
Epoch 245/250
220/220 [==============================] - 0s 322us/step - loss: 0.0580 - acc: 0.9773
Epoch 246/250
220/220 [==============================] - 0s 313us/step - loss: 0.0571 - acc: 0.9727
Epoch 247/250
220/220 [==============================] - 0s 326us/step - loss: 0.0556 - acc: 0.9773
Epoch 248/250
220/220 [==============================] - 0s 322us/step - loss: 0.0570 - acc: 0.9773
Epoch 249/250
220/220 [==============================] - 0s 326us/step - loss: 0.0570 - acc: 0.9773
Epoch 250/250
220/220 [==============================] - 0s 317us/step - loss: 0.0561 - acc: 0.9818
Best Parameters: {'batch_size': 8, 'epochs': 250, 'h1_size': 6, 'h2_size': 6, 'optimizer': 'rmsprop'}
Best Accuracy: 0.8772727272727273
```

Figura 34. Comprobación de los mejores parámetros de la Red Neuronal Artificial

B. Regresión Logística

Utilizando la librería GridSearchCV para encontrar el mejor resultado de los parámetros:

```
# Regresión Logística
param_grid = {'C': [0.1,0.5,1,10,50,100], 'max_iter': [250], 'fit_intercept':[True],'intercept_scaling':[1],
              'penalty':['l2'], 'tol':[0.00001,0.0001,0.000001]}
log_primal_Grid = GridSearchCV(LogisticRegression(solver='lbfgs'),param_grid, cv=10, refit=True, verbose=0)
log_primal_Grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(log_primal_Grid)
```

```
0.8772727272727273
{'C': 0.1, 'fit_intercept': True, 'intercept_scaling': 1, 'max_iter': 250, 'penalty': 'l2', 'tol': 1e-05}
LogisticRegression(C=0.1, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=250,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=1e-05, verbose=0,
                   warm_start=False)
```

Figura 35. Resultado de los mejores parámetros de Regresión Logística

Se comprobó el resultado solo con los parámetros determinados de la librería GridSearchCV:

```
# Regresión Logística
param_grid = {'C': [0.1], 'max_iter': [250], 'fit_intercept':[True],'intercept_scaling':[1],
              'penalty':['l2'], 'tol':[0.1]}
log_primal_Grid = GridSearchCV(LogisticRegression(solver='lbfgs'),param_grid, cv=10, refit=True, verbose=0)
log_primal_Grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(log_primal_Grid)
```

```
0.8772727272727273
{'C': 0.1, 'fit_intercept': True, 'intercept_scaling': 1, 'max_iter': 250, 'penalty': 'l2', 'tol': 0.1}
LogisticRegression(C=0.1, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=250,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.1, verbose=0,
                   warm_start=False)
```

Figura 36. Comprobación de los mejores resultados de la Regresión Logística

Al elevar el parámetro C, se reduce la precisión:

```
# Fit primal logistic regression
param_grid = {'C': [0.5], 'max_iter': [250], 'fit_intercept':[True],'intercept_scaling':[1],
              'penalty':['l2'], 'tol':[0.1]}
log_primal_Grid = GridSearchCV(LogisticRegression(solver='lbfgs'),param_grid, cv=10, refit=True, verbose=0)
log_primal_Grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(log_primal_Grid)
```

```
0.8681818181818182
{'C': 0.5, 'fit_intercept': True, 'intercept_scaling': 1, 'max_iter': 250, 'penalty': 'l2', 'tol': 0.1}
LogisticRegression(C=0.5, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=250,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.1, verbose=0,
                   warm_start=False)
```

Figura 37. Primera prueba con diferentes parámetros de la Regresión Logística

Si aumentamos el parámetro C, las iteraciones y reducimos tol, se reduce aún más la precisión:

```
# Fit primal logistic regression
param_grid = {'C': [50], 'max_iter': [500], 'fit_intercept':[True],'intercept_scaling':[1],
              'penalty':['l2'], 'tol':[0.00001]}
log_primal_Grid = GridSearchCV(LogisticRegression(solver='lbfgs'),param_grid, cv=10, refit=True, verbose=0)
log_primal_Grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(log_primal_Grid)
```

```
0.8363636363636363
{'C': 50, 'fit_intercept': True, 'intercept_scaling': 1, 'max_iter': 500, 'penalty': 'l2', 'tol': 1e-05}
LogisticRegression(C=50, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=500,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=1e-05, verbose=0,
                   warm_start=False)
```

Figura 38. Segunda prueba con diferentes parámetros de la Regresión Logística

C. Máquinas de Vector de Soporte

Utilizando la librería GridSearchCV para encontrar el mejor resultado de los parámetros:

```
# Fit SVM with RBF Kernel
param_grid = {'C': [0.5,100,150], 'gamma': [0.1,0.01,0.001],'probability':[True],'kernel': ['rbf']}
SVM_grid = GridSearchCV(SVC(), param_grid, cv=10, refit=True, verbose=0)
SVM_grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(SVM_grid)
```

```
0.8772727272727273
{'C': 0.5, 'gamma': 0.1, 'kernel': 'rbf', 'probability': True}
SVC(C=0.5, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.1, kernel='rbf',
    max_iter=-1, probability=True, random_state=None, shrinking=True, tol=0.001,
    verbose=False)
```

Figura 39. Resultado de los mejores parámetros de Máquinas de Vector de Soporte

Se comprobó el resultado solo con los parámetros determinados de la librería GridSearchCV:

```
# Fit SVM with pol kernel
param_grid = {'C': [0.5], 'gamma': [0.1],'probability':[True],'kernel': ['poly'],'degree':[2] }
SVM_grid = GridSearchCV(SVC(), param_grid, cv=3, refit=True, verbose=0)
SVM_grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(SVM_grid)
```

```
0.8772727272727273
{'C': 0.5, 'degree': 2, 'gamma': 0.1, 'kernel': 'poly', 'probability': True}
SVC(C=0.5, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=2, gamma=0.1, kernel='poly',
    max_iter=-1, probability=True, random_state=None, shrinking=True, tol=0.001,
    verbose=False)
```

Figura 40. Comprobación de los mejores resultados de la Máquina Vector de Soporte

Al aumentar el parámetro C, pero reduciendo la gamma se mantiene la precisión:

```
# Fit SVM with pol kernel
param_grid = {'C': [1], 'gamma': [0.001],'probability':[True],'kernel': ['poly'],'degree':[3] }
SVM_grid = GridSearchCV(SVC(), param_grid, cv=10, refit=True, verbose=0)
SVM_grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(SVM_grid)
```

```
0.8772727272727273
{'C': 1, 'degree': 3, 'gamma': 0.001, 'kernel': 'poly', 'probability': True}
SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.001, kernel='poly',
    max_iter=-1, probability=True, random_state=None, shrinking=True, tol=0.001,
    verbose=False)
```

Figura 41. Primera prueba con diferentes parámetros de la MVS

Si aumentamos mucho el parámetro C, la precisión se reduce drásticamente:

```
# Fit SVM with pol kernel
param_grid = {'C': [50], 'gamma': [0.1],'probability':[True],'kernel': ['poly'],'degree':[2] }
SVM_grid = GridSearchCV(SVC(), param_grid, cv=10, refit=True, verbose=0)
SVM_grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(SVM_grid)
```

```
0.8227272727272728
{'C': 50, 'degree': 2, 'gamma': 0.1, 'kernel': 'poly', 'probability': True}
SVC(C=50, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=2, gamma=0.1, kernel='poly',
    max_iter=-1, probability=True, random_state=None, shrinking=True, tol=0.001,
    verbose=False)
```

Figura 42. Segunda prueba con diferentes parámetros de la MVS

D. Bosque Aleatorio

Utilizando la librería GridSearchCV para encontrar el mejor resultado de los parámetros:

```
# Fit random forest classifier
param_grid = {'max_depth': [3, 5, 6, 7, 8], 'max_features': [2,4,6,7,8,9],'n_estimators':[50,100],
              'min_samples_split': [3, 5, 6, 7]}
RanFor_grid = GridSearchCV(RandomForestClassifier(), param_grid, cv=5, refit=True, verbose=0)
RanFor_grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(RanFor_grid)
```

```
0.8818181818181818
{'max_depth': 6, 'max_features': 2, 'min_samples_split': 6, 'n_estimators': 50}
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=6, max_features=2, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=6,
                       min_weight_fraction_leaf=0.0, n_estimators=50,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Figura 43. Resultado de los mejores parámetros de Bosque Aleatorio

Se comprobó el resultado solo con los parámetros determinados de la librería GridSearchCV:

```
# Fit random forest classifier
param_grid = {'max_depth': [6], 'max_features': [2],'n_estimators':[50],
              'min_samples_split': [6]}
RanFor_grid = GridSearchCV(RandomForestClassifier(), param_grid, cv=5, refit=True, verbose=0)
RanFor_grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(RanFor_grid)
```

```
0.8818181818181818
{'max_depth': 6, 'max_features': 2, 'min_samples_split': 6, 'n_estimators': 50}
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=6, max_features=2, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=6,
                       min_weight_fraction_leaf=0.0, n_estimators=50,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Figura 44. Comprobación de los mejores resultados de Bosque Aleatorio

Al elevar los parámetros "max_depth", "max_features" y "min_samples_split", se reduce la precisión:

```
# Fit random forest classifier
param_grid = {'max_depth': [8], 'max_features': [9],'n_estimators':[50],'min_samples_split': [7]}
RanFor_grid = GridSearchCV(RandomForestClassifier(), param_grid, cv=10, refit=True, verbose=0)
RanFor_grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(RanFor_grid)
```

```
0.8681818181818182
{'max_depth': 8, 'max_features': 9, 'min_samples_split': 7, 'n_estimators': 50}
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=8, max_features=9, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=7,
                       min_weight_fraction_leaf=0.0, n_estimators=50,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Figura 45. Primera prueba con diferentes parámetros de Bosque Aleatorio

Si bajamos los valores de los mismos parámetros, se reduce la precisión:

```
# Fit random forest classifier
param_grid = {'max_depth': [3], 'max_features': [2],'n_estimators':[50],'min_samples_split': [3]}
RanFor_grid = GridSearchCV(RandomForestClassifier(), param_grid, cv=5, refit=True, verbose=0)
RanFor_grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(RanFor_grid)
```

```
0.8772727272727273
{'max_depth': 3, 'max_features': 2, 'min_samples_split': 3, 'n_estimators': 50}
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=3, max_features=2, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=3,
                       min_weight_fraction_leaf=0.0, n_estimators=50,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Figura 46. Segunda prueba con diferentes parámetros de Bosque Aleatorio

E. Potenciación de Gradiente

Utilizando la librería GridSearchCV para encontrar el mejor resultado de los parámetros:

```
# Fit Extreme Gradient boosting classifier
param_grid = {'max_depth': [5,6,7,8], 'gamma': [0.01,0.001,0.001],'min_child_weight':[1,5,10],
              'learning_rate': [0.05,0.1, 0.2, 0.3], 'n_estimators':[5,10,20,100]}
xgb_grid = GridSearchCV(XGBClassifier(), param_grid, cv=5, refit=True, verbose=0)
xgb_grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(xgb_grid)
```

```
0.9
{'gamma': 0.01, 'learning_rate': 0.2, 'max_depth': 6, 'min_child_weight': 1, 'n_estimators': 100}
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0.01,
              learning_rate=0.2, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=None, n_estimators=100, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=1, verbosity=1)
```

Figura 47. Resultado de los mejores parámetros de Potenciación de Gradiente

Se comprobó el resultado solo con los parámetros determinados de la librería GridSearchCV:

```
# Fit Extreme Gradient boosting classifier
param_grid = {'max_depth': [6], 'gamma': [0.01],'min_child_weight':[1], 'learning_rate': [0.2], 'n_estimators':[100]}
xgb_grid = GridSearchCV(XGBClassifier(), param_grid, cv=5, refit=True, verbose=0)
xgb_grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(xgb_grid)
```

```
0.9
{'gamma': 0.01, 'learning_rate': 0.2, 'max_depth': 6, 'min_child_weight': 1, 'n_estimators': 100}
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0.01,
              learning_rate=0.2, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=None, n_estimators=100, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=1, verbosity=1)
```

Figura 48. Comprobación de los mejores resultados de Potenciación de Gradiente

Al reducir el parámetro “gamma” y elevar los parámetros “max_depth”, “min_child_weight” y “learning_rate”, se reduce la precisión:

```
# Fit Extreme Gradient boosting classifier
param_grid = {'max_depth': [8], 'gamma': [0.0001],'min_child_weight':[10], 'learning_rate': [0.3], 'n_estimators':[100]}
xgb_grid = GridSearchCV(XGBClassifier(), param_grid, cv=5, refit=True, verbose=0)
xgb_grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(xgb_grid)
```

```
0.8772727272727273
{'gamma': 0.0001, 'learning_rate': 0.3, 'max_depth': 8, 'min_child_weight': 10, 'n_estimators': 100}
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0.0001,
              learning_rate=0.3, max_delta_step=0, max_depth=8,
              min_child_weight=10, missing=None, n_estimators=100, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=1, verbosity=1)
```

Figura 49. Primera prueba con diferentes parámetros de Potenciación de Gradiente

La precisión se reduce drásticamente si reducimos en gran medida el parámetro “n_estimators”:

```
# Fit Extreme Gradient Boosting classifier
param_grid = {'max_depth': [5], 'gamma': [0.01],'min_child_weight':[1], 'learning_rate': [0.05], 'n_estimators':[5]}
xgb_grid = GridSearchCV(XGBClassifier(), param_grid, cv=5, refit=True, verbose=0)
xgb_grid.fit(df_train.loc[:, df_train.columns != 'Renuncio'],df_train.Renuncio)
best_model(xgb_grid)

0.8454545454545455
{'gamma': 0.01, 'learning_rate': 0.05, 'max_depth': 5, 'min_child_weight': 1, 'n_estimators': 5}
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0.01,
              learning_rate=0.05, max_delta_step=0, max_depth=5,
              min_child_weight=1, missing=None, n_estimators=5, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=1, verbosity=1)
```

Figura 50. Segunda prueba con diferentes parámetros de Potenciación de Gradiente

1.3.5. Verificación de resultados

Para verificar los resultados obtenidos de estas técnicas se ha optado por utilizar un generador de datos sintéticos porque la cantidad de datos reales es muy pequeña entonces a partir de la herramienta DataSynthetizer se generó un archivo con 10000 datos correlacionados a los datos originales.

La distribución de los datos sintéticos se detalla a continuación:

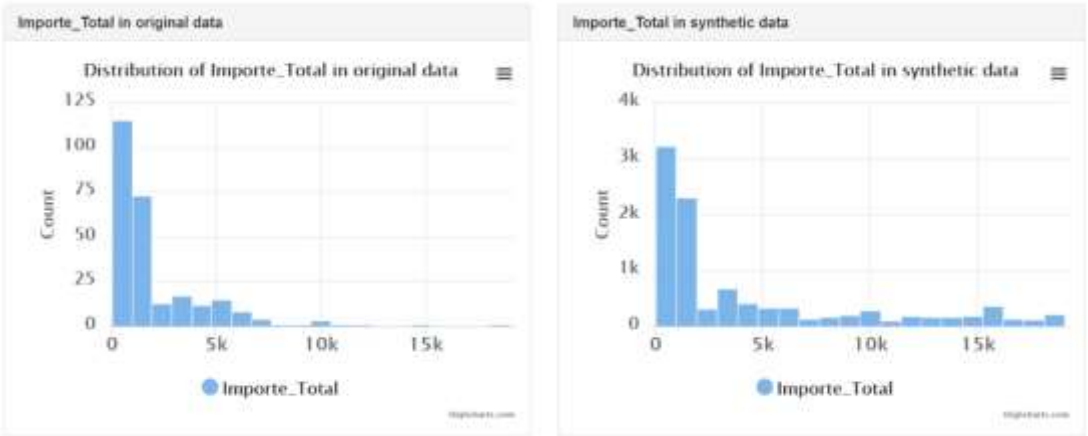


Figura 51. Distribución de datos del total de importes en los datos originales y los datos sintéticos

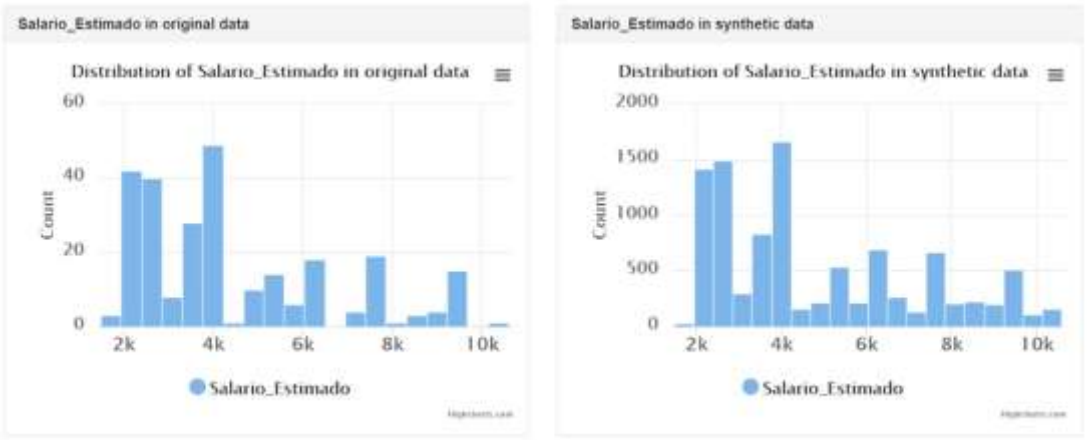


Figura 52. Distribución de datos del salario estimado en los datos originales y los datos sintéticos

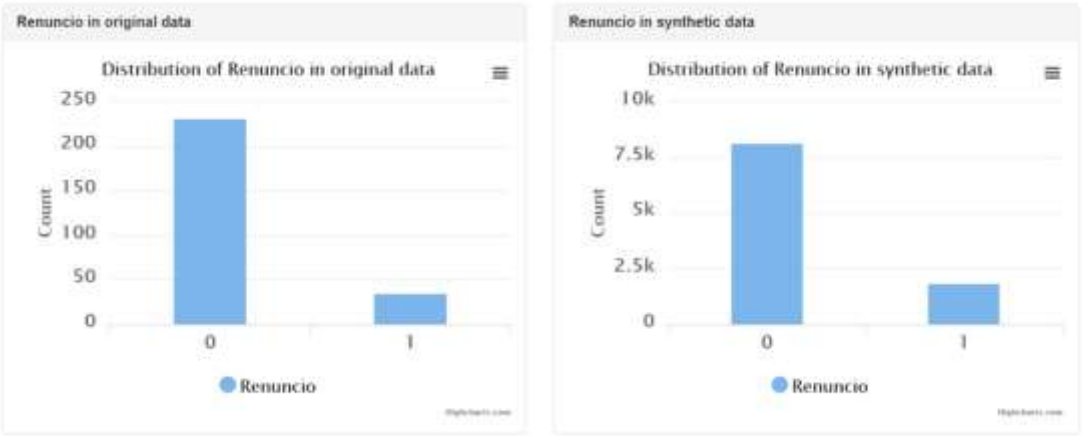


Figura 53. Distribución de datos de las renuncias en los datos originales y los datos sintéticos

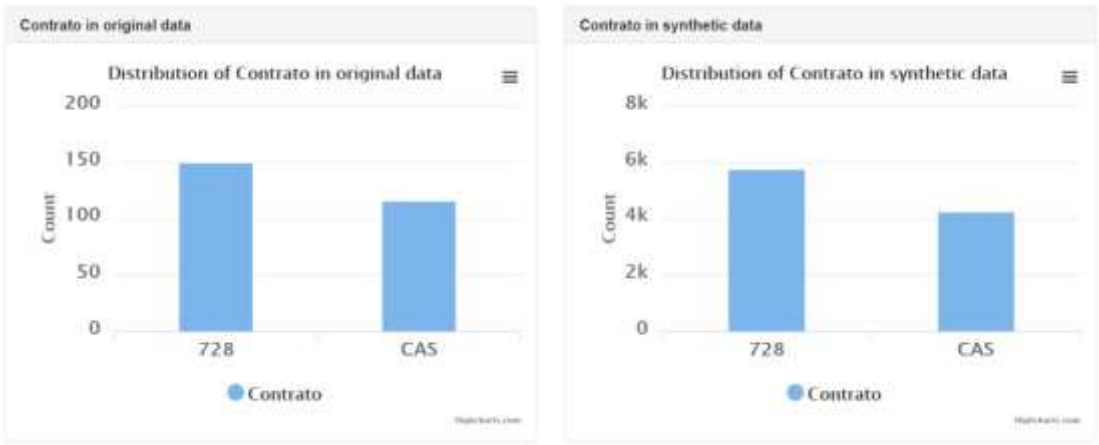


Figura 54. Distribución de datos del tipo de contrato en los datos originales y los datos sintéticos

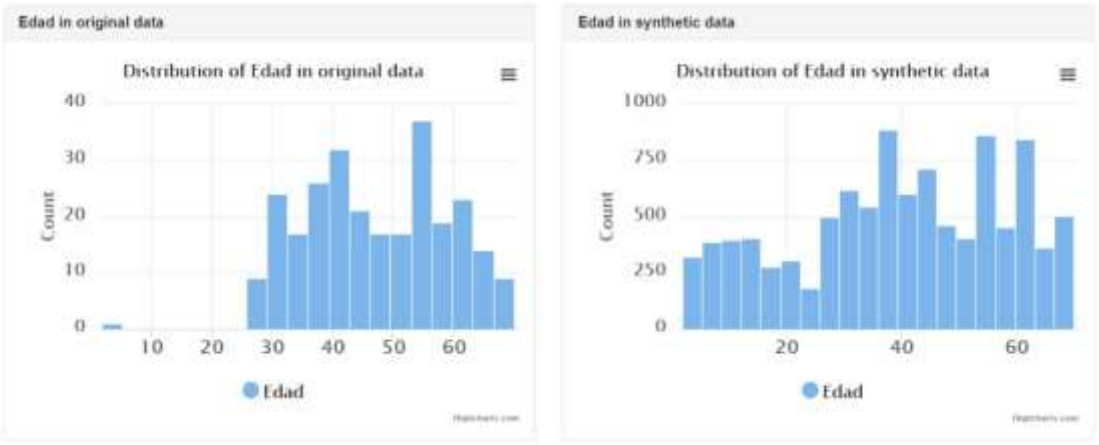


Figura 55. Distribución de datos de la edad en los datos originales y los datos sintéticos

Tabla 6.

Resultados de las técnicas con datos reales y datos sintéticos

Técnica Supervisada	**Resultado de precisión con datos reales**	**Resultado de precisión con datos sintéticos**
Redes Neuronales Artificiales	87.3%	84.2%
Regresión Logística	87.7%	83.5%
Máquinas de Vector de Soporte	87.7%	84%
Bosque Aleatorio	88.2%	88.9%
Potenciación de Gradiente	90%	91.5%

1.3.6. Validación de las técnicas supervisadas

Se validaron las técnicas de dos formas diferentes, la primera de forma local, es decir, con las técnicas depurando en el computador y la segunda a través de un prototipo web en donde se desarrolló una interfaz gráfica con el que el usuario pueda ingresar los campos del socio, cabe indicar que las técnicas se depuran en la nube.

A. Validación con casos reales de los socios

Para el proceso de validación se ha puesto a prueba las técnicas, para ello con dos casos de estudio, el primero con datos reales de un socio que se retiró de la cooperativa y otro caso en el que se mantuvo. Los atributos del socio que se retiró se muestran en la Tabla 7 y los atributos del socio que se mantuvo se muestran en la Tabla 8.

Tabla 7.

Atributos de un socio que se retiró de la cooperativa

Atributo	**Valor**
Sucursal	NORTE
Contrato	728
Sexo	Femenino
Estado civil	Soltera
Edad	48
Tiempo en la cooperativa	22
Importe total	10650
Préstamo con la cooperativa	Sí
Deudas con la cooperativa	Sí
Salario estimado	4200

Tabla 8.

Atributos de un socio que se mantuvo en la cooperativa

Atributo	**Valor**
Sucursal	CENTRO
Contrato	728
Sexo	Masculino
Estado civil	Casado
Edad	42
Tiempo en la cooperativa	21
Importe total	3500
Préstamo con la cooperativa	No

Deudas con la cooperativa	Sí
Salario estimado	2200

Se realizaron estas pruebas en los modelos con datos reales y el modelo con datos sintéticos, obteniendo los siguientes resultados:

```
# Sucursal: NORTE -> 0 0, columnas 0 y 1
# Contrato: 728 -> 0, columna 2
# Sexo: Femenino -> 0, columna 3
# Estado civil: Soltero -> 1, columna 4
# Edad: 48 -> columna 5
# Tiempo en la cooperativa: 22 meses, columna 6
# Importe total: 10650, columna 7
# Préstamo con la cooperativa: Sí, columna 8
# Deudas con la cooperativa: Sí, columna 9
# Salario estimado: 4200, columna 10
# Renuncia: Sí
new_prediction = grid_search.predict(sc.transform(
    np.array([[0, 0, 0, 0, 1, 48, 22, 10650, 1, 1, 4200]])))

new_prediction = (new_prediction > 0.5)
print (new_prediction)
```

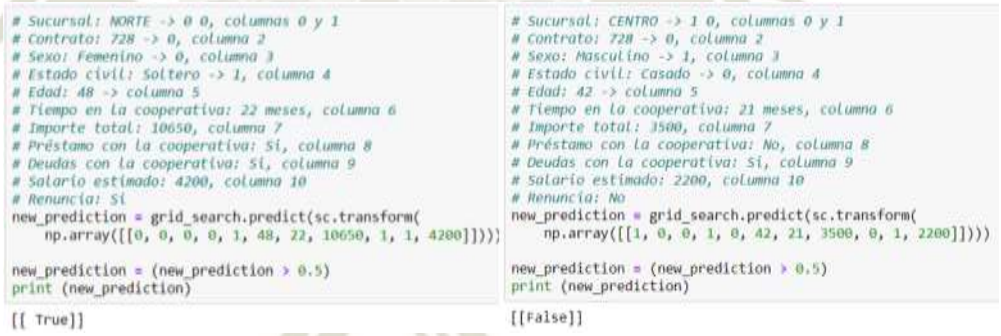
```
[[False]]
```

```
# Sucursal: CENTRO -> 1 0, columnas 0 y 1
# Contrato: 728 -> 0, columna 2
# Sexo: Masculino -> 1, columna 3
# Estado civil: Casado -> 0, columna 4
# Edad: 42 -> columna 5
# Tiempo en la cooperativa: 21 meses, columna 6
# Importe total: 3500, columna 7
# Préstamo con la cooperativa: No, columna 8
# Deudas con la cooperativa: Sí, columna 9
# Salario estimado: 2200, columna 10
# Renuncia: No
new_prediction = grid_search.predict(sc.transform(
    np.array([[1, 0, 0, 1, 0, 42, 21, 3500, 0, 1, 2200]])))

new_prediction = (new_prediction > 0.5)
print (new_prediction)
```

```
[[False]]
```

Figura 56. Resultados de la predicción con datos reales



```
# Sucursal: NORTE -> 0 0, columnas 0 y 1
# Contrato: 728 -> 0, columna 2
# Sexo: Femenino -> 0, columna 3
# Estado civil: Soltero -> 1, columna 4
# Edad: 48 -> columna 5
# Tiempo en la cooperativa: 22 meses, columna 6
# Importe total: 10650, columna 7
# Préstamo con la cooperativa: Sí, columna 8
# Deudas con la cooperativa: Sí, columna 9
# Salario estimado: 4200, columna 10
# Renuncia: Sí
new_prediction = grid_search.predict(sc.transform(
    np.array([[0, 0, 0, 0, 1, 48, 22, 10650, 1, 1, 4200]])))

new_prediction = (new_prediction > 0.5)
print (new_prediction)
```

```
[[ True]]
```

```
# Sucursal: CENTRO -> 1 0, columnas 0 y 1
# Contrato: 728 -> 0, columna 2
# Sexo: Masculino -> 1, columna 3
# Estado civil: Casado -> 0, columna 4
# Edad: 42 -> columna 5
# Tiempo en la cooperativa: 21 meses, columna 6
# Importe total: 3500, columna 7
# Préstamo con la cooperativa: No, columna 8
# Deudas con la cooperativa: Sí, columna 9
# Salario estimado: 2200, columna 10
# Renuncia: No
new_prediction = grid_search.predict(sc.transform(
    np.array([[1, 0, 0, 1, 0, 42, 21, 3500, 0, 1, 2200]])))

new_prediction = (new_prediction > 0.5)
print (new_prediction)
```

```
[[False]]
```

Figura 57. Resultados de la predicción con datos sintéticos

B. Prototipo para la validación de las técnicas supervisadas

Se desarrolló un prototipo en un entorno web para la validación de las técnicas, siguiendo el Modelo de Prototipos de Software

Gur (2019), define la metodología de Prototipo como un modelo de desarrollo de software en donde el prototipo es desarrollado, probado y luego se vuelve a desarrollar hasta que un modelo de prototipo es aceptado. Este modelo funciona mejor en escenarios donde no son conocidos todos los requerimientos del proyecto porque es iterativo, con constantes pruebas y errores que se dan entre el desarrollador y el cliente.

El modelo de prototipos consta de las siguientes etapas:

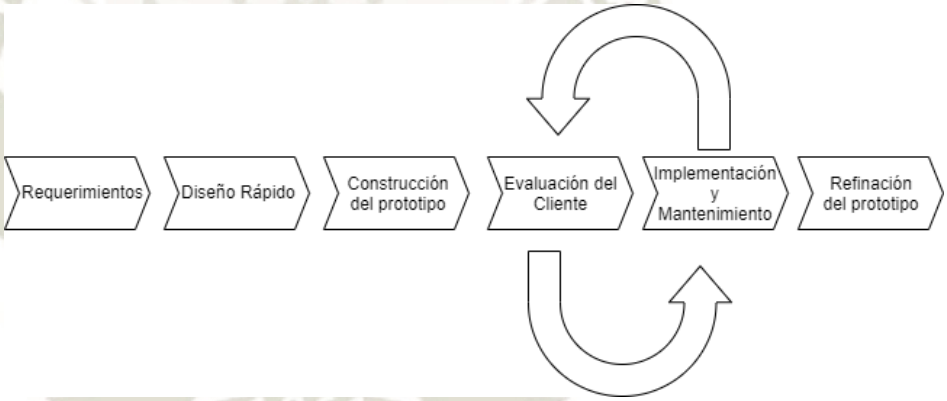


Figura 58. Etapas del Modelo de Prototipos según Gur (2019)

Para el presente trabajo de investigación solo se llegará a la etapa de construcción del prototipo porque el enfoque principal será de validar las técnicas supervisadas de aprendizaje automático, por lo cual no habrá una comunicación retroactiva con el cliente, para las pruebas ni refinar el prototipo.

- Requerimientos

En esta etapa se definieron los requerimientos del prototipo, los cuales se detallan en la siguiente tabla:

Tabla 9.

Requerimientos del prototipo

Requerimiento	**Descripción**
Registro de datos	El cliente debe ingresar los datos de los socios del que quiere realizar la predicción, por lo cual es necesario el uso de un formulario con todos los campos necesarios.
Entrenamiento de técnica	Se requiere que las técnicas sean implementadas en la nube y no en un servidor local para que pueda ser accedido remotamente. Además, se debe tener la capacidad para poder realizar el entrenamiento de los datos y realizar la predicción.
Generación de respuesta	Se requiere un servicio web que pueda acceder a la técnica y enviar los parámetros de los socios ingresados por el cliente, y a su vez, debe retornar una respuesta con la

	predicción de si el socio renuncia o no.
Pruebas	Se debe levantar al servicio web de en la nube y no localmente para que ser accedido desde cualquier lugar del mundo y realizar todas las pruebas necesarias validando la precisión de la técnica.

- Diseño Rápido

Se realizó un diseño simple del sistema consistiendo de un formulario con todos los campos necesarios para los datos de los socios que requieren las técnicas supervisadas de aprendizaje automático.

Una vez llenado el formulario y enviada la información, se mostrará un "modal" diciendo si el socio renunciará a la cooperativa o sino si ha renunciado.

Las pantallas del prototipo son las siguientes:

Figura 59. Pantalla del formulario

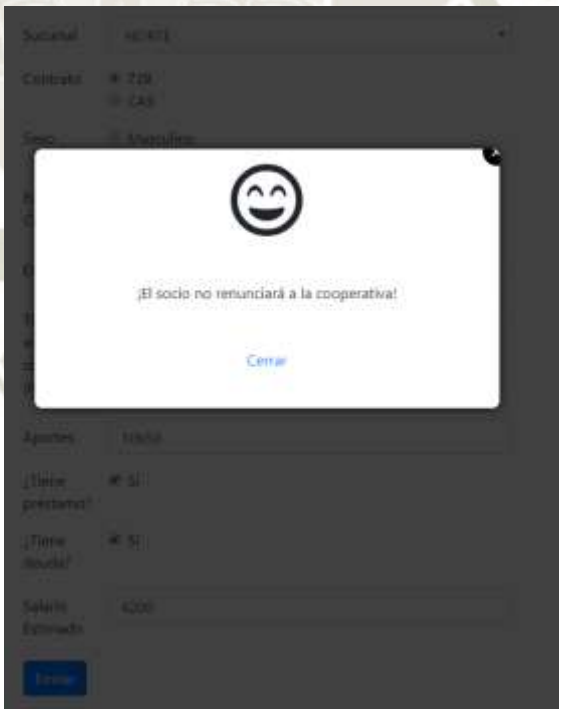


Figura 60. Modal de No Renuncia

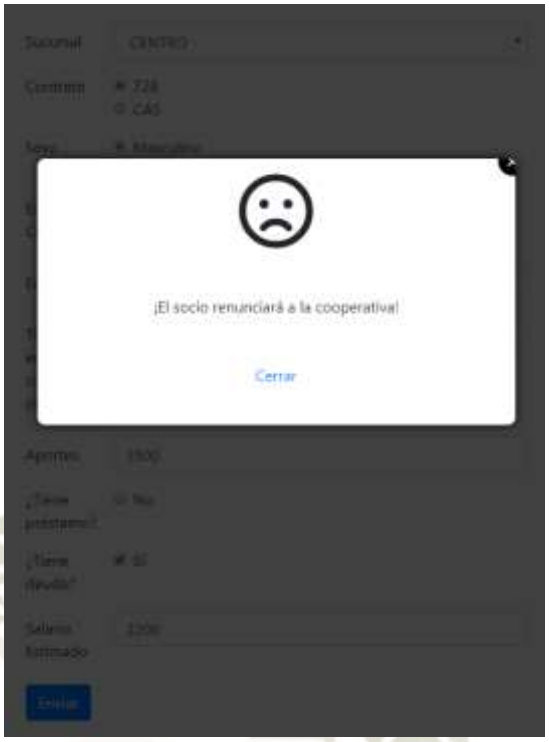


Figura 61. Modal de Renuncia

- Construcción del Prototipo

Para poder realizar la implementación del modelo se optó por utilizar las siguientes herramientas: SageMaker, Lambda y API Gateway todas de Amazon Web Services.

SageMaker

Según Amazon (2019), esta herramienta permite a los desarrolladores poder crear, realizar el entrenamiento e implementar modelos de aprendizaje automático rápidamente en la nube.

Lambda

Según Amazon (2019), esta herramienta permite ejecutar código sin la necesidad de utilizar servidores.

API Gateway

Según Amazon (2019), es un servicio para la crear, publicar, mantener y proteger una API a cualquier escala.

La arquitectura del prototipo será la siguiente:

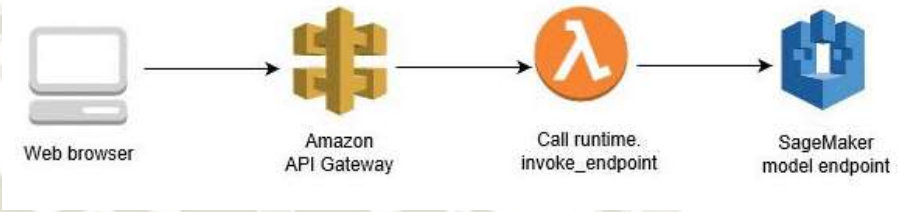


Figura 62. Arquitectura del Prototipo

SageMaker permite trabajar con Jupyter en la nube, por lo que se colocó todo el código necesario para la técnica y fue ejecutado, además de la creación del endpoint para recibir los datos por parte del usuario para realizar predicciones.

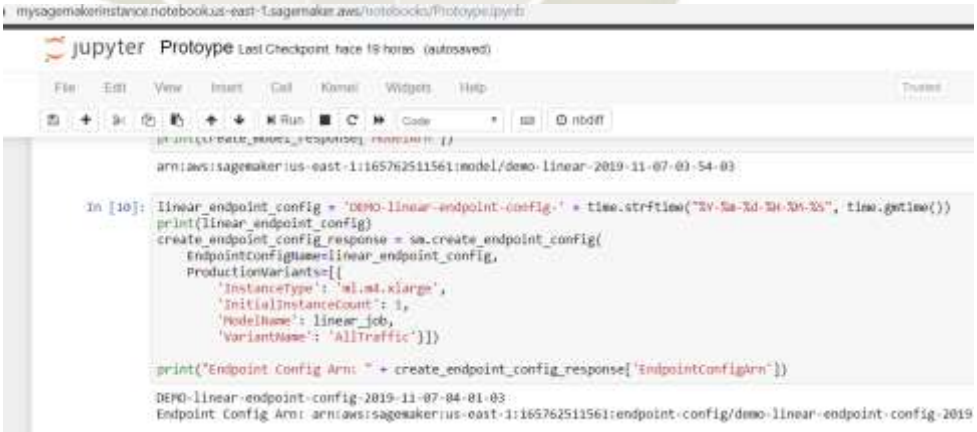


Figura 63. Creación del Endopint en SageMaker

Para poder enviar los parámetros del usuario al modelo de aprendizaje automático se utiliza la función predictMemberChurn que está implementada en Lambda, permitiendo desarrollar el prototipo sin la necesidad de un servidor.

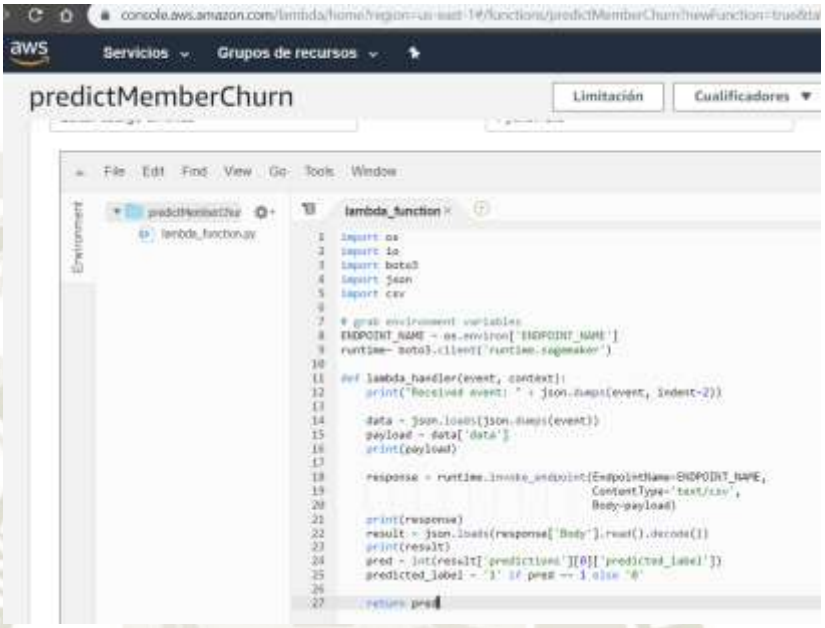


Figura 64. Función predictMemberChurn en Lambda

Para realizar el servicio web, se utilizó la herramienta API Getaway que invoca a la función Lambda predictMemberChurn. Esta nos crea una url en la nube donde podemos hacer los servicios web.

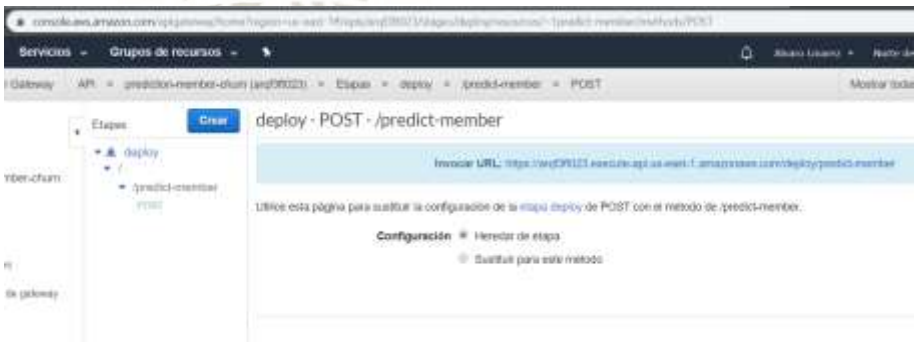


Figura 65. API Gateway implementada

Finalmente, se tuvo que invocar a este servicio web, por lo cual se utilizó jquery para el envío de los parámetros del formulario, se recibe la respuesta del API y si es 0, significa que el socio no renunciará, pero si la respuesta es 1, significa que el socio renunciará a la cooperativa, mostrando los 'modal' para cada caso descritos en el punto anterior.

```
function submitFormRenuncia(){
    var myForm = document.getElementById('form_renuncia');
    var json = toJSONString(myForm);
    $.ajax({
        url: 'https://arqf3ft023.execute-api.us-east-1.amazonaws.com/deploy/predict-member',
        type: 'post',
        dataType: 'json',
        data: json,
        success: function(data) {
            if (data === 1) {
                $('#modal_renuncia').modal();
            } else {
                $('#modal_no_renuncia').modal();
            }
        },
        error: function (error) {
            console.log(error);
        }
    });
}
```

Figura 66. Invocación del servicio web con la url del API

1.4. Estudio del desempeño de las técnicas

Para evaluar el desempeño de las técnicas se hizo una investigación sobre la complejidad de cada uno, tanto para el entrenamiento como para las pruebas. Además, se tomó el tiempo que requirió cada técnica en el desarrollo del trabajo de investigación.

1.4.1. Complejidad de las técnicas

Según Adamchick (2009), la complejidad está referida a la rapidez o lentitud de un algoritmo, además, define a la complejidad como una función numérica T(n): Tiempo contra el tamaño de la entrada n.

El objetivo de la complejidad es clasificar los algoritmos según sus desempeños. Se representa la función T(n) usando la notación *"big-O"* para expresar la complejidad de tiempo de ejecución de un algoritmo. Por ejemplo, la siguiente declaración (ver ecuación 11) de un algoritmo con complejidad de tiempo cuadrática.

$$T(n) = O(n^2) \tag{11}$$

Como indica Kernel (2018), para calcular la complejidad de las técnicas se colocó *n* a la cantidad de entrenamiento, *p* a la cantidad de características, n_{trees} al número de árboles, n_{sv}, al número de vectores de soporte y n_{li} al número de neuronas en cada capa *i* de la red neuronal, obteniendo los siguientes resultados en la Tabla 10.

Tabla 10

Complejidad de las técnicas utilizadas

Técnica	**Entrenamiento**	**Predicción**
Redes Neuronales Artificiales	$O(pn_{li})$	$O(pn_{l1} + n_{l1} + n_{l2} + \ldots)$
Regresión Logística	$O(p^2n + p^3)$	$O(p)$
Máquinas de Vector de Soporte	$O(n^2p + n^3)$	$O(n_{sv}p)$
Bosque Aleatorio	$O(n^2pn_{trees})$	$O(pn_{trees})$
Potenciación de Gradiente	$O(npn_{trees})$	$O(pn_{trees})$

Para depurar estas técnicas en el ámbito de hardware se utilizó una laptop con un procesador Core i7 de octava generación de 2.2 Ghz, una tarjeta gráfica NViDIA GEFORCE GTX 1060 de 6 GB y con una memoria RAM de 16 GB. A continuación, en la Tabla se muestra el tiempo que tomó cada una de las técnicas en la depuración.

Tabla 11

Tiempos estimados de cada técnica

Técnica	**Tiempo**
Redes Neuronales Artificiales	7 minutos y 17 segundos
Regresión Logística	5 minutos y 20 segundos
Máquinas de Vector de Soporte	6 minutos y 5 segundos
Bosque Aleatorio	5 minutos y 45 segundos
Potenciación de Gradiente	5 minutos y 35 segundos

Capítulo 2: Resultados

En este apartado se presentan los resultados de precisión y la matriz de confusión de cada técnica utilizada y luego, se analizan y se discuten dichos resultados.

2.1. Resultados de las técnicas supervisadas de aprendizaje automático

Para la aplicación de las técnicas supervisadas de aprendizaje automático elegidas que son redes neuronales artificiales, regresión logística, máquinas vectores soporte, bosque aleatorio y potenciación de gradiente se utilizaron datos de los socios reales y datos generados sintéticamente que fueron de 274 datos reales y 10000 datos sintéticos. De la cantidad de datos en ambos casos el 80% fue utilizado para entrenar y el 20% para realizar las pruebas. Las técnicas fueron implementadas con las librerías de aprendizaje automático de Python, para cada una de estas técnicas se detalla la matriz de confusión y la precisión que se obtiene a partir de ella.

A continuación, se mostrará un cuadro con los mejores resultados de las técnicas de aprendizaje automático elegidas utilizando como datos de entrada el archivo CSV con los datos de los socios que se preparó en el capítulo anterior.

Tabla 12.

Resultados de precisión de las técnicas empleadas

Técnica Supervisada	**Resultados de precisión de datos reales**	**Resultados de precisión de datos sintéticos**
Redes Neuronales Artificiales	87.3%	84.2%

Regresión Logística	87.7%	83.7%
Máquinas de Vector de Soporte	87.7%	85.8%
Bosque Aleatorio	88.2%	90.4%
Potenciación de Gradiente	90%	90.6%

2.1.1. Redes Neuronales Artificiales

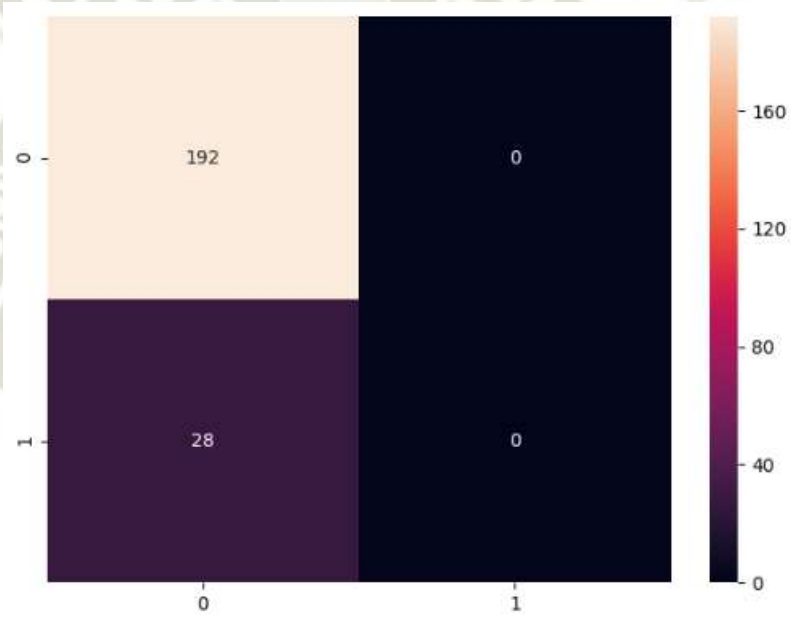


Figura 67. Matriz de confusión de la Red Neuronal Artificial con datos reales

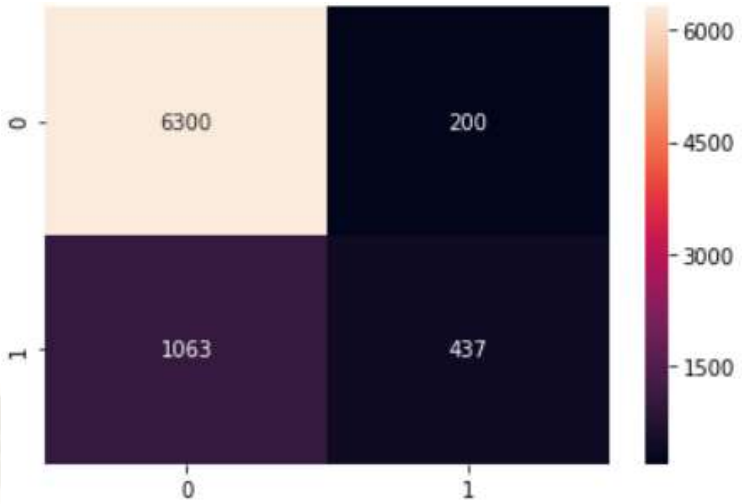


Figura 68. Matriz de confusión de la Red Neuronal Artificial con datos sintéticos

2.1.2. Regresión Logística

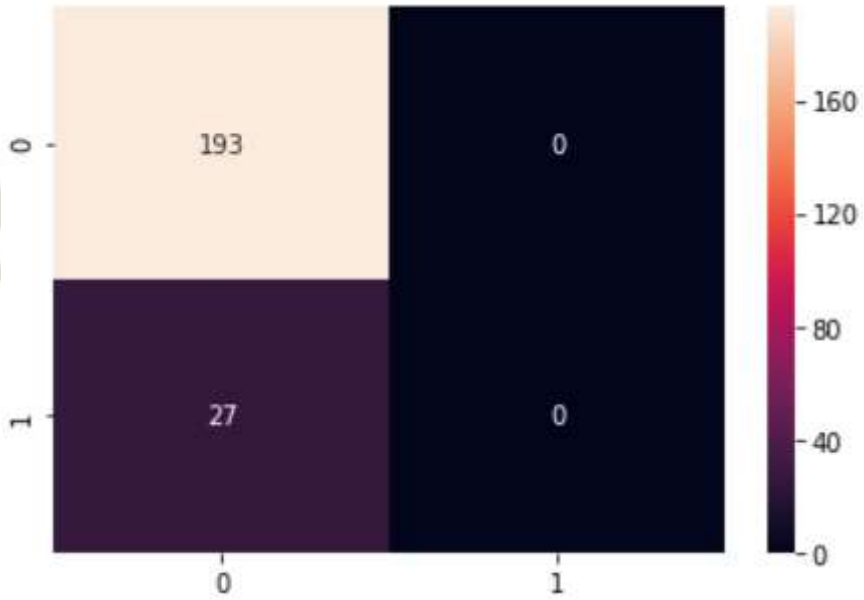


Figura 69. Matriz de confusión de Regresión Logística con datos reales

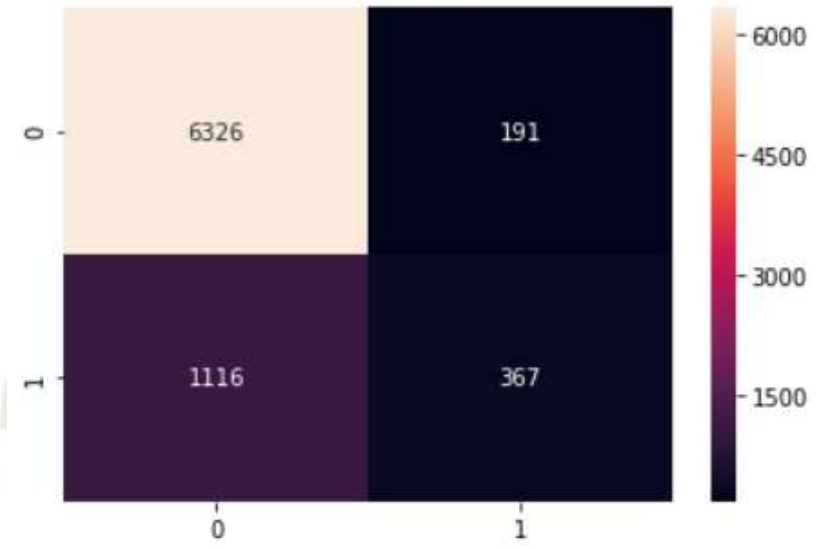


Figura 70. Matriz de confusión de Regresión Logística con datos sintéticos

2.1.3. Máquinas de Vector de Soporte

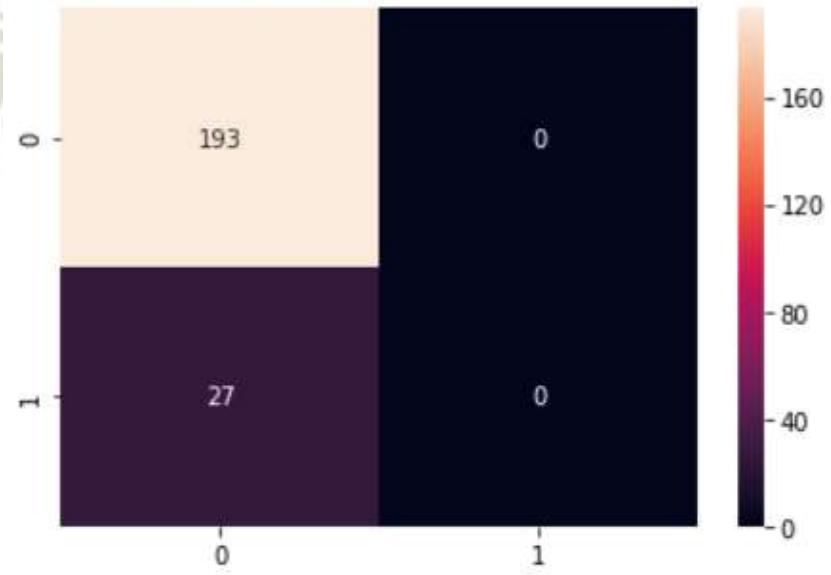


Figura 71. Matriz de confusión de la Máquina de Vector de Soporte con datos reales

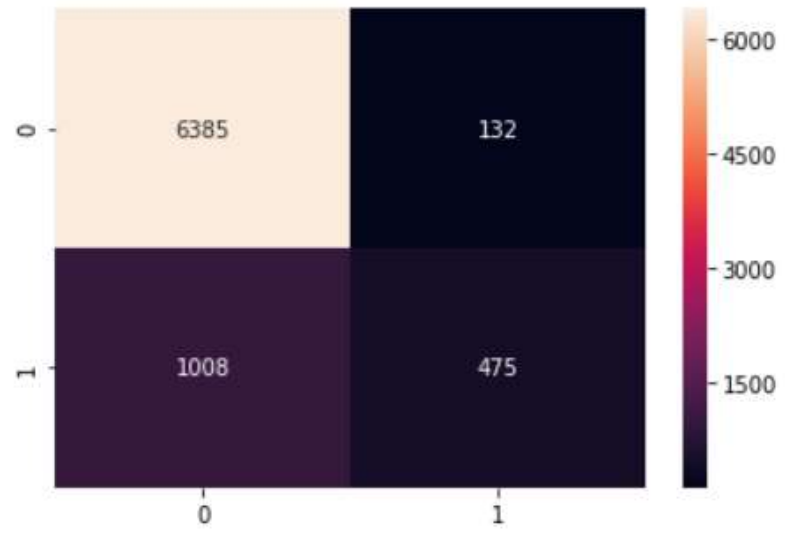


Figura 72. Matriz de confusión de la Máquina de Vector de Soporte con datos sintéticos

2.1.4. Bosque Aleatorio

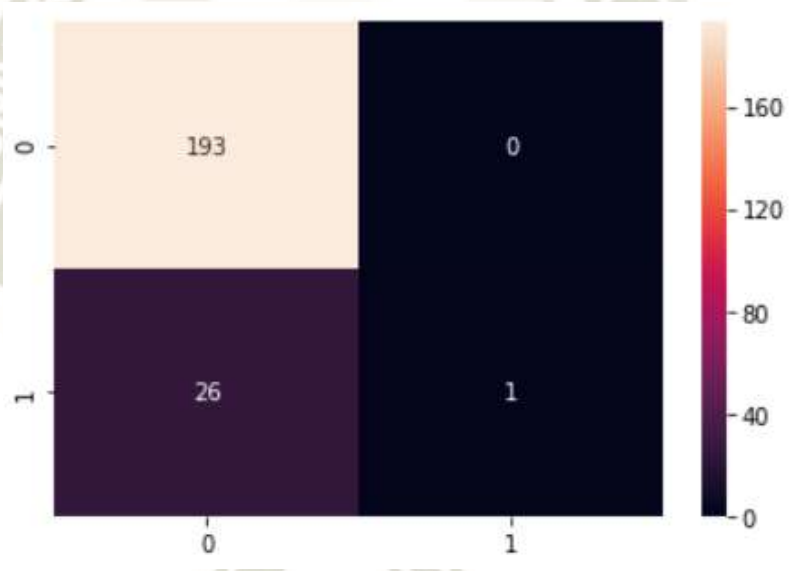


Figura 73. Matriz de confusión de la Bosque Aleatorio con datos reales

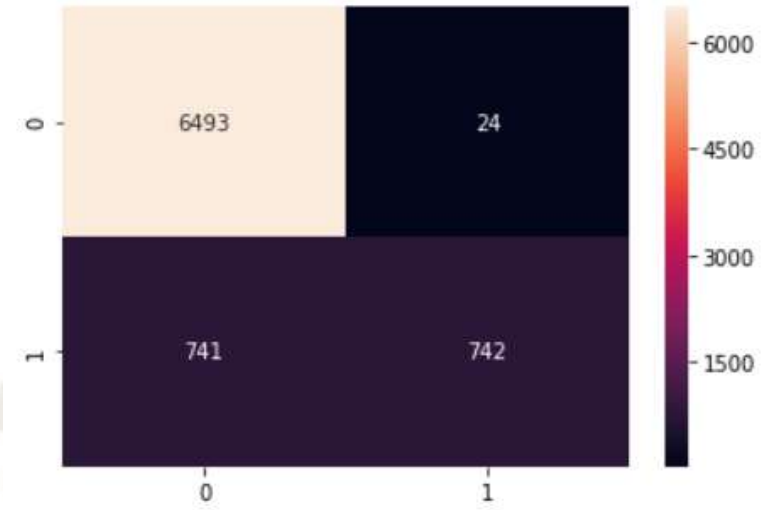


Figura 74. Matriz de confusión de la Bosque Aleatorio con datos sintéticos

2.1.5. Potenciación de Gradiente

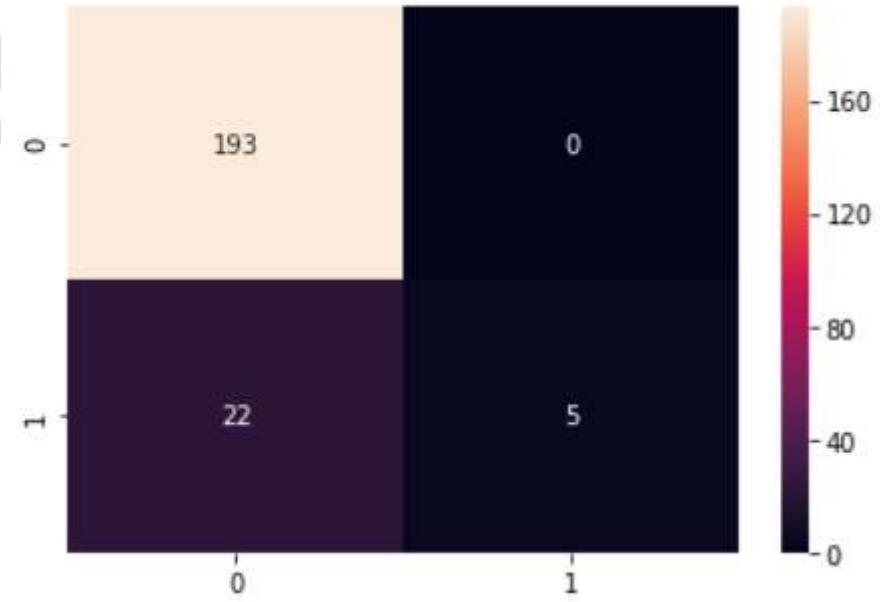


Figura 75. Matriz de confusión de la Potenciación de Gradiente con datos reales

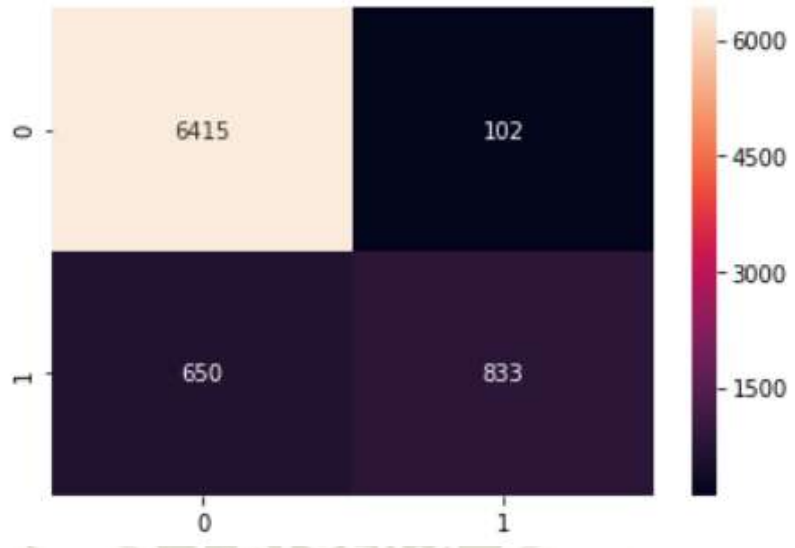


Figura 76. Matriz de confusión de la Potenciación de Gradiente con datos sintéticos

2.2. Análisis y discusión de los resultados

A partir de los gráficos del punto anterior se observa que los mejores resultados de precisión son de las técnicas de bosque aleatorio y de potenciación de gradiente para ambos casos, con datos reales y sintéticos. Con los datos reales el bosque aleatorio obtuvo un 88.2% y potenciación de gradiente un 90%, un punto importante a enmarcar es que con mayor cantidad de datos se incrementaron estos porcentajes para estas dos técnicas con un 90.4% y un 90.6% respectivamente, mientras que las demás técnicas disminuyeron su precisión. Aunque también es importante indicar que las otras técnicas no se encuentran muy por debajo de las dos mejores, la técnica con menor porcentaje de precisión, que son las redes neuronales artificiales con un valor de 87.3% difieren en menos de tres puntos porcentuales de la potenciación de gradiente. Para los casos de regresión logística y máquina de vector de soporte, se tienen el mismo porcentaje de 87.7%, dos puntos por debajo de la mejor técnica.

Un punto crucial de los resultados es que las técnicas entrenadas con datos reales otorgan muy pocos valores positivos, en los casos de redes neuronales artificiales, regresión

logística y máquinas de vector de soporte no hay ni uno solo, haciendo que la tarea de predicción sea una tarea muy complicada. Con los datos sintéticos se obtuvieron buena cantidad tanto de valores positivos como negativos, haciendo más sencilla la predicción de renuncia de socios.

Los atributos utilizados en esta investigación, así como el filtro de datos y todo lo involucrado en el preprocesamiento fueron altamente relevantes para llegar a estos resultados de precisión porque se obtuvo una alta precisión tanto con los datos reales de los socios como con los datos sintéticos.

Conclusiones

Finalizada la predicción de renuncia de socios de una cooperativa cerrada de la ciudad de Arequipa utilizando técnicas supervisadas de aprendizaje automático se llegó a las siguientes conclusiones:

1. Se consiguió la recolección de todos los datos relevantes de los socios de la cooperativa, se realizó un preprocesamiento de los datos y con ello, se pudo consolidar toda esta información en un archivo CSV para que sea menos complejo el proceso de analizar dichos datos.

2. Se realizó una investigación de técnicas supervisadas de predicción de datos de aprendizaje automático llegando seleccionar cinco de ellas que mejor se adecuaban al caso de estudio, y estas son redes neuronales, regresión logística, máquinas de vector de soporte, bosque aleatorio y potenciación de gradiente.

3. Se estableció que la técnica supervisada de aprendizaje automático con mayor precisión para la predicción de renuncia de socios es potenciación de gradiente, alcanzando un 90% de precisión con datos reales y con datos sintéticos alcanzó un 90.6%.

4. Se demostró que los resultados de aplicar estas técnicas de aprendizaje automático para la predicción de renuncia de socios son consistentes porque se realizó la aplicación de las mismas técnicas con datos sintéticos de mayor tamaño siendo los resultados muy similares, incluso se elevaron los resultados de precisión, tal es el

caso de bosque aleatorio y potenciación de gradiente. Cabe indicar que, con datos de mayor tamaño, la técnica de bosque aleatorio y potenciación de gradiente obtuvieron una mayor precisión, con datos reales la técnica de bosque aleatorio se logró un 88.2% y potenciación de gradiente un 90%, con los datos sintéticos con bosque aleatorio se logró un 88.9% y con potenciación de gradiente se alcanzó un 91.5%, en cambio las otras técnicas presentaron una reducción en la precisión, las redes neuronales artificiales de un 87.3% se redujeron a un 84.2%, regresión logística de un 87.7% a un 83.7% y máquinas de vector de soporte de un 87.7% a un 85.8%.

5. Se consiguió comprobar la efectividad de estas técnicas de aprendizaje automático para la predicción de renuncia de socios al poner a prueba estas técnicas con dos casos reales, el primero era de un socio que había renunciado a la cooperativa y el segundo de un socio que se mantuvo en ella. Se colocaron todos los datos de estos socios en las técnicas, los resultados con las técnicas entrenadas con datos reales no fueron acertadas, mientras que las técnicas entrenadas con datos sintéticos obtuvieron un resultado satisfactorio. Además, se implementó un prototipo web utilizando Amazon Web Services para realizar la validación, consistiendo de un formulario en donde el usuario puede colocar los datos de cualquier socio y verificar si este renunciará o no.

Recomendaciones y Trabajos Futuros

Como recomendaciones para un mejor manejo de la información por parte de los colaboradores de la cooperativa COOSUNAT, se propone utilizar una validación de los datos al momento en el que los socios se inscriben, ya que este proceso se realiza de forma manual por parte del socio y hay muchas ocasiones en los que los campos del formulario de inscripción se quedan en blanco.

Al momento de la generación de datos sintéticos se puede mejorar la calidad de los datos utilizando una previa validación en los parámetros de los datos reales para evitar que los valores de los datos sean muy distantes.

El tiempo que toma cada técnica supervisada para realizar el proceso de entrenamiento de la información es alto, y esto se incrementa más a medida que existan una mayor cantidad de datos. En promedio para cada técnica se tuvo un tiempo de proceso de entrenamiento entre 10 a 15 minutos, pero esto se elevó en gran medida cuando se hizo la aplicación de las técnicas con los datos sintéticos llevando a 40 minutos el tiempo de entrenamiento. Por lo que se propone para realizar estas técnicas, utilizar una computadora con una tarjeta gráfica NViDIA, como mínimo de la generación GTX, esta tarjeta gráfica permite trabajar en paralelo varios procesos y así reducir el tiempo de estos. Es importante aclarar que este proceso de entrenamiento solo se realiza una sola vez. Cuando el proceso concluye las respuestas de las predicciones de las técnicas se generan de forma inmediata.

La cooperativa desde el mes de agosto ha empezado a captar la CTS de los socios el cual, este es un atributo muy interesante y puede elevar la precisión de las técnicas. Por lo que, un trabajo futuro puede ser optimizar la precisión de estas técnicas mediante la inclusión de nuevos parámetros como el mencionado anteriormente.

Otro trabajo futuro es el desarrollo de un sistema web y/o móvil con el que el usuario, ya sea colaborador o el mismo gerente de la cooperativa, pueda no solo ver el resultado de la predicción de renuncia de socios como se vio en el prototipo sino que también muestre reportes de socios que estén con una mediana probabilidad de renuncia, reportes de aportes, préstamos, deudas, entre otros, todo con el fin de que el gerente pueda tomar mejores decisiones con la cooperativa.

Referencias Bibliográficas

Accenture (2017). Building Confidence Solving Banking's Cybersecurity Conundrum. Versión de 16 de febrero de 2017. Recuperado de https://www.accenture.com/t20170216T011141__w__/us-en///www.accenture.com/_acnmedia/PDF-44/Accenture-Building-Confidence-Solving-Banking-Cybersecurity-Conundrum.pdf#zoom=50

Adamchik V. (2009). Algorithmic Complexity. Recuperado de https://www.cs.cmu.edu/~adamchik/15-121/lectures/Algorithmic%20Complexity/complexity.html

Amazon (2019). Call an Amazon SageMaker model endpoint using Amazon API Gateway and AWS Lambda. Recuperado de https://aws.amazon.com/es/blogs/machine-learning/call-an-amazon-sagemaker-model-endpoint-using-amazon-api-gateway-and-aws-lambda/

Anaconda (2018). Anaconda. Recuperado de https://www.anaconda.com/about-us

Anaconda Cloud (2016). Anaconda Navigator. Recuperado de https://anaconda.org/anaconda/anaconda-navigator

Bernazzani, S. (2018). What Is Churn Rate? [Definition]. Recuperado de https://blog.hubspot.com/service/what-is-churn-rate

Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-solano, F., Caicedo, O. M. (2018): A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. Journal of Internet Services and Applications 9:16, DOI: 10.1186/s13174-018-0087-2

Brownlee, J. (2016). What is Deep Learning? Recuperado de https://machinelearningmastery.com/what-is-deep-learning/

Chanakya S. (2018). Logistic Regression. Recuperado de http://chanakya.ca/2018/05/23/logistic-regression/

Codecademy (2018). Integrated Development Environment. Recuperado de https://www.codecademy.com/articles/what-is-an-ide

Conduce tu Empresa (2019). ¿Qué es CAS? Derechos y obligaciones & Contrato Administrativo de Servicios. Recuperado de https://blog.conducetuempresa.com/2019/01/que-es-cas.html

Córdova J. (2016). COOSUNAT, Quienes Somos. Versión de 03 de julio de 2017. Recuperado de http://www.coosunat.org.pe/quienessomos.php

Córdova, E. (2017). Análisis predictivo de muerte y sobrevida de pacientes hospitalizados mediante clasificadores supervisados (Tesis). Recuperado de http://tesis.ucsm.edu.pe/repositorio/bitstream/handle/UCSM/7320/71.0604.IS.pdf

DeepAi (2017). Artificial Intelligence. Recuperado de https://deepai.org/machine-learning-glossary-and-terms/artificial-intelligence

Dormehl L. (2019). What is an artificial neural network? Recuperado de https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/

Dormehl, L. (2019). What is an artificial neural network? Here's everything you need to know. Recuperado de https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/

Faris H. (2018). A Hybrid Swarm Intelligent Neural Network Model for Customer Churn Prediction and Identifying the Influencing Factors. Information, 9, 288-305. DOI: 10.3390/info9110288

FENACREP (2017). ¿Qué es la FENACREP? Recuperado de https://www.fenacrep.org/1-10-nosotros

FENACREP (2017). ¿Qué es una COOPAC? Recuperado de https://www.fenacrep.org/2-18-coopac

FENACREP (2017). Principios Cooperativos. Recuperado de https://www.fenacrep.org/2-20-principios-cooperativos

FENACREP (2018). Reporte Corporativo. Versión de 31 de marzo de 2018. Recuperado de https://www.fenacrep.org/assets/media/imagenes/boletin-cooperativo/1-trimestre---2018---con-sangrado.pdf

Funahashi, H. y Horiuchi J. (2017). Characteristics of the churning process in continuous butter manufacture and modelling using an artificial neural network. International Diary Journal, 18, 323-328. doi:10.1016/j.idairyj.2007.08.001

Future of Life (2016). Benefits & Risks of Artificial Intelligence. Recuperado de https://futureoflife.org/background/benefits-risks-of-artificial-intelligence

Galleto M. (2016). What is Customer Churn? Recuperado de https://www.ngdata.com/what-is-customer-churn/

Gentiluomo L., Roessner D., Augustijn D., Svilenov H., Kulakova A., Mahapatra S., Winter G., Streicher W., Rinnan A., Peters G., Harris P. y Frieß W. (2019). Application of interpretable artificial neural networks to early monoclonal antibodies development. European Journal of Pharmaceutics and Biopharmaceutics, 141, 81-89. DOI: 10.1016/j.ejpb.2019.05.017

Guerra R. (2018). SUNAT: ¿Qué es?, ¿qué significa y cuáles son sus funciones? Recuperado de https://elcomercio.pe/economia/personal/sunat-significa-son-funciones-noticia-498862-noticia/

Gur (2019). Prototyping Model in Software Engineering: Methodology, Process, Approach. Recuperado de https://www.guru99.com/software-engineering-prototyping-model.html

Hong, K. (2016). (Batch) Gradient Descent Algorithm. Recuperado de https://www.bogotobogo.com/python/python_numpy_batch_gradient_descent_algorithm.php

Hong, K. (2016). Backpropagation of errors. Recuperado de https://bogotobogo.com/python/scikit-learn/Artificial-Neural-Network-ANN-4-Backpropagation.php

Iqbal, M.M., Saleem, Y. y Naseer, K. (2018): Multimedia based student-teacher smart interaction framework using multi-agents in eLearning. Multimedia Tools and Applications 77, 5003-5026. DOI: 10.1007/s11042-017-4615-z

Jupyter (2014). Jupyter Project and Community. Recuperado de https://jupyter.org/about

Karppi T. y Granata Y. (2019): Non-artificial non-intelligence: Amazon's Alexa and the frictions of AI. AI & SOCIETY 34, 867–876. DOI: 10.1007/s00146-019-00896-w

Kernel (2018). Computational complexity of machine learning algorithms. Recuperado de https://www.thekerneltrip.com/machine/learning/computational-complexity-learning-algorithms/

Khalid Awang, M., Ridwan Ismail, M., Makhtar, M., Nordin A Rahman, M., y Rasid Mamat, A. (2018). Performance Comparison of Neural Network Training Algorithms for Modeling Customer Churn Prediction. International Journal of Engineering & Technology, 7, 35-37. DOI: 10.14419/ijet.v7i2.15.11196

Kim S., Lee A. y Mun J. (2018). A Surrogate Modeling for Storm Surge Prediction Using an Artificial Neural Network. Journal of Coastal Research, 85, 866-870. DOI: 10.2112/SI85-174.1

Kumar, S. y Kumar M. (2019). Predicting Customer Churn Using Artifical Neural Network. Sptinger Reference Medizin, 1000, 299-306. DOI: 10.1007/978-3-030-20257-6_25

Le J. (2018). A Gentle Introduction to Neural Networks for Machine Learning. Recuperado de https://www.codementor.io/james_aka_yale/a-gentle-introduction-to-neural-networks-for-machine-learning-hkijvz7lp

López, J. y Pastor I. (2015). Bankruptcy visualization and prediction using neural networks: A study of U.S. commercial banks. ScienceDirect, 42(2015), 2857-2869.

Lorberfeld A. (2019). Machine Learning Algorithms In Layman's Terms, Part 1. Recuperado de https://towardsdatascience.com/machine-learning-algorithms-in-laymans-terms-part-1-d0368d769a7b

Lozano L. y Fernandezz J. (2016). Razonamiento Basado en Casos: Una Visión General. Versión de 10 de enero de 2016. Recuperado de https://www.infor.uva.es/~calonso/IAI/TrabajoAlumnos/Razonamiento%20basado%20en%20casos.pdf

Maleki H., Sorooshian A., Goudarzi G., Baboli Z., Birgani Y. y Rahmati M. (2019). Air pollution prediction by using an artificial neural network model. Clean Technologies and Environmental Policy, 21, 1341-1352. DOI: 10.1007/s10098-019-01709-w

MathWorks. (2018). What Is the Genetic Algorithm? Recuperado de https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html

Mindfire Solutions (2017). Advantages and Disadvantages of Python Programming Language. Recuperado de https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-python-programming-language-fd0b394f2121

Ministerio de la Producción. (2019). ¿Qué es una Cooperativa? Recuperado de https://www.produce.gob.pe/index.php/cooperativas/que-es-una-cooperativa

Pérez, I. (2018). Redes neuronales artificiales y sus aplicaciones en beneficio humano. Recuperado de http://ciencia.unam.mx/leer/773/redes-neuronales-artificiales-contribuyen-en-el-desarrollo-de-aplicaciones-en-beneficio-humano

Ping H., Stoyanovich J. y Howe B. (2017). DataSynthesizer: Privacy-Preserving Synthetic Datasets. Proceedings of the 29th International Conference on Scientific and Statistical Database Management-SSDBM 2017. DOI: 10.1145/3085504.3091117

Pizarro M. (2018). ¿Cuáles son los tipos de contrato laborales y sus beneficios? Recuperado de https://gestion.pe/economia/management-empleo/son-tipos-contrato-laborales-beneficios-238789-noticia/

Pycharm (2019). PyCharm. Recuperado de https://www.jetbrains.com/pycharm/features/

Python Contributors (2019). What is Python? Executive Summary. Recuperado de https://www.python.org/doc/essays/blurb/

Rouse M. (2018). AI (Artificial Intelligence). Recuperado de https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence

Rouse M. (2019). Artificial Neural Network. Recuperado de https://searchenterpriseai.techtarget.com/definition/neural-network

Rouse, M. (2018). Recurrent Neural Networks. Recuperado de https://searchenterpriseai.techtarget.com/definition/recurrent-neural-networks

Saishruthi S. (2018). Linear Regression — Detailed View. Recuperado de https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86

SAS (2017). Machine Learning. Recuperado de https://www.sas.com/en_us/insights/analytics/machine-learning.html

SBS (2018). ¿Quiénes somos? Recuperado de http://www.sbs.gob.pe/quienessomos

Scikit-learn (2019). Scikit-learn: Machine Learning in Python. Recuperado de https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Singh H. (2018). Understanding Gradient Boosting Machines. Recuperado de https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

Singh, A. (2017). Activation functions and it's types-Which is better? Recuperado de https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f

Soni D. (2018). Supervised vs. Unsupervised Learning. Recuperado de https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

Sulla, J. (2015). Aplicación de técnicas supervisadas de minería de datos para determinar la predicción de deserción académica (Tesis). Recuperado de http://tesis.ucsm.edu.pe/repositorio/bitstream/handle/UCSM/3612/K7.0252.SE.pdf

SuperDataScience Team (2018). Artificial Neural Networks - Gradient Descent. Recuperado de https://www.superdatascience.com/blogs/artificial-neural-networks-gradient-descent

Techopedia (2016). Labeled Data. Recuperado de https://www.techopedia.com/definition/33695/labeled-data

Vafeiadis T., Diamantaras K., Sarigiannidis G., Chatzisavvas K. (2015). A Comparison of Machine Learning Techniques for Customer Churn Prediction. Simulation Modelling Practice and Theory, 55, 1-9. DOI:10.1016/j.simpat.2015.03.003

Vasconcellos D., Artes R., Ayres F. y Fonseca A. (2017). Estimating credit and profit scoring of a Brazilian credit unión with logistic regression and machine-learning techniques. RAUSP Management Journal, 54, 321-336. DOI: 10.1108/RAUSP-03-2018-0003

World Council (2018). World Council of Credit Unios, 2017 Statistical Report. The global network of Credit Unions and financial cooperatives. Versión de noviembre de 2018. Recuperado de http://www.woccu.org/documents/2017_Statistical_Report-Revised_Nov_2018

World Council (2019). Why Credit Unions? Recuperado de https://www.woccu.org/about/credit_unions

Yadav A. (2018). Support Vector Machines (SVM). Recuperado de https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589

Yiu T. (2019). Understanding Random Forest. Recuperado de https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Zhou V. (2019). Machine Learning for Beginners: An Introduction to Neural Networks. Recuperado de https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-d49f22d238f9

Zolidah, K., Zaidah I. y Muhammad S. (2014). Customer Churn Prediction using Recurrent Neural Network with Reinforcement Learning Algorithm in Mobile Phone Users. International Journal of Intelligent Information Processing (IJIIP), Volume5, Number1.

Apéndices

Apéndice A: Plan de Tesis

1. PLANTEAMIENTO DEL PROBLEMA

1.1. Planteamiento del Problema

El Consejo Mundial de Cooperativas de Ahorro y Crédito o WOCCU por sus siglas en inglés (2019), indica que una cooperativa es una institución solidaria que es propiedad de los socios, quienes la controlan democráticamente y la operan, con el fin de maximizar el beneficio económico, proporcionando servicios financieros a precios competitivos y justos.

Según el informe estadístico de WOCCU (2018), a nivel mundial, las cooperativas son aproximadamente 89026, ubicadas en 117 países que atienden a 260 millones de socios, además de tener 1.7 billones de dólares en ahorros y aportaciones, 1.5 billones de dólares en préstamos y 2.1 billones de dólares en activos.

A pesar de estos grandes números, las cooperativas a nivel mundial lidian con varios problemas, entre ellos, la renuncia de socios, que es uno de los problemas más importantes que afectan en gran medida no solo a las cooperativas, sino a las instituciones financieras en todo el mundo. Según un estudio realizado por Accenture (2017), se encontró que los bancos y las cooperativas en Norteamérica pierden aproximadamente de 20 a 25% de nuevos socios en el primer año, y el costo promedio de adquirir un nuevo socio para las cooperativas es de 442 dólares, un costo elevado que podría conducir a grandes pérdidas para las cooperativas.

No obstante, en el Perú, un estudio realizado por la Federación Nacional de Cooperativas de Ahorro y Crédito del Perú o FENACREP (2018), indica que, en los últimos 5 años, el nivel de socios en las cooperativas creció en 45.27%, pero esa alza se ha estabilizado en el último año con un 0.36%. Además, el estudio indica que, Arequipa es la ciudad más baja en distribución de socios a nivel nacional con un 2.47%, que equivalen aproximadamente a 41920 socios.

Según Córdova (2016), la Cooperativa de Ahorro y Crédito de Trabajadores de la SUNAT “COOSUNAT”, es una cooperativa de tipo cerrada (solamente pueden ser socios trabajadores de la SUNAT), y que es de nivel 2 ya que su monto total de activos es mayor a 600 UIT. En esta cooperativa se ha visto que los socios están renunciando cada vez más a ser parte de ella. Las consecuencias que puede traer este problema son varias, de las cuales, la más crítica de todas, es que la cooperativa podría perder liquidez, conllevando a la quiebra de la misma.

Por lo tanto, la predicción de la renuncia de socios sería de gran utilidad para la cooperativa, ya que se determinaría las probabilidades, de acuerdo con información sociodemográfica histórica de los socios, si un socio va a renunciar a la cooperativa. Con estos datos, se podría analizar patrones, características, entre otros, que conllevan a la mayoría de los socios a desencantarse por la cooperativa y así, se podrían tomar mejores decisiones para poder reducir la tasa de renuncia de los socios.

1.2. Línea y Sublínea de Investigación

Línea de Investigación: Inteligencia Artificial.

Sublínea de Investigación: Algoritmos genéticos, redes neuronales y neuro evolución.

1.3. Palabras Clave

Red neuronal artificial, rotación de socios, cooperativa, aprendizaje automático.

2. OBJETIVOS DEL PROYECTO

2.1. General

Predecir la renuncia de socios de una cooperativa utilizando técnicas supervisadas de aprendizaje automático.

2.2. Específicos

- Reunir los datos de los socios que se encuentran distribuidos de forma desordenada en la base de datos de la cooperativa.
- Elegir las técnicas supervisadas de aprendizaje automático adecuadas para poder realizar el análisis de los socios de la cooperativa.
- Aplicar las técnicas de aprendizaje automático a la cooperativa.
- Verificar los resultados obtenidos con un conjunto de datos sintéticos generados a partir de los datos reales de la cooperativa con el fin de encontrar el número máximo de precisión.
- Realizar la validación de las técnicas utilizadas para comprobar la efectividad de las mismas.

3. FUNDAMENTOS TEÓRICOS

3.1. Estado del Arte

Las cooperativas en el Perú son muy importantes para el desarrollo económico, además de la inclusión financiera que logran porque llegan a los sectores más lejanos y vulnerables del país.

Según el Ministerio de Producción (2019), una cooperativa es una organización que agrupa a varias personas con la finalidad de realizar una actividad empresarial. Su modo de operar se basa en la cooperación de todos los socios. Todos "cooperan" para poder beneficiarse en forma directa para obtener un bien, un servicio o trabajo en mejores condiciones.

Partiendo de este concepto de cooperación, un grupo de trabajadores de la SUNAT fundaron la "Cooperativa de Ahorro y Crédito de Oficiales de la SUNAT" con nombre abreviado "COOSUNAT", donde cualquier trabajador de la SUNAT ya sea cesante, jubilado, sin distinción de régimen laboral o tipo de contrato puede inscribirse como socio.

La cooperativa a pesar de tener pocos años de haber sido fundada, ha crecido a pasos agigantados, ha logrado llegar a la categoría "Nivel II A" por tener un capital de más de dos millones y medio de soles.

Los importes de los socios a la cooperativa son descontados de sus sueldos brutos de la SUNAT, los socios eligen la cantidad que desean que se les descuente mediante una solicitud a mesa de partes de cualquier sucursal de la SUNAT.

Pero a pesar de todos los beneficios que la cooperativa ofrece, existen socios que no se adecúan o que no desean seguir aportando, entre otros motivos, derivando en la renuncia de éstos a la cooperativa, generando una tasa de pérdida de clientes.

Bernazzani (2018), indica que la tasa de pérdida de clientes es el porcentaje de los clientes o suscriptores de una empresa que cancelan o no renuevan su suscripción durante un periodo de tiempo dado. La tasa de pérdida es una medición critica para empresas cuyos clientes pagan de manera frecuente o aquellas basadas en suscripciones.

Hay diversos métodos para reducir la tasa de pérdida de clientes como:

- Comenzar con el pie derecho con el cliente.
- Solicitar feedback en momentos clave.
- Comunicarse activamente con el cliente.
- Analizar la tasa de pérdida cuando ocurra para mejorar el servicio al cliente.

Aparte de los métodos señalados por Bernazzani, en la actualidad existen técnicas para predecir si un cliente dejará o no una empresa en un tiempo determinado mediante

información histórica de éstos utilizando técnicas de aprendizaje automático como es el caso de las redes neuronales artificiales, todo con el objetivo de lograr un mejor análisis y reducir la tasa de pérdida de clientes.

Según Brownlee (2016), aprendizaje automático es un subcampo de aprendizaje automático relacionado con algoritmos inspirados en la estructura y funcionamiento del cerebro llamados redes neuronales artificiales.

La cooperativa tiene la necesidad de saber cuándo y debido a qué motivo un socio va a dejar la cooperativa, para satisfacer esta necesidad, se empleará un sistema inteligente que mediante redes neuronales artificiales se determinará el porcentaje de que un socio dejaré o no la cooperativa.

López y Pastor (2014) desarrollaron un modelo de redes neuronales para estudiar la quiebra de los bancos estadounidenses. Gracias al modelo los inversores, depositantes y otros participantes en los mercados de capital pueden evaluar el perfil de riesgo de su inversión. Aunque este modelo tiene sus limitaciones como la necesidad de muchos cálculos para la salida, visualización completa y el no poder controlar los factores macroeconómicos que puedan afectar la predisposición de los bancos a quebrar.

Un ejemplo de uso de redes neuronales artificiales para la predicción de datos es el de Funahashi y Horiuchi (2017), ellos necesitaban predecir el contenido de agua de la mantequilla y modelar las características del proceso de batido a partir de una red neuronal artificial. Gracias al uso de este modelo, se pudo realizar un gran análisis, concluyendo que el control de la temperatura de alimentación de la crema es muy importante en la fabricación de mantequilla.

Vafeiadis, Diamantaras, Sarigiannidis, Chatzisavvas (2015), realizaron una comparación de técnicas de aprendizaje automático para predecir la renuncia de clientes en la industria de telecomunicaciones. Los métodos con mejor rendimiento fueron la Red de Propagación hacia atrás y el Árbol de Decisión, ambos métodos lograron una precisión de 94% y 77% respectivamente.

Faris (2018), realizó un modelo híbrido basado en optimización de enjambre de partículas y en una red neuronal de la industria de telecomunicaciones para predecir la tasa de renuncia de clientes. La optimización de enjambre de partículas fue utilizada para mejorar los pesos de las variables de entrada y optimizar la estructura de la red neuronal simultáneamente para incrementar el poder de la precisión. Se basaron en dos conjuntos de datos de dos compañías de telecomunicaciones obteniendo como resultado que el modelo propuesto puede incrementar significativamente la tasa de cobertura de la rotación de clientes en comparación a otros clasificadores, además indica que la automática optimización de la red neuronal elimino el esfuerzo que se necesitaba para obtener el mejor número de capas ocultas de la red neuronal.

Khalid, Ridwan, Makhtar, Nordin, y Rasid (2018), realizaron una comparación de algoritmos de redes neuronales para la predicción de rotación de clientes para una

compañía de telecomunicaciones de Malasia. Los algoritmos comparados fueron Propagación posterior de Levenberg Marquardt, retro propagación BFGS Quasi-Newton, propagación posterior de gradiente conjugado con actualizaciones Fletcher-Reeves. Su análisis mostró que la red neuronal entrenada con el algoritmo Levenberg Marquart obtuvo la mayor precisión con un 94.82%, además concluyeron que todas los predictores comparados son aceptables para la predicción de tasa renuncia de clientes. El modelo óptimo de red neuronal para los autores consiste de catorce variables de entrada, un nodo oculto y una variable de salida con el algoritmo de Levenberg Marquart.

Kim, Lee y Mun (2018), desarrollaron un modelo para predecir las mareas de tormenta en Corea del Sur utilizando una Red Neuronal Artificial, para ello se emplearon datos históricos de 59 tormentas que sucedieron entre 1978 y 2014 en dicho país. Las variables de entrada fueron longitud, velocidad de movimiento, latitud, dirección de rumbo, presión central, radio de velocidad del viento y velocidad máxima del viento. Para medir el rendimiento de este modelo se expresó como el coeficiente de correlación, los coeficientes máximos y mínimos fueron 0.861 y 0.979 respectivamente.

Kumar S. y Kumar M. (2019), mediante redes neuronales artificiales y con diferentes funciones de activación realizaron la predicción de la tasa de rotación de clientes de un conjunto de datos de una empresa de telecomunicaciones para determinar los factores que influyen en los clientes para su renuncia, logrando una alta precisión de más del 80%.

Maleki, Sorooshian, Goudarzi, Baboli, Birgani y Rahmati (2019), realizaron un estudio para evaluar la efectividad de una Red Neuronal Artificial para predecir las concentraciones de contaminantes atmosféricos por hora y dos índices de calidad del aire que son Índice de Calidad del Aire (AQI) e Índice de Salud de la Calidad del Aire (AQHI) en la ciudad de Ahvaz, Irán. Determinaron que los valores de coeficiente de correlación y el error cuadrático medio fueron 0.87 y 59.9 respectivamente. Además, concluyeron que la aplicación de una Red Neuronal Artificial es factible para ciudades como Ahvaz para pronosticar la calidad del aire con la finalidad de prevenir los efectos en la salud.

Gentiluomo, Roessner, Augustijn, Svilenov, Kulakova, Mahapatra, Winter, Streicher, Rinnanc, Peters, Harris y Frieß (2019), utilizaron Redes Neuronales Artificiales para predecir las propiedades biofísicas de los anticuerpos monoclonales terapéuticos (temperatura de fusión, temperatura de inicio de agregación, parámetro de interacción y la concentración de sal de la composición de aminoácidos). Al solo usar la composición de aminoácidos mantuvieron Redes Neuronales Artificiales simples, permitiendo una alta aplicabilidad general, robustez e interpretabilidad. Los autores obtuvieron 0.94 como resultados de coeficiente de correlación y alrededor de 20 fue el error cuadrático obtenido.

Un ejemplo de uso de redes neuronales recurrentes en la predicción de rotación de clientes fue el realizado por Zolidah, Zaidah y Syahir (2014), ellos implementaron una red neuronal recurrente de Elman y una red neuronal recurrente de Jordan con aprendizaje de refuerzo para predecir la tasa de rotación de los usuarios de celulares. El proyecto pudo demostrar que la red neuronal recurrente de Jordan proporciona una mejor precisión que la red neuronal recurrente de Elman.

En el ámbito de cooperativas, Vasnconcellos, Arte, Ayres y Fonseca (2019), analizaron la puntuación de créditos de una cooperativa en Brasil utilizando el método de regresión logística y bosques aleatorios que son técnicas de aprendizaje automático, concluyendo que la técnica de bosque aleatorio funciona mejor que el método de regresión logística para la puntuación crediticia.

A nivel nacional, Sulla (2015), utilizó técnicas supervisadas de minería de datos para poder predecir la deserción de estudiantes de una universidad, para ello utilizo varias técnicas supervisadas de minería de datos árboles de decisión, redes neuronales, redes bayesianas, entre otras. Llegando a la conclusión de que los estudiantes que abandonan la universidad tienen solo 11 cursos aprobados y un promedio final de tan solo 7.84.

Córdova (2017), realizó un sistema de predicción que tenía el objetivo de predecir la muerte y sobrevida de pacientes del hospital Honorio Delgado de Arequipa mediante técnicas de redes neuronales de tipo Backpropagation, clasificadores bayesianos y máquinas de vectores de soporte.

Dada la investigación realizada se puede concluir que la predicción de datos facilita mucho la toma de decisiones por parte de las organizaciones, gracias al uso de las redes neuronales artificiales se pueden llegar a obtener resultados precisos y confiables, y no solo esto, también se puede hallar patrones complejos en los datos que a simple vista es muy difícil de detectar, logrando un análisis más completo.

Este proyecto estará enfocado en desarrollar e implementar un sistema inteligente a partir de un modelo de red neuronal artificial con la finalidad de predecir si un socio de la cooperativa COOSUNAT renunciará o no en un futuro. Se logrará facilitar la toma de decisiones de la gerencia y mejorar el servicio que tiene la cooperativa con los socios.

3.2. Bases Teóricas del Proyecto

3.2.1. Redes Neuronales Artificiales

Dormehl (2019), indica que las redes neuronales artificiales son sistemas inspirados en cómo funciona el cerebro, pretendiendo replicar la manera en que los humanos aprenden. Las redes neuronales artificiales consisten en capas de entrada, capas oculta y capas de salida, son herramientas muy buenas para encontrar patrones muy grandes o muy complejos para que un humano pueda extraerlos y hacer aprender a la máquina a reconocerlos.

3.2.2. Función de Activación

Singh (2017), indica que las funciones de activación son muy importantes para que las redes neuronales artificiales puedan aprender y se pueda dar más sentido a cosas muy complicadas, además de introducir propiedades no lineales a la red neuronal artificial. El objetivo principal de una función de activación es convertir una señal de una neurona de entrada a una señal de salida.

3.2.3. Descenso de Gradiente

Hong (2016), indica que el descenso de gradiente es un algoritmo de optimización que funciona mediante la búsqueda eficiente de parámetros, la intersección y la pendiente para la regresión lineal.

3.2.4. Propagación hacia atrás

Hong (2016), indica que Backpropagation es un algoritmo utilizado para entrenar las redes neuronales artificiales, pudiendo actualizar los pesos eficientemente. Usualmente es usado con método de optimización de descenso de gradiente. Eremenko (2018), indica que básicamente, Backpropagation ocurre cuando se retroalimentan los datos finales a través de la red neuronal y luego se ajustan las sinapsis ponderadas entre el valor de entrada y la neurona, al repetir este ciclo y ajustar los pesos, se reduce la función de costo.

3.2.5. Razonamiento basado en casos

Lozano y Fernández (2016), indican que el razonamiento basado en caso es un paradigma de resolución de problemas capaz de usar conocimiento de experiencias previas en concreto, además, el razonamiento basado en casos provee un acercamiento al aprendizaje incremental porque almacena una experiencia nueva cada vez que un problema es resuelto.

3.2.6. Algoritmos Genéticos

Mathworks (2018), indica que un al algoritmo genético es un método para resolver problemas de optimización restringidos y no restringidos basados en la selección natural, el proceso que impulsa la evolución biológica. El algoritmo genético modifica repetidamente una población de soluciones individuales.

4. PRESENTACIÓN DEL PROYECTO

4.1. Justificación

Las cooperativas son instituciones que se deben fundamentalmente a sus socios, sin sus aportes periódicos o si se reduce el número de aportantes sería casi imposible su funcionamiento, ya que al tener menos aportes significaría tener menos liquidez poniéndolos en una situación financiera crítica, por lo tanto, maximizar la liquidez es un objetivo fundamental de la cooperativa por lo que reducir la tasa de renuncia de socios es muy importante.

Con las técnicas se aprendizaje automático se podrá saber cuál es el porcentaje que tiene un socio de renunciar a la cooperativa, asimismo, cuáles son los patrones y características de los socios renunciantes. Con esta información, el gerente general podrá visualizar y analizar las tendencias de los socios que podrían renunciar en un futuro cercano y con ello tomar mejores decisiones respecto al accionar de la cooperativa como por ejemplo: las tasas, la cantidad de cuotas a pagar, el crédito que se puede otorgar, así como también, como accionar si el asociado es de contrato CAS o no, entre otros), todo con el fin de mantener a los socios en la cooperativa y así evitar que la cooperativa pierda liquidez.

Se aplicarán técnicas de aprendizaje automático para un caso de la vida real, se verá a detalle cada paso a realizar para la aplicación de esta técnica hasta los resultados obtenidos y su posterior análisis, marcando un precedente de utilización de esta técnica para la predicción de datos, que puede ser de sumo provecho para futuras investigaciones que pueden realizarse de la aplicación de estas técnicas.

4.2. Resumen del Proyecto

4.2.1. Descripción del Proyecto a Medio y Largo Plazo

El proyecto se basa en analizar y clasificar los datos de todos los socios inscritos en la cooperativa desde los inicios de esta, hasta la actualidad; para finalmente, lograr la predicción de que, si un socio se mantendrá o no en la cooperativa; una vez finalizada esta tarea, se pretende visualizar los resultados obtenidos y analizar el nivel de confianza de la clasificación para que de esta forma se puedan realizar las mejoras correspondientes.

4.2.2. Usuarios del Proyecto

- Consejo Administrativo
- Área de Gerencia
- Área de Contabilidad

4.2.3. Beneficios

A. Detección de principales problemas que ocasionan la renuncia de los socios.

B. Reducción de rotación de socios de la cooperativa.

C. Ordenamiento de los datos de los socios de la cooperativa, ya que estos se encuentran dispersos, algunos no están normalizados y en otros casos son incongruentes.

D. Crear un importante valor añadido para la cooperativa.

E. Mejorar la toma de decisiones para reducir la renuncia de los socios de la cooperativa.

4.2.4. Localización

El sistema inteligente estará ubicado en el servidor del área de gerencia.

4.2.5. Análisis del Futuro del Proyecto

Una vez finalizada la predicción de los socios de la cooperativa se plantea continuar realizando mejoras al algoritmo de clasificación ya que se puede lograr incrementar el porcentaje de la confianza de este.

4.2.6. Riesgos que debemos afrontar

A. Un riesgo para tomar en cuenta es no lograr un alto porcentaje de confiabilidad en el algoritmo de clasificación.

B. Los patrones detectados pueden ser difíciles de entender.

5. PLAN DE IMPLANTACIÓN DEL PROYECTO

5.1. Definición del Proyecto

5.1.1. Aspectos Técnicos

- El lenguaje de programación a utilizar será Python por su simplicidad al usar y porque posee varios paquetes de aprendizaje automático para el análisis de datos.

- Para la recolección y limpieza de datos se utilizará los macros de Excel por las ventajas que traen como reducir la tasa de errores y reducir el tiempo de trabajo ya que la información de los socios de la cooperativa se encuentra en hojas de cálculo de Excel y en varios casos en una sola hoja están los registros de todos los meses desde iniciada la cooperativa, es decir, más de once mil registros.

- Se utilizarán Tensorflow, Pytorch y Keras porque son librerías especializadas de aprendizaje automático para Python y son de código abierto.

- Se utilizará el IDE Pycharm de la empresa Jetbrains por su facilidad de uso, también porque da una visualización con detalle de cada variable, proceso, errores al momento de la codificación.

- Se utilizará Scikit-learn, que es una librería de aprendizaje automático principalmente para mejorar los modelos con un ajuste de parámetros efectivo y lo más importante, para preprocesar los datos, para que los modelos puedan aprender en las mejores condiciones.

- Se usarán las librerías: Numpy para realizar cálculos elevados y manipular arreglos de alta dimensión, Matplotlib para trazar gráficos intuitivos y Pandas para importar y manipular conjuntos de datos de la manera más eficiente.

5.1.2. Aspectos Económicos

Para realizar este proyecto se usará software (Pentaho), lenguaje de programación (Python) y librerías de código abierto, y el IDE Pycharm que su modo de prueba por lo que el costo en licencias será mínimo.

Costos	
Tiempo de programación	800.00
Servicio de Luz	50.00
Internet	80.00
Papelería	3.00

Movilidad	20.00

Ingresos	
Sueldo Neto	1000.00
Por c/socio que se puede mantener en la cooperativa gracias al sistema	5.00

Costo/Beneficio = 1000/953 = 1.05

5.1.3. Aspectos Comerciales

Se podría hacer una patente con el software y la metodología propuesta.

6. METODOLOGÍA A EMPLEAR

Paso	**Resultados Esperados**
Reunir los datos de los socios que se encuentran distribuidos de manera desordenada en la base de datos de la cooperativa.	Datos de los socios consolidados en un archivo de tipo csv (valores separados por coma).
Elegir técnicas supervisadas de aprendizaje automático para el análisis.	Cuadro de un conjunto de técnicas supervisadas de aprendizaje automático a partir de la realización de un análisis exhaustivo de rendimiento y precisión de cada técnica.
Aplicar las técnicas de aprendizaje automático a la cooperativa.	Listado de resultados de la precisión de las técnicas y el porcentaje hallado que tienen los socios de renunciar a la cooperativa.
Verificar los resultados obtenidos a partir de datos reales con un conjunto de datos sintéticos para hallar la máxima precisión.	Cuadro de resultados en donde se tiene la comparación de las precisiones de los datos reales con el conjunto de datos sintéticos.
Validar las técnicas utilizadas para comprobar su efectividad.	Cuadro comparativo de la aplicación y nivel de acierto de las técnicas utilizadas con los datos reales de los socios de la cooperativa.

7. REFERENCIAS

Accenture. (2017). Building Confidence Solving Banking's Cybersecurity Conundrum. Versión de 16 de febrero de 2017. Recuperado de https://www.accenture.com/t20170216T011141__w__/us-en///www.accenture.com/_acnmedia/PDF-44/Accenture-Building-Confidence-Solving-Banking-Cybersecurity-Conundrum.pdf#zoom=50

Bernazzani, S. (2018). What Is Churn Rate? [Definition]. Recuperado de https://blog.hubspot.com/service/what-is-churn-rate

Brownlee, J. (2016). What is Deep Learning? Recuperado de https://machinelearningmastery.com/what-is-deep-learning/

Córdova J. (2016). COOSUNAT, Quienes Somos. Versión de 03 de julio de 2017. Recuperado de http://www.coosunat.org.pe/quienessomos.php

Córdova, E. (2017). Análisis predictivo de muerte y sobrevida de pacientes hospitalizados mediante clasificadores supervisados (Tesis). Recuperado de http://tesis.ucsm.edu.pe/repositorio/bitstream/handle/UCSM/7320/71.0604.IS.pdf

Dormehl, L. (2019). What is an artificial neural network? Here's everything you need to know. Recuperado de https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/

Faris H. (2018). A Hybrid Swarm Intelligent Neural Network Model for Customer Churn Prediction and Identifying the Influencing Factors. Information, 9, 288-305. DOI: 10.3390/info9110288

FENACREP (2018). Reporte Corporativo. Versión de 31 de marzo de 2018. Recuperado de https://www.fenacrep.org/assets/media/imagenes/boletin-cooperativo/1-trimestre---2018---con-sangrado.pdf

Funahashi, H. y Horiuchi J. (2017). Characteristics of the churning process in continuous butter manufacture and modelling using an artificial neural network. International Diary Journal, 18, 323-328. doi:10.1016/j.idairyj.2007.08.001

Gentiluomo L., Roessner D., Augustijn D., Svilenov H., Kulakova A., Mahapatra S., Winter G., Streicher W., Rinnan A., Peters G., Harris P. y Frieß W. (2019). Application of interpretable artificial neural networks to early monoclonal antibodies development. European Journal of Pharmaceutics and Biopharmaceutics, 141, 81-89. DOI: 10.1016/j.ejpb.2019.05.017

Hong, K. (2016). (Batch) Gradient Descent Algorithm. Recuperado de https://www.bogotobogo.com/python/python_numpy_batch_gradient_descent_algorithm.php

Hong, K. (2016). Backpropagation of errors. Recuperado de https://bogotobogo.com/python/scikit-learn/Artificial-Neural-Network-ANN-4-Backpropagation.php

Khalid Awang, M., Ridwan Ismail, M., Makhtar, M., Nordin A Rahman, M., y Rasid Mamat, A. (2018). Performance Comparison of Neural Network Training Algorithms for Modeling Customer Churn Prediction. International Journal of Engineering & Technology, 7, 35-37. DOI: 10.14419/ijet.v7i2.15.11196

Kim S., Lee A. y Mun J. (2018). A Surrogate Modeling for Storm Surge Prediction Using an Artificial Neural Network. Journal of Coastal Research, 85, 866-870. DOI: 10.2112/SI85-174.1

Kumar, S. y Kumar M. (2019). Predicting Customer Churn Using Artifical Neural Network. Sptinger Reference Medizin, 1000, 299-306. DOI: 10.1007/978-3-030-20257-6_25

López, J. y Pastor I. (2015). Bankruptcy visualization and prediction using neural networks: A study of U.S. commercial banks. ScienceDirect, 42(2015), 2857-2869.

Lozano L. y Fernandezz J. (2016). Razonamiento Basado en Casos: Una Visión General. Versión de 10 de enero de 2016. Recuperado de https://www.infor.uva.es/~calonso/IAI/TrabajoAlumnos/Razonamiento%20basado%20en%20casos.pdf

Maleki H., Sorooshian A., Goudarzi G., Baboli Z., Birgani Y. y Rahmati M. (2019). Air pollution prediction by using an artificial neural network model. Clean Technologies and Environmental Policy, 21, 1341-1352. DOI: 10.1007/s10098-019-01709-w

MathWorks. (2018). What Is the Genetic Algorithm? Recuperado de https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html

Ministerio de la Producción. (2019). ¿Qué es una Cooperativa? Recuperado de https://www.produce.gob.pe/index.php/cooperativas/que-es-una-cooperativa

Pérez, I. (2018). Redes neuronales artificiales y sus aplicaciones en beneficio humano. Recuperado de http://ciencia.unam.mx/leer/773/redes-neuronales-artificiales-contribuyen-en-el-desarrollo-de-aplicaciones-en-beneficio-humano

Rouse, M. (2018). Recurrent Neural Networks. Recuperado de https://searchenterpriseai.techtarget.com/definition/recurrent-neural-networks

Singh, A. (2017). Activation functions and it's types-Which is better? Recuperado de https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f

Sulla, J. (2015). Aplicación de técnicas supervisadas de minería de datos para determinar la predicción de deserción académica (Tesis). Recuperado de http://tesis.ucsm.edu.pe/repositorio/bitstream/handle/UCSM/3612/K7.0252.SE.pdf

SuperDataScience Team (2018). Artificial Neural Networks - Gradient Descent. Recuperado de https://www.superdatascience.com/blogs/artificial-neural-networks-gradient-descent

Vafeiadis T., Diamantaras K., Sarigiannidis G., Chatzisavvas K. (2015). A Comparison of Machine Learning Techniques for Customer Churn Prediction. Simulation Modelling Practice and Theory, 55, 1-9. DOI:10.1016/j.simpat.2015.03.003

Vasconcellos D., Artes R., Ayres F. y Fonseca A. (2017). Estimating credit and profit scoring of a Brazilian credit unión with logistic regression and machine-learning techniques. RAUSP Management Journal, 54, 321-336. doi: 10.1108/RAUSP-03-2018-0003

World Council (2019). Why Credit Unions?. Recuperado de https://www.woccu.org/about/credit_unions

World Council. (2018). World Council of Credit Unios, 2017 Statistical Report. The global network of Credit Unions and financial cooperatives. Versión de noviembre de 2018. Recuperado de http://www.woccu.org/documents/2017_Statistical_Report-Revised_Nov_2018

Zolidah, K., Zaidah I. y Muhammad S. (2014). Customer Churn Prediction using Recurrent Neural Network with Reinforcement Learning Algorithm in Mobile Phone Users. International Journal of Intelligent Information Processing (IJIIP), Volume5, Number1.

Apéndice B: Ficha de inscripción de Socio de COOSUNAT

COOPERATIVA DE AHORRO Y CRÉDITO DE TRABAJADORES DE LA SUNAT

RUC 20600279484

FICHA DE INSCRIPCIÓN

COOSUNAT

Nombres y Apellidos:	
Reg. SUNAT y DNI:	
Dirección actual completa, incl. distrito, provincia y dpto.:	
Estado civil:	
Nombres de dependientes a su cargo:	
Teléfonos (celulares y fijo):	
Correo electrónico de SUNAT:	
Correo electrónico personal:	
Nombre en Facebook:	
Lugar de nacimiento:	
Fecha de nacimiento y edad:	
Sede de trabajo en SUNAT:	
Código de unidad organizacional:	
Cargo en la SUNAT:	
Anexo de oficina en SUNAT:	
Fecha de ingreso a SUNAT o Ex Aduanas:	
Banco y cuenta de haberes:	
Profesión (si tuviese):	
Grado Académico (si tuviese):	
Régimen Laboral actual:	
Ciudad y fecha:	

INCLUYO COPIA DE MI DNI.

FIRMA