Facultad de Ingenierías

MODELO PARA LA DETECCIÓN DE RIESGO DE CRÉDITO EN ENTIDADES FINANCIERAS BAJO UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL

ANDRÉS FELIPE BRAVO GIRALDO

DIEGO ALEJANDRO RESTREPO SANCHEZ

Prof. JULIO CÉSAR LÓPEZ BETANCUR

TRABAJO DE GRADO

PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PEREIRA, RISARALDA ENERO 9 DE 2020

Contenido

1. Introducción

1.1. Presentación del proyecto

En la actualidad el ambiente económico se encuentra en una etapa de cambio cubierto por una ola de revolución tecnológica donde el uso de los datos y la información son clave a la hora de tomar decisiones trascendentales que eventualmente al darle buen manejo son positivas para cualquier organización; cabe resaltar que existen infinidades de compañías donde actualmente se producen datos cada día, pero estos datos no se convierten en información cuyo valor suele ser incalculable para las mismas. Por lo que para ello se pretende implementar un modelo de predicción dónde utilizando información histórica generada a través de los usuarios de una entidad del sector solidario que sirve para evaluar la pérdida esperada en entidades financieras utilizando técnicas de inteligencia artificial y así evaluar el nivel de riesgo de crédito que representa una persona natural o jurídica dándole así valor a los datos recopilados previamente.

El caso de las entidades del sector solidario como las Cooperativas de ahorro y crédito para las cuales su principal actividad consta de ofrecer servicios financieros a los asociados de estas, actividad que las incluye en un sector con una alta supervisión por los entes de control como lo son la Superintendencia Financiera y la superintendencia de Economía Solidaria debido a que contempla actividades económicas con fines lucrativos.

1.2. Planteamiento del problema

Cuando se habla de otorgar créditos a una persona en el sector financiero implícitamente se asume un riesgo que latente a la hora de realizar o adjudicar un crédito cualquier individuo y así mismo generar un aumento imprevisible de deudas incobrables generando así una disminución de las utilidades obtenidas por las entidades. Para ello algunas entidades han optado por diseñar una especie de sistema que pretende analizar diversos factores económicos y sociales del individuo en cuestión y determinar el riesgo que representa ampliar la cartera de créditos adjudicando uno a una persona.

Las entidades anteriormente mencionadas realizan estudios de crédito con base en información otorgada por un panel de expertos, sin embargo, este tipo de estudio está sujeto al error humano. La empresa del Sector Solidario Cooperativa FAVI viene utilizando una herramienta llamada scoring elaborada en un libro de Excel y que fue inspeccionada bajo el juicio de expertos, el cual por medio de información de la persona que solicita un crédito y al realizar su estudio crediticio entrega como resultado un valor que califica la pérdida esperada por la entidad en relación a la persona que se encuentra en estudio estableciendo así un parámetro medible que se determina como riesgo de crédito. El scoring ha trabajado de una forma aceptable con los pesos asignados por personas diestras en el ámbito financiero y con la información que exige la normativa legal vigente, sin embargo, sigue existiendo un factor humano ligado a errores que puede ser corregible mediante el uso de técnicas de inteligencia artificial.

1.3. Objetivos

1.3.1. Objetivo general

Implementar un modelo aplicando técnicas de inteligencia artificial que permita evaluar el nivel de riesgo crediticio para la apertura de un crédito a una persona natural o jurídica.

1.3.2. Objetivos específicos

- Consultar sobre técnicas y trabajos de inteligencia artificial para utilizar en la creación de modelo.
- Definir las variables a evaluar en el modelo de inteligencia artificial.
- Creación de modelos prototipos de inteligencia artificial a evaluar.
- Evaluación de modelos de inteligencia artificial.
- Selección de modelo de inteligencia artificial.
- Entrenamiento de modelo de Inteligencia artificial con los datos obtenidos por la entidad financiera FAVI.
- Evaluar resultados obtenidos a partir de otros datos, los cuales son diferentes al de entrenamiento.

1.4. Justificación

Se pretende crear una herramienta que permita identificar el riesgo crediticio a considerar por una entidad financiera cuando un individuo solicita un crédito; esta herramienta se llevará a cabo utilizando técnicas de inteligencia artificial debido a que las herramientas construidas a juicio de expertos están sujetas al error humano y a análisis subjetivos.

1.5. Límites

Esta investigación contempla la implementación de un algoritmo cuyo contenido involucra técnicas de inteligencia artificial cuyo propósito se centra en clasificar el nivel de riesgo que representa una persona para una entidad financiera al momento de tomar un crédito en 7 categorías diferentes, dicho algoritmo se alimenta una data de información histórica de la cartera y clientes que fue otorgada por la entidad financiera del sector solidario FAVI.

El proceso investigativo se lleva a cabo en las instalaciones de la Universidad Tecnológica de Pereira en apoyo del docente guía Julio César López Betancur en el periodo de tiempo que comprende segundo semestre académico del año 2019.

1.6. Metodología

1.6.1. Contenidos

Con el fin de obtener resultados fehacientes y realizar un buen manejo de los datos otorgados para el análisis y posterior implementación de algoritmos que contemplan técnicas de inteligencia artificial. Se procede a utilizar herramientas de desarrollo como lo son Jupyter con librerías de Machine Learning contempladas a continuación:

- Tensor Flow
- Keras
- PyTorch
- NumPy
- Scipy
- NLTK

- Lasagne
- Pyrenn
- JavaScript
- H5py

Se selecciona el lenguaje de programación Python en su versión 3.7 para dar solución a la implementación bajo una metodología ágil de desarrollo como lo es Scrum.

Se tomó la decisión de crear algoritmos bajo los parámetros de algunas Técnicas de Inteligencia Artificial para obtener mejores resultados a partir de los datos históricos y mitigar el error humano, ya que es difícil procesar una elevada cantidad datos históricos para escoger los valores adecuados dentro del scoring. Se crearon 4 algoritmos que implementan redes neuronales, máquinas de vector de soporte, algoritmos genéticos y cadenas de Márkov.

De los 4 algoritmos realizados encontramos que la red neuronal es el que obtiene mejores resultados, durante el proceso de la creación de la red neuronal utilizamos diferentes cantidades de neuronas, las cuales fueron 25, 50, 100, 200.

1.6.2. Diseño de técnicas de recolección de información

En colaboración con el docente guía y la entidad FAVI UTP se procede a solicitar información histórica de clientes comprendida en los periodos 2017, 2018 y 2019 de las cuales se procede a seleccionar los clientes que tienen o han tenido cartera de crédito con la entidad y que este ha estado activo como mínimo 6 meses.

1.6.3. Universo

El universo considerado para este proceso investigativo considera a todas las entidades financieras del sector solidario presentes en el territorio nacional.

1.6.4. Población y muestra

Para el caso concreto del proyecto de investigación se procede a tomar una data histórica de 3 años que están comprendidos entre los años 2017, 2018 y 2019 de los clientes asociados a la cooperativa FAVI contemplando así 1092 registros de usuarios asociados con características de asociados, no asociados y retirados, posterior a ello se observan los créditos otorgados a cada una de las personas asociadas y el comportamiento en el transcurso del compromiso adquirido con la entidad.

1.6.5. Delimitación del estudio

El estudio está delimitado por las siguientes variables:

- Espacial: La entidad cooperativa del sector solidario FAVI UTP y sus asociados.
- Demográfica: Una población de 1600 asociados
- Temporal: Se tomará el comportamiento histórico de los años 2017, 2018 y 2019.
- Temática: Medición de riesgos financieros – riesgo de crédito.

1.6.6. Técnicas de análisis

- Análisis de escenarios: Es una técnica que consiste en analizar una variedad de datos o eventos futuros y se utiliza cuando no se está seguro sobre qué decisión tomar dado un suceso determinado.

- Redes neuronales: Una red neuronal simula el comportamiento cerebral del ser humano en el proceso para la toma de decisiones en inteligencias artificiales.

2. Marco teórico

2.1. Técnicas de Inteligencia Artificial para la detección de riesgo de crédito en entidades financieras

Para hablar de inteligencia artificial primero se deben conocer el comportamiento de cualquier ser vivo cuya inteligencia es alimentada por información externa recibida a través de los sentidos y que está ligada a la experiencia de acuerdo a sus necesidades ya que permiten mejorar la toma de decisiones al momento de evaluar diversas situaciones para garantizar su supervivencia, así como cualquier ser inteligente, el ser humano recibe cualquier tipo de información y con base en ella se razona, aprende y se resuelven problemas. Para algunos autores la inteligencia artificial la observan como un sistema que actúa, mientras otros la ven como el desarrollo de sistemas capaces de razonar o pensar. Hay quienes asocian la inteligencia artificial con el ser humano (asumiendo que es inteligente), es por esta razón que algunos han propuesto cambiarle el nombre a la Inteligencia Artificial para evitar la mala imagen que está pueda ocasionar Fernando Berzal (2018).

En una era en la que la tecnología es un gran aliado para el ser humano como herramienta para analizar y eventualmente tomar decisiones, el avance en el uso de técnicas que permitan imitar el comportamiento humano es indispensable para mejorar la productividad o mitigar los

riesgos en la toma de decisiones Poole, David (2018). Llegar a una definición de lo que es la Inteligencia Artificial es prácticamente imposible, ya que todos no van a estar de acuerdo. “El estudio de cómo hacer que los ordenadores hagan cosas, que los humanos hacen mejor” Elaine Rich (1983).

2.2. Algoritmos genéticos

2.2.1. Definición

Los algoritmos genéticos son una técnica de búsqueda basada en la teoría de la evolución biológica de los seres vivos que busca replicar el comportamiento evolutivo de una especie inspirado en el hecho de transmitir los mejores componentes genético-moleculares de una generación a otra partiendo de que cada generación de humanos es el resultado de la selección del mejor material genético de los progenitores J. H. Holland. University of Michigan Press, Ann Arbor. (1975) D. E. Goldberg y Addison-Wesley, Longman Publishing Co (1989). En la década de los años 70’s el PhD ingeniero John Henry Holland dio origen a una de las ramas de la inteligencia artificial conocida como algoritmos genéticos.

Los algoritmos evolutivos son estrategias matemáticas, métodos de optimización que buscan soluciones partiendo de la evolución biológica. Los algoritmos genéticos están diseñados para mantener un conjunto de diversas entidades en las cuales se albergan posibles soluciones realizando una especie de competencia entre ellas y otorgando la mejor solución a la siguiente generación garantizando así una mejor solución cada vez que se ejecuta Bäck, T. (1996).

La computación evolutiva se destaca como una de las ramas de inteligencia artificial que involucra problemas de optimización combinatoria inspirada en mecanismos de evolución

biológica, siguiendo los principios de la teoría de la evolución propuesta por el naturalista británico Dr. Charles Robert Darwin en el año de 1859.

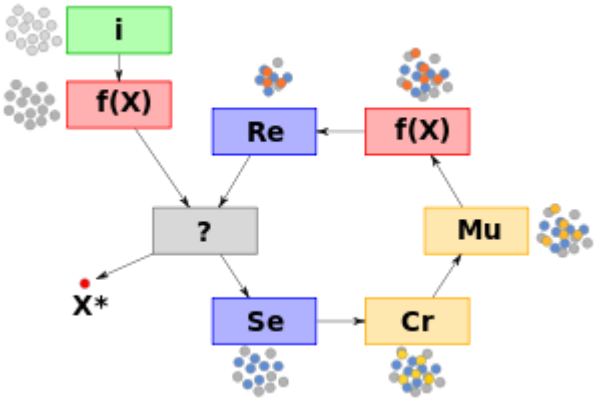
Los algoritmos evolutivos siguen la terminología de un proceso evolutivo definiendo las entidades como los cromosomas que son encargadas de representar soluciones y el conjunto de estas. En un proceso evolutivo las generaciones siguientes sufren cambios cuando los operadores genéticos definidos como los cromosomas del algoritmo se cruzan con otros para formar un nuevo individuo de la población a esto se le conoce como mutaciones, conservando así solo los individuos que logren sobrevivir a las pruebas teniendo como resultado una mejor población a las anteriores Bäck, T. (1996).

El pseudocódigo que puede representar un algoritmo evolutivo genético está dado de la siguiente manera:

```
t:= 0
Inicialización P(t)
Evaluación([P'(t)])
Hacer
        P'(t): =variación[P'(t)]
        Evaluación[P'(t)]
        P(t+1): =selección[P'(t) U Q]
        t: =t+1
mientras no se cumpla condición de término
```

Para ello se debe tener en cuenta que:

- La inicialización corresponde a la creación del conjunto inicial, es decir, la población inicial (P (0)), para tener una evaluación uniforme se procede a realizar esta asignación de manera aleatoria.
- La población de individuos está representada por P(t), donde t pretende representar la generación a la que pertenece dicha población.
- La asignación de indicador o capacidad de la solución del problema propuesto es denominada como la evaluación del modelo, para ello cada individuo de la población P(t) a evaluar debe pasar por una función de desempeño.
- El proceso de mutación es demasiado importante ya que da lugar a una nueva generación denominada P'(t) obtenida por las recombinaciones y mutaciones de la población P(t).
- La existencia del conjunto Q varía dependiendo de la selección que se realice a la población a evaluar siendo este un conjunto especial de individuos considerados para una posible selección, el contenido del conjunto Q puede ser vacío.
- La selección de una población modificada da como resultado una generación siguiente denominada P(t+1), P'(t) con los individuos que hayan pasado la función de desempeño utilizada.
- El nivel de convergencia o criterio de detención del algoritmo está denominado por el número máximo de generaciones o un tiempo determinado de ejecución del algoritmo entre otras variantes determinadas previamente.



i	inicialización
f(X)	evaluación
?	condición de término
Se	selección
Cr	cruzamiento
Mu	mutación
Re	reemplazo
X*	mejor solución

2.3. Redes neuronales

2.3.1. Definición

Las redes neuronales artificiales son unos modelos simplificado de las redes neuronales inmersas en nuestros cerebros. Por lo tanto, nunca será del todo correcto. En palabras de Manfred Eigen, Premio nobel de Química, una teoría sólo tiene la alternativa de ser correcta o incorrecta. Y un modelo incluye una tercera posibilidad, que sea correcto pero irrelevante. Modelar nuestro cerebro al nivel de neuronas individuales puede hacer que nuestro modelo sea muy ineficiente. Es por esta razón que resulta mucho más interesante analizar el comportamiento de las redes neuronales a nivel de estructuras neuroanatómicas.

Desde el punto de vista computacional, resultaría muy poco conveniente intentar simular una red neuronal compleja a nivel molecular. Ahora bien, desde el punto de vista de la Inteligencia Artificial y del Deep learning, nos interesa construir modelos computacionales cuya simulación en un ordenador sea lo más eficiente posible. posiblemente no nos interesa demasiado analizar con detalle el funcionamiento de los canales de iones en la membrana de la neurona como lo hace el BCI (Brain-Computer Interfaces), tampoco estudiar la liberación y reabsorción de neurotransmisores en el espacio sináptico entre dos neuronas. Puede que lo único que nos interese es que la señal proveniente de una proveniente de una entrada nos afecte el estado de una neurona. En función al nivel de activación, la salida de la neurona podrá variar o no. Es por esta razón que habitualmente se modele las sinapsis como un simple número real, a esto se le denomina peso asociado a la conexión.

Podemos considerar cada neurona como un nodo, las conexiones entre unas neuronas y otras las podemos modelar mediante enlaces. La efectividad de una sinapsis a la hora de

desencadenar una reacción en la neurona depende de distinto factores, como su localización espacial o su propia eficiencia. Todas esas variaciones las podemos modelar fácilmente con la ayuda de pesos numéricos, estos pesos serán positivos o negativos para sinapsis excitatorias e inhibitorias respectivamente. Como resultado obtenemos una red formada por nodos que representan neuronas y enlaces que representan sinapsis, los enlaces con pesos dan a lugar un grafo ponderado, generalmente dirigido, aunque en algunos modelos de redes neuronales también se admiten enlaces no dirigidos.

2.4. Máquinas de vector de soporte

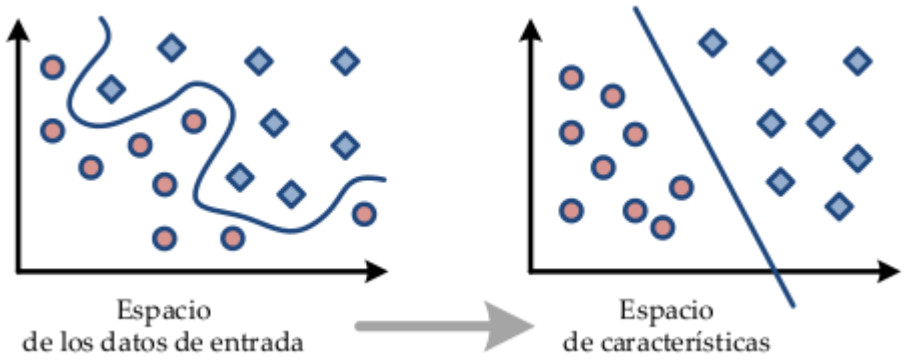
2.4.1. Definición

Las máquinas de vectores de soporte son una clase de modelos estadísticos desarrollados por primera vez a mediados de la década de 1960 por Vladimir Vapnik. En años posteriores, el modelo ha evolucionado considerablemente hasta convertirse en una de las herramientas de aprendizaje automático más flexibles y efectivas disponibles. Es un algoritmo de aprendizaje supervisado que se puede utilizar para resolver problemas de clasificación y regresión, aunque el enfoque actual se centra solo en la clasificación.

Los algoritmos de reconocimiento de patrones son una técnica que a partir de datos conocidos utilizando juicios de similitud les permite realizar analogías, Las técnicas que aplica este algoritmo es de razonamiento analógico; si observamos algo parecido con lo que conocemos, es porque posiblemente se comporte de la misma manera al que ya conocemos. La similitud siempre será subjetiva y dependerá de la perspectiva en la que veamos los datos. Peter Hart, Vladimir Vapnik y Douglas Hofstadter son algunos de los nombres más conocidos en las técnicas de patrones, Peter Hart es conocido por sus aportaciones a la Inteligencia Artificial y a la robótica

como investigador del SRI (Stanford Research Institute), su libro “Pattern Classification” Richard O. Duda, Peter E. Hart y David G. Stork (2000). Vladimir Vapnik, matemático ruso que trabajó en los laboratorios Bell de la AT&T, fue el coinventor de las máquinas de vectores de soporte SVM, Douglas Hofstadter, de la Universidad de Indiana, es hijo del Nobel de Física Robert Hofstadter y recibió el premio Pulitzer en 1979 por su obra “Gödel, Escher, Bach” Douglas R. Hofstadter. Godel, Escher (1979), Fernando Berzal (2018).

La técnica en la que se basan todos estos algoritmos son los vecinos más cercanos, buscando casos similares para establecer analogías. Un buen ejemplo son los sistemas de recomendación de Amazon o Netflix. Una forma de clasificar es asignando la misma clase a los k casos vecinos más cercanos; La asignación de un k es muy importante ya que para un k muy pequeño el nuestro clasificador puede ser sensible al ruido en los datos. Y si escogemos un k muy grande el vecindario de un punto puede incluir puntos de otras clases y estas pueden confundir a nuestro clasificador, el k tiene que estar en un término medio Fernando Berzal (2018).



Las máquinas de vectores de soporte SVM se basan en construir un clasificador, por ejemplo, lineal, en un espacio de características ampliado, puede ser infinito y este no corresponde al espacio de características asociado a los atributos que se disponen en el conjunto de entrenamiento. Las máquinas emplean el denominado truco del kernel que son unas funciones de similitud entre el hiperplano y el punto. Las SVM se hicieron muy populares a finales del siglo

XX, y tal vez el método de clasificación más utilizado durante la primera década del siglo XXI Fernando Berzal (2018).

Para resumirlo, este algoritmo busca un hiperplano linealmente separable o un límite de decisión que separe a los miembros de una clase de la otra. Si existe tal hiperplano, ¡el trabajo está hecho! Si tal hiperplano no existe, SVM usa un mapeo no lineal para transformar los datos de entrenamiento en una dimensión superior. Luego busca el hiperplano de separación lineal óptimo. Con un mapeo no lineal apropiado a una dimensión suficientemente alta, los datos de dos clases siempre pueden estar separados por un hiperplano. El algoritmo SVM encuentra este hiperplano utilizando vectores de soporte y márgenes. Como algoritmo de entrenamiento, SVM puede no ser muy rápido en comparación con otros métodos de clasificación, pero debido a su capacidad para modelar límites no lineales complejos, SVM tiene una alta precisión. SVM es comparativamente menos propenso al sobreajuste. SVM se ha aplicado con éxito al reconocimiento de dígitos escritos a mano, clasificación de texto, identificación de hablantes, etc.

2.5. Cadenas de Márkov

2.5.1. Definición

La matemática como herramienta permite realizar estudios al comportamiento de ciertos eventos, ha permitido observar fenómenos aleatorios y procesos estocásticos. Un fenómeno determinista se compone de eventos aleatorios generado por resultados únicos y/o previsibles obtenidos experimentalmente bajo las mismas condiciones de evaluación.

El matemático ruso Andréi Márkov introdujo en el año de 1906 un modelo matemático nombrado en su nombre como cadenas de Márkov que consistía en un algoritmo probabilístico cuyo comportamiento se determina exclusivamente en el evento inmediatamente anterior.

Para la teoría de la probabilidad es importante cuantificar los resultados obtenidos, por tal motivo se asigna un valor cuantificable a cada uno de los mismos. Al hablar de las cadenas de Márkov se debe conocer que es un algoritmo que basa su ejecución y aprendizaje en la probabilidad, más exactamente en los procesos estocásticos donde se realiza una observación a una serie de eventos y con base en las acciones inmediatamente anteriores se puede tener una probabilidad del comportamiento que puedan tener los siguientes eventos y los valores que pueda tomar en un instante de tiempo A.T. Bharucha-Reid (1960) y West, D. (2000).

2.6. Algunos estudios realizados

Según la investigación realizada por West (2000), teniendo en cuenta la investigación disponible sobre la predicción de dificultades financieras, sugiere que los modelos de redes neuronales muestran buen potencial, pero carecen de las ventajas de las técnicas estadísticas clásicas. El autor sugiere emplear mayor número de repeticiones para la formación de redes neuronales para obtener mayor predicción, dada la naturaleza estocástica del proceso. El autor realiza una comparación de métodos no paramétricos, fundamentalmente redes neuronales, y métodos paramétricos:

- Métodos no paramétricos
    - Redes neuronales MOE (Mixture of experts)
    - Redes neuronales RDF (Radial basis function)
    - Redes neuronales MLP (Multi-layer perceptrón)

    - Redes neuronales LVQ (Learning vector quantization)
    - Redes neuronales FAR (Fuzzy adaptive resonance)
    - K vecinos más próximos
    - Estimación núcleo de densidad o densidad de kernel.
    - Árbol de decisión CART
- Métodos paramétricos
    - Análisis discriminante lineal
    - Regresión logística

En la investigación se utilizan dos conjuntos de datos reales con particiones en datos de entrenamiento y de prueba, con 10 veces la validación cruzada. Se realizan diez repeticiones de cada ensayo de redes neuronales y por último aplica a los modelos el test de McNemar ChiSquare, que según Dietterich (1998) ha demostrado ser el test más eficiente para los algoritmos de aprendizaje supervisado.

Analizando los resultados obtenidos, West (2000) llega a la conclusión que se obtienen resultados muy similares en ambas bases de datos.

En aplicaciones de riesgo crediticio y detección de fraude Baesens et al., (2003b).

En el trabajo de Fan and Palaniswami (2000) se predicen situaciones de incumplimiento en compañías. Los autores utilizan una muestra de 174 empresas australianas (86 con incumplimiento), y aplican modelos predictivos utilizados por los autores Altman (1968), Orgler (1980), Lincoln (1982) y uno propio (en el que utilizan las variables de los trabajos de estos tres autores más otras 5 adicionales). Los autores realizan la investigación comparando cuatro técnicas para la predicción: SVM, análisis discriminante, redes neuronales multiperceptron y aprendizaje de cuantificación vectorial. Concluyen que la mejor técnica predictiva es SVM.

En los trabajos de H ardle et al. (2004) se utiliza la técnica SVM comparándola con análisis discriminante para predecir la quiebra de 84 compañías. Llegan a la conclusión que las diferencias obtenidas aplicando las dos técnicas no es estadísticamente significativa, por tanto, no pueden concluir que SVM es mejor clasificador que el análisis discriminante.

En el artículo Baesens et al. (2003b) se desarrolla un estudio comparativo de diferentes técnicas de clasificación en ocho conjuntos de datos reales de créditos de instituciones financieras (Benelux1, Benelux2, UK1, UK2, UK3, UK4, Alemania y Australia).

3. Análisis y desarrollo

3.1. Preparación de entorno de desarrollo

La implementación del software correspondiente a la red neuronal y la depuración de variables se realizó sobre el sistema operativo Ubuntu en su versión 18.04.1, utilizando los lenguajes de programación Python con las librerías TensorFlow y Keras, los datasets que contienen la información con la cual se alimenta la red neuronal fueron construidos en el lenguaje de programación JavaScript. Para ello se prepara el entorno de desarrollo instalando las siguientes dependencias ejecutando las siguientes instrucciones en la terminal del dispositivo:

- Instalación de paquetes

- sudo apt update
- sudo apt install python3-dev
- sudo apt install python3-pip
- sudo pip3 install -U virtualenv
- sudo npm install

- Verificación de versiones

```
python --version    # > 3.4
pip3 --version      # >= 19.0
virtualenv --version
```

- Creación de un entorno virtual para el proyecto en el que posteriormente se va a instalar la librería de Python llamada TensorFlow

```
virtualenv --system-site-packages -p python3 ./creditRisk
```

- Activación del entorno virtual

```
source ./creditRisk/bin/activate
```

- Activación del módulo pip en el nuevo entorno

```
pip install --upgrade pip
```

- Para observar los paquetes instalados dentro del entorno

```
pip list
```

- Para salir del entorno

```
deactivate
```

- Instalar TensorFlow

- pip install tensorflow
- sudo pip install h5py

- Otras instalaciones

- pip install matplotlib
- pip install pandas

- Verificar instalación

- python -c "import tensorflow as tf;print(tf.reduce_sum(tf.random.normal([1000, 1000])))"

3.2. Análisis y depuración de los datos

La red neuronal descrita en este proyecto se alimenta de información otorgada por la entidad financiera FAVI UTP, dicha información se encuentra segmentada en diversas de hojas de cálculo en formato ".xlsx" provenientes del archivo "análisis cartera.xlsx", sin embargo las hojas de cálculo contienen información que es relevante para la investigación en curso y otros datos que no se tendrán en cuenta para ello se procede a definir cuantas y cuales son las variables a evaluar por la red neuronal obtenidas de las siguientes hojas de cálculo:

- Clientes

- Clientes Favi
- Cartera
- Aportes
- Cal2017
- Cal2018
- Cal2019

La información con la cual se procede a realizar el entrenamiento está contenida en las hojas de cálculo Clientes Favi y las hojas de cálculo Cal2017, Cal2018 y Cal2019.

Obteniendo de esta manera información financiera y personal de cada uno de los clientes de la cooperativa FAVI UTP.

Posterior a ello se procede a determinar cuáles son las variables determinantes para considerar para evaluación en el momento que un usuario solicita un crédito a la cooperativa FAVI UTP seleccionando así las siguientes variables:

0. Edad
1. Asociado
2. Estado
3. Antigüedad
4. Personas a cargo
5. Estado civil
6. Posee propiedad
7. Posee otros activos

8. Total aportes sociales
9. Total ahorros
10. Pasivo obligaciones
11. Valor comercial propiedad
12. Activos inversiones
13. Otros activos
14. Total activos
15. Total pasivos
16. Total ingresos
17. Salario
18. Ingresos honorarios
19. Ingresos arriendos
20. Ingresos financieros
21. Otros ingresos
22. Egresos financieros
23. Egresos familiares
24. Egresos fijos hogar
25. Egresos personales
26. Egresos arriendo hipoteca
27. Otros egresos

Calificaciones consideradas por cada uno de los archivos históricos otorgados por la entidad está considerada por variables de tipo alfabético descritas de la siguiente forma:

A. Representa un retraso de 0 días.
B. Representa un retraso entre 30 y 60 días.
C. Representa un retraso entre 60 y 90 días.
D. Representa un retraso entre 90 y 120 días.
E. Representa un retraso de más de 120 días.

Luego de establecer cuáles son las variables a considerar en la construcción y evaluación de la red neuronal se procede a crear los archivos que contendrán la data, de esta manera se descartan las variables no consideradas en el proyecto de investigación. Para la creación de los archivos cuyo contenido es la data principal de la evaluación se utilizan los archivos de extensión ".js" como lo son "dataGenerator.js" y "dataOrganization.js".

La función del archivo "dataGenerator.js" es convertir los archivos de tipo JSON partiendo de archivos de tipo CSV, luego se procede a obtener la información de los archivos calendario y Clientes Favi generando así la data de entrada y salida de la red neuronal, posterior a esto se procede a asignar la calificación de cada uno de los clientes proveniente de los archivos cal2017, cal2018 y cal2019 respectivamente utilizando el archivo "dataOrganization.js".

Se procede a realizar un proceso de estandarización de variables para tener una mejor precisión a la hora de entrenar la red neuronal, para ello se procede a dividir entre 100 las variables edad, personas a cargo y antigüedad para obtener valores comprendidos entre 0 y 1, luego se dividen entre 100000000 todas las variables de carácter financiero como lo son Total aportes sociales Total ahorros, Pasivo obligaciones, Valor comercial propiedad, Activos

inversiones, Otros activos, Total activos, Total pasivos, Total ingresos, Salario, Ingresos honorarios, Ingresos arriendos, Ingresos financieros, Otros ingresos, Egresos financieros, Egresos familiares, Egresos fijos hogar, Egresos personales, Egresos arriendo hipoteca, Otros egresos.

3.3. Construcción de red neuronal.

La red neuronal implementada para este proyecto está documentada en un repositorio de GitHub y se encuentra como un anexo de este documento.

Para la construcción de la red neuronal previamente se toman en cuenta cuales son los parámetros que esta va a evaluar, es decir las variables de entrada para las que se tuvieron en cuenta los archivos de *datasets* creados con antelación compuestos por archivos de tipo *JSON* que se cargan con los datos de cada uno de los clientes de la cooperativa FAVI UTP y cuyo contenido es información personal, financiera y comportamental de cada asociado.

Se determinan cuales son las salidas esperadas tras el entrenamiento de la red neuronal, es decir, cuáles son las categorías de calificación de riesgo obtenidas posterior al entrenamiento definiéndose de la siguiente manera:

0. sin retrasos.
1. que no pase de B, últimos 6 meses no haya pasado de A.
2. A y B.
3. en A, B y C, y que en los últimos 6 meses no haya pasado A.
4. en A, B y C, y que en los últimos 6 meses no haya pasado B.
5. No cumple con los anteriores.
6. no es activo, menor de edad o no asociado.

Se definen tres capas y la cantidad de neuronas que estas poseerán para realizar la etapa de entrenamiento en este caso de estudio se procede a seleccionar 5 distribuciones diferentes de redes neuronales variando entre ellas la cantidad de neuronas que posee cada capa como se observa en la siguiente tabla:

Input	**First**	**Second**	**Third**	**output**	**Total**
28	50	50	50	7	150
28	50	100	50	7	200
28	100	100	100	7	300
28	100	150	100	7	350
28	150	150	150	7	450

Siendo Input la cantidad de variables de entrada a evaluar, *First* la primera capa de neuronas, *Second* la segunda capa de neuronas, *Third* la tercera capa de neuronas, output la cantidad de salidas posibles a entregar.

La primera red neuronal definida consta de 28 variables de entrada, 50 neuronas en su primera capa, 50 neuronas en la segunda capa y 50 neuronas en su tercera capa para un total de 150 neuronas con 7 salidas posibles.

La segunda red neuronal definida consta de 28 variables de entrada, 50 neuronas en su primera capa, 100 neuronas en la segunda capa y 50 neuronas en su tercera capa para un total de 200 neuronas con 7 salidas posibles.

La tercera red neuronal definida consta de 28 variables de entrada, 100 neuronas en su primera capa, 100 neuronas en la segunda capa y 100 neuronas en su tercera capa para un total de 300 neuronas con 7 salidas posibles.

La cuarta red neuronal definida consta de 28 variables de entrada, 100 neuronas en su primera capa, 150 neuronas en la segunda capa y 100 neuronas en su tercera capa para un total de 350 neuronas con 7 salidas posibles.

La quinta red neuronal definida consta de 28 variables de entrada, 150 neuronas en su primera capa, 150 neuronas en la segunda capa y 150 neuronas en su tercera capa para un total de 450 neuronas con 7 salidas posibles.

Posterior a definir cuáles son las entradas y salidas se procede a determinar la cantidad de épocas bajo las que se va a entrenar la red neuronal una y otra vez definidas de la siguiente manera 100, 200, 300, 400, 500, 600, 700, 800, 900 y 1000 épocas para cada vez que se ejecuta una red neuronal.

Luego de haber definido la cantidad de épocas se procede a escoger cuales son las funciones de activación que son las encargadas de devolver una salida a partir de un valor de entrada determinado; para este caso particular se procedió a evaluar el comportamiento de las siguientes funciones de activación *exponential, relu, hard_sigmoid, linear, sigmoid, softmax, softplus* y por último se procede a seleccionar los optimizadores para poder controlar cual es la

pérdida esperada en determinado momento, para ello se escogieron los siguientes optimizadores *Adamax, Adam, Adadelta, Adagrad, Ftrl, Nadam, RMSprop, SGD.*

3.4. Entrenamiento de red neuronal

Luego de definir previamente las variables de entrada, salidas de entrenamiento, épocas, optimizadores y funciones de activación se procede a crear un algoritmo el cual consiste en alimentar la red neuronal con la data definida previamente en el archivo input.txt. Posterior a ello el algoritmo inicia ejecutando la red neuronal variando la cantidad de neuronas por cada ejecución tomando valores de 150, 200, 300, 350 y 400 neuronas seguido a esto en cada ejecución se toman valores de la cantidad de épocas variando entre 100, 200, 300, 400, 500, 600, 700, 800, 900 y 1000 épocas para cada vez que se ejecuta una red neuronal, seguido a esto se evalúan 10 optimizadores diferentes cada vez que se ejecuta una de las redes neuronales, estas funciones de activación son las siguientes: *exponential, relu, hard_sigmoid, linear, sigmoid, softmax, softplus,* por último se evalúan 10 optimizadores diferentes en cada una de las ejecuciones de las diferentes disposiciones de redes neuronales, estas funciones de activación son las siguientes: *Adamax, Adam, Adadelta, Adagrad, Ftrl, Nadam, RMSprop, SGD.*

Los parámetros anteriormente descritos se cargan de la siguiente manera recorriendo las siguientes listas:

optimizers = ['Adamax', 'Adam', 'Adadelta', 'Adagrad', 'Ftrl', 'Nadam', 'RMSprop', 'SGD']

pasos = [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000]

activations = ['exponential', 'relu', 'hard_sigmoid', 'linear', 'sigmoid', 'softmax', 'softplus']

Luego de realizar todas las iteraciones con las diferentes combinaciones de parámetros entregados a la red neuronal se obtiene un total de 2800 entrenamientos diferentes con sus respectivos valores de entrenamiento y los valores con los cuales se prueba el resultado, esta información se entrega en un archivo llamado results.txt que se almacenará en el mismo directorio donde se encuentra el código fuente.

La red neuronal entrega 2800 archivos con pesos diferentes almacenados en un directorio llamado *weights* que debe crearse previamente en el directorio donde está el código fuente, internamente este directorio contiene los archivos individuales para cada entrenamiento cuyo interior está compuesto por los pesos con los cuales se debe cargar la red neuronal en una futura ejecución, estos archivos tienen nombre de la siguiente forma 1****.h5 variando los "*" entre 0000 y 2799.

El entrenamiento de la red neuronal se realiza mediante la ejecución del comando "python neuronalNetwork.py" en la consola de la terminal.

La manera de ejecutar la red neuronal con uno de los archivos generados que contienen los pesos es cargar el modelo en el código descomentando las siguientes líneas y e introduciendo el nombre del archivo con la extensión nombre_de_archivo.h5 a analizar de la siguiente manera:

```
model = keras.models.load_model( './weights/' + 'nombre_de_archivo.h5' )

model.summary()
```

y comentar las líneas de código posterior a ellas.

En la fase de entrenamiento el algoritmo lee el archivo input.txt y procede a tomar el 80% de la cantidad total de datos almacenados en el mismo correspondiendo a 874 datos de los 1092 existentes, dejando 218 datos para ejecutar pruebas y validar la fiabilidad del entrenamiento.

3.5. Análisis de información obtenida por parte del entrenamiento

Se observa la precisión por parte de la red neuronal con respecto a los datos de entrenamiento y los datos de prueba posterior a ello se promedian las dos precisiones para obtener la mejor precisión respecto al promedio de las redes neuronales ejecutadas en el entrenamiento, se logra observar que el optimizador RMSprop otorga el mejor comportamiento para todas las ejecuciones, también se puede observar que la cantidad de épocas que se encuentran entre los valores de 700 y 1000 arrojan los mejores resultados, como por ejemplo el archivo de nombre 11611.h5 cuya función de activación es *relu*, un optimizador *RMSprop*, 900 épocas y 300 neuronas arroja una precisión del 94.91% clasificando los usuarios de FAVI UTP en una de las 7 categorías mencionadas anteriormente.

{

"file":"11611.h5",

"functionactivation":"relu",

"optimizer":"RMSprop",

"epoch":900,

"firstlayer":100,

"secondlayer":100,

"thirdlayer":100,

"Trainaccuracy":"0.99427265",

"Testaccuracy":"0.9041096",

"avgaccuracy":"0.9491910934448242",

"resulttrain":[6,0,0,0,0,3,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,5,3,0,6,6,0,0,0,0,6,0,0,0,0,0,0,0,0
,0,6,0,0,0,6,0,0,0,0,0,6,0,5,0,0,5,0,0,1,6,6,0,0,0,6,0,0,0,0,0,0,0,1,0,0,0,6,0,0,0,0,0,0,0,0,0,0,0
,6,0,0,5,0,0,0,0,0,0,0,0,0,6,0,1,0,1,0,0,0,0,0,0,0,0,3,1,6,0,0,0,0,0,0,0,0,0,6,6,0,0,0,0,0,0,0,0,3
,0,0,0,0,6,0,0,6,0,6,0,5,1,0,0,0,0,0,0,6,0,0,6,0,0,0,0,0,0,0,0,0,0,0,0,6,3,0,5,0,0,0,0,6,0,6,0,6,6,3,0
,0,1,0,0,0,0,0,5,0,0,0,0,0,0,0,0,1,1,0,0,0,6,0,0,0,1,0,6,0,0,0,0,6,0,0,0,0,0,0,0,6,0,0,1,0,0,6,6,6,0,6,0
,0,6,0,0,0,6,0,0,0,0,0,6,0,6,0,0,5,0,0,6,6,0,0,0,0,0,5,0,0,0,0,0,0,0,0,0,0,1,0,0,0,6,0,0,0,0,6,0,6,0,0,6,0
,6,1,6,5,0,0,0,6,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,6,0,0,5,0,1,0,0,0,0,0,6,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,6,0,0,0,0,0,0,0,6,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,6,0,0,0,6,6,0,0,0,0,0
,0,0,0,0,6,0,0,0,0,0,0,6,0,3,0,0,0,0,0,0,0,6,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,0,6,0,0,0,0,0,3,0,0,0,0,0
,0,0,0,0,0,0,6,3,5,0,6,6,0,6,0,0,0,0,0,6,6,6,5,0,6,0,0,0,0,1,0,0,0,0,3,0,0,0,3,6,0,0,0,0,0,0,0,0,0,0,6,0,6
,6,0,0,0,6,3,0,0,0,0,0,1,1,6,0,6,0,0,0,1,6,0,0,0,1,6,1,1,0,6,6,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,0,0
,0,0,0,0,0,0,0,5,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,0,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,6,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,1,5,6,0,0,0,0,0,0,0,0,0,1,6,0,0,0,6,0,0,0,6,6,0,0,5,0,0,0,0,6,3,0,0,0,0,0,0,0,6,1,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,6,0,6,0,6,0,0,0,1,0,0,0,0,0,0,3,0,0,0,1,6,0,0,0,0,0,0,0,0,0,0,0,0,6,0,0,0,5,1,0,3,0
,0,3,0,0,5,0,0,0,0,0,0,6,0,0,0,0,0,0,0,0,6,6,6,0,6,0,1,0,1,0,6,0,0,0,0,0,0,0,0,0,0,0,0,5,1,0,0,0,1,0,0,0,6,1,0

,0,6,0,0,0,0,0,6,0,5,0,0,6,6,0,0,0,6,0,0,1,0,0,0,0,0,6,0,0,0,0,0,0,0,6,6,0,6,0,0,0,0,0,0,0,0,0,0,0,0
,0,1,0,0,0,0,0,0,0,3,0,0,0,0,1,6,6,0,0,0,0,0,0,5,0,0,0,3,0,0,6,0,1,0,0,0,0,0,0,0,0,0,0,0,6,0,0],

"resulttest":[0,0,0,0,0,0,0,0,6,0,6,0,0,0,6,0,0,0,0,0,0,0,0,0,0,6,0,0,6,6,0,0,0,0,0,0,0,0,0,0,0,6,0,0,
0,6,0,0,0,0,0,0,0,6,0,6,0,6,1,0,0,6,0,0,0,5,0,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,0,6,0,0,6,6,0,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,
3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,0,0,0,3,0,0,0,6,0,0,6,0,0,0,0,0,0,0,6,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,6,0,0,0,0,0,0,0,0,0,0,0,0,6,0,0,0]

}

Cada uno de los valores almacenados en la lista resulttrain corresponden a la calificación obtenida por cada uno de los clientes de la cooperativa FAVI UTP haciendo referencia al 80% anteriormente mencionado. Los datos almacenados en resulttest hacen referencia a las calificaciones de las personas con relación al 20% de los datos cargados inicialmente.

Posterior a la ejecución y entrenamiento de la red neuronal se logran obtener las siguientes configuraciones para la ejecución de casos de prueba:

Archivo	funcion de activacion	optimizador	Epocas	Capa 1	Capa 2	Capa 3	Precisión
12164.h5	relu	Nadam	900	100	150	150	0,92006779
11345.h5	relu	Adamax	500	100	100	100	0,92052281
12788.h5	hard_sigmoid	RMSprop	1000	150	150	150	0,92111123
12680.h5	softplus	RMSprop	800	150	150	150	0,92167616
12633.h5	relu	Adamax	800	150	150	150	0,92172325
12248.h5	relu	Adam	100	150	150	150	0,92223322
10904.h5	relu	Adam	700	50	100	100	0,92231166
12669.h5	hard_sigmoid	Nadam	800	150	150	150	0,92281377
11576.h5	relu	Adam	900	100	100	100	0,9234885
12129.h5	relu	Adamax	900	100	150	150	0,92399067

12689.h5	relu	Adamax	900	150	150	150	0,92401421
11625.h5	relu	Adamax	1000	100	100	100	0,92457128
10624.h5	relu	Adam	200	50	100	100	0,92508906
11828.h5	relu	Nadam	300	100	150	150	0,92514396
12465.h5	relu	Adamax	500	150	150	150	0,09256932
12612.h5	relu	Nadam	700	150	150	150	0,92577165
12556.h5	relu	Nadam	600	150	150	150	0,92633653
11779.h5	relu	RMSprop	200	100	150	150	0,92683864
12220.h5	relu	Nadam	1000	100	150	150	0,92691708
11184.h5	relu	Adam	200	100	100	100	0,92796057
11051.h5	relu	RMSprop	900	50	100	100	0,92807043
12332.h5	relu	Nadam	200	150	150	150	0,92855686
12416.h5	relu	Adam	400	150	150	150	0,92861176
12024.h5	relu	Adam	700	100	150	150	0,92917669
12731.h5	relu	RMSprop	900	150	150	150	0,92921591
11772.h5	relu	Nadam	200	100	150	150	0,92968667
12584.h5	relu	Adam	700	150	150	150	0,92976511
11044.h5	relu	Nadam	900	50	100	100	0,93030643
11912.h5	relu	Adam	500	100	150	150	0,93087918
11492.h5	relu	Nadam	700	100	100	100	0,93090272
12787.h5	relu	RMSprop	1000	150	150	150	0,931499
11604.h5	relu	Nadam	900	100	100	100	0,93205607
11352.h5	relu	Adam	500	100	100	100	0,93258172
12472.h5	relu	Adam	500	150	150	150	0,93260527
12500.h5	relu	Nadam	500	150	150	150	0,93317795
11408.h5	relu	Adam	600	100	100	100	0,93430781
11996.h5	relu	Nadam	600	100	150	150	0,93488836
12052.h5	relu	Nadam	700	100	150	150	0,93546891
10547.h5	relu	RMSprop	1000	50	50	50	0,93603384
12563.h5	relu	RMSprop	600	150	150	150	0,93659091
12668.h5	relu	Nadam	800	150	150	150	0,93661439
12192.h5	relu	Adam	1000	100	150	150	0,937195
12619.h5	relu	RMSprop	700	150	150	150	0,9394781
12136.h5	relu	Adam	900	100	150	150	0,94060791
12780.h5	relu	Nadam	1000	150	150	150	0,94176906
11835.h5	relu	RMSprop	300	100	150	150	0,09422947
12696.h5	relu	Adam	900	150	150	150	0,94232607
12115.h5	relu	RMSprop	800	100	150	150	0,9446249
12171.h5	relu	RMSprop	900	100	150	150	0,09451976
12724.h5	relu	Nadam	900	150	150	150	0,94576252
12059.h5	relu	RMSprop	700	100	150	150	0,94860268

11611.h5	relu	RMSprop	900	100	100	100	0,94919109

4. Aspectos administrativos

4.1. Recurso humano

El proyecto de investigación contó con el apoyo del docente guía Ingeniero Julio César López Betancur, los estudiantes del programa de Ingeniería de Sistemas y Computación Andrés Felipe Bravo Giraldo y Diego Alejandro Restrepo Sánchez.

4.2. Presupuesto

Este proyecto de investigación no tuvo gastos financieros en ninguna de sus etapas.

4.3. Cronograma

Actividad	***Fecha de inicio***	***Duración***
• *Consultar sobre técnicas de inteligencia artificial a utilizar para creación de modelo.*	*01 de noviembre 2019*	*4 semanas*
• *Consulta de trabajos realizados utilizando las técnicas de inteligencia artificial a utilizar*	*01 de noviembre 2019*	*4 semanas*

• *Definición de variables a evaluar en el modelo de inteligencia artificial.*	*01 de diciembre 2019*	*1 semanas*
• *Creación de modelos de inteligencia artificial a evaluar.*	*06 de diciembre 2019*	*1 semana*
• *Evaluación de modelos de inteligencia artificial.*	*13 de diciembre 2019*	*1 semana*
• *Selección de modelo de inteligencia artificial.*	*18 de diciembre 2019*	*5 días*
• *Ajuste de modelo de inteligencia artificial.*	*23 de diciembre 2019*	*5 semana*
• *Entrenamiento de modelo de Inteligencia artificial con los datos obtenidos de la entidad financiera FAVI.*	*28 de diciembre 2019*	*1 semanas*
• *Evaluar resultados obtenidos bajo el entrenamiento del modelo.*	*04 de enero 2020*	*5 días*
• *Conclusiones*	*09 de enero 2019*	*1 semanas*

5. Recomendaciones

- Tener una data mas amplia para obtener un mejor entrenamiento de la red neuronal.

- Considerar datos históricos más antiguos.
- Considerar los datos de clientes a los cuales se les negó un crédito en la entidad financiera.
- Considerar información proveniente de las centrales de riesgo.

6. Anexos

El código fuente relacionado a este proyecto de investigación se encuentra como anexo en el directorio cuyo nombre es "*CreditRisk"* y se encuentra grabado junto con este documento en el CD entregado a la facultad de Ingenierías de la Universidad Tecnológica de Pereira.

7. Referencias bibliográficas

Poole, David (2018). Computational Intelligence: A Logical Approach. Nueva York: Oxford University Press. p. 1.

Fernando Berzal (2018) Book REDES NEURONALES & DEEP LEARNING - ISBN 13: 978-84-338-6311-9

J. H. Holland. University of Michigan Press, Ann Arbor. (1975). *Adaptation in Natural and Artificial Systems*.

D. E. Goldberg. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*.

Elaine Rich. Artificial Intelligence. McGraw-Hill, (1983). ISBN 0070522618

Richard O. Duda, Peter E. Hart, y David G. Stork. Pattern Classification. Wiley-Interscience, 2nd edition, 2000. ISBN 0471056693

Douglas R. Hofstadter. Godel, Escher, Bach: An Eternal Golden Braid. Basic Books, (1979). ISBN 0465026850

Bäck, T. (1996) *Evolutionary Algorithms in Theory and Practice*, Oxford, NY.

A.T. Bharucha-Reid (1960). Elements Of The Theory of Markov Processes And Their Applications. McGraw Hill Series in Probability and Statistics.

West, D. (2000). Neural network credit scoring models. Computers and Operations Research, 27:1131-1152

Dietterich, T. (1998). Approximate statistical test for comparing supervised classification learning algorithms. Neural computation, 10(7):1895-1923.

Fan, A. and Palaniswami, M. (2000). Selecting bankruptcy predictors using a support vector machine approach. 6:354-359.

Altman, E. (1968). Financial Ratios, discriminant analysis and the prediction of corporate bankruptcy. Journal of Finance, 23($):589-609.

Baesens, Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., and Vanthienen, J. (2003b). Benchmarking state of the art classification algorithms for credit scoring. Journal of the Operational Research Society, 54(6):627-635.