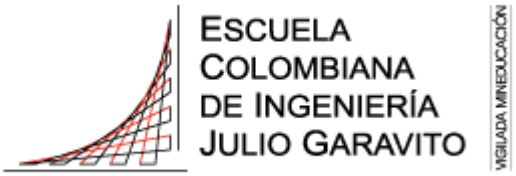
DISEÑO DE UN MODELO DE EVALUACIÓN DE PROYECTOS A PARTIR DE HERRAMIENTAS DE MACHINE LEARNING O APRENDIZAJE AUTOMATIZADO.



JORGE SEBASTIAN CARO MESA

CAMILO ANDRES CRUZ RODRIGUEZ

LAURA TATIANA NOVA BARRETO

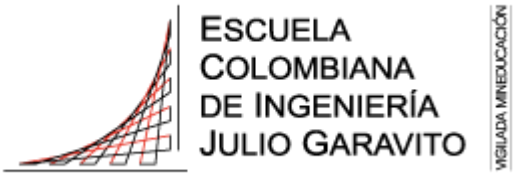
ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

UNIDAD DE PROYECTOS

MAESTRÍA EN DESARROLLO Y GERENCIA INTEGRAL DE PROYECTOS

2021

DISEÑO DE UN MODELO DE EVALUACIÓN DE PROYECTOS A PARTIR DE HERRAMIENTAS DE MACHINE LEARNING O APRENDIZAJE AUTOMATIZADO.



JORGE SEBASTIAN CARO MESA

CAMILO ANDRES CRUZ RODRIGUEZ

LAURA TATIANA NOVA BARRETO

Trabajo de grado

Director

Ing. M.Sc. Rodrigo Buzeta

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

UNIDAD DE PROYECTOS

MAESTRÍA EN DESARROLLO Y GERENCIA INTEGRAL DE PROYECTOS

2021

Publicado en 2021 por la Escuela Colombiana de Ingeniería Julio Garavito (Ak 45 No. 205-59 Bogotá D.C., Colombia).

NOTA DE ACEPTACIÓN:

El Trabajo de Grado titulado “Diseño de un modelo de evaluación de proyectos de construcción a partir de herramientas de machine Learning o aprendizaje automatizado.”, presentado por los estudiantes Jorge Sebastian Caro Mesa, Camilo Andrés Cruz Rodríguez y Laura Tatiana Nova, quienes optan al título de Magister en Desarrollo y Gerencia Integral de Proyectos, cumple con todos los lineamientos y los requisitos exigidos por la Unidad de Proyectos de la Escuela Colombiana de Ingeniería Julio Garavito.

Ing. M.Sc. Rodrigo Buzeta

Director Trabajo de Grado

Bogotá D.C., 14 de mayo de 2021.

AGRADECIMIENTOS

Un agradecimiento especial al Ing. M.Sc. Camilo Blanco Vargas, experto en iniciativas de tecnología y Machine Learning por su colaboración intelectual, retroalimentación e interés en este Trabajo de Grado

RESUMEN

Organizaciones como The Standish Group publican anualmente estadísticas del porcentaje de éxito o fracaso de proyectos de TI (Tecnología Informática), las cuales revelan que alrededor del 31.1% de los proyectos que se realizan en el transcurso de un año fracasan o son cancelados, el 52.7% superan en costo, tiempo y/o funcionalidad prometida y solo el 16.2% de los mismos se consideran exitosos. Ahora bien, estas cifras corresponden al sector de las Tecnologías de la Información y la Comunicación (TIC), el cual para el año 2017 representaba el 6.5% del PIB mundial; en este sentido, si estas cifras se proyectaran de forma equivalente en el sector de la construcción, cuyo aporte al PIB mundial en el mismo año superaba el 13%, se confirmaría la existencia de una situación alarmante en la ejecución de los proyectos.

Cabe señalar que las cifras anteriores, coinciden con las estadísticas entregadas por el Project Management Institute en su informe anual, puesto que para el 2019 los proyectos que fracasaron en su totalidad superaban el 18% y los que no cumplieron con sus objetivos el 39%. No obstante, aquellas empresas que han venido implementado nuevas prácticas, conocidas como “PMTQ Innovators”, han logrado disminuir en un 42% los proyectos que han excedido su presupuesto inicial y en un 46% aquellos que excedieron el tiempo planeado.

Por lo anterior, son las alarmantes cifras las que ratifican la necesidad de nuevas herramientas que contribuyan a la disminución de las tasas de fracaso de los proyectos; es aquí donde toma fuerza la incursión del ámbito tecnológico en la evaluación de proyectos y donde esta investigación demuestra que es posible predecir el éxito de los proyectos haciendo uso de herramientas y algoritmos del Machine Learning, a través de un sistema de aprendizaje supervisado, modelado por medio de una red neuronal artificial entrenada a partir de bases de

datos históricos de proyectos ejecutados, a los cuales se les calcula su éxito bajo los lineamientos de la triple restricción: costo, tiempo y alcance.

Es así, que se da a conocer una herramienta que proporciona a la evaluación de proyectos un indicador diferente a las métricas financieras tradicionales (TIR, VPN, Rentabilidad del Negocio, ROI y B/C (Costo - beneficio)) para la toma de decisiones, con el fin de minimizar la tasa de proyectos fracasados. Finalmente, al entrenar esta herramienta con la data histórica de la última década de los proyectos de infraestructura en Colombia, se determina que si el Estado hubiese usado este modelo, más de 45.000 proyectos no exitosos con costos superiores a 60.000 millones de pesos, pudieron haber sido reestructurados o rechazados previo a su inicio; adicionalmente, se hubieran podido destinar a otros proyectos, recursos superiores a los 5.000 millones de pesos, que fueron ejecutados en más de 1000 contratos fracasados por año.

Palabras clave. Gestión de proyectos, aprendizaje automático, inteligencia artificial, predicción, evaluación de proyectos

ABSTRACT

Organizations such as The Standish Group annually publish statistics on the percentage of success or failure of IT (Information Technology) projects, which reveal that around 31.1% of projects carried out in the course of a year fail or are canceled, 52.7% exceed in cost, time and/or promised functionality, and only 16.2% of them are considered successful. Now, these figures correspond to the Information and Communication Technologies (ICT) sector, which for the year 2017 represented 6.5% of world GDP; accordingly, if these figures were projected equivalently in the construction sector, whose contribution to world GDP in the same year exceeded 13%, the existence of an alarming situation in the execution of projects would be confirmed.

It should be noted that the figures above correspond with the statistics provided by the Project Management Institute in its annual report, since by 2019 the projects that failed in their entirety exceeded 18% and those that did not meet their objectives were above 39%. However, those companies that have been implementing new practices, known as “PMTQ Innovators”, have managed to reduce by 42% the projects that have exceeded their initial budget, and by 46% those that exceeded the planned time.

Therefore, it is the alarming figures that ratify the need for new tools that contribute to the reduction of the failure rates of projects; it is here where the incursion of the technological field in the evaluation of projects takes force and where this research shows that it is possible to predict the success of projects by making use of Machine Learning tools and algorithms, through a supervised learning system modeled by through an artificial neural network, trained from historical databases of executed projects, whose success is calculated under the guidelines of the triple restriction: cost, time and scope.

Thus, is released a tool that provides project evaluation with a different indicator from traditional financial metrics (IRR, NPV, Business Profitability, ROI, and C / B (Cost-benefit)) for decision making, in order to minimize the rate of failed projects. Finally, when training this tool with the historical data of the last decade of infrastructure projects in Colombia, it is determined that if the State had used this model, more than 45,000 unsuccessful projects with costs exceeding 60.000 million pesos, could have been restructured or rejected before its inception; additionally, above 5.000 million pesos, which were executed in more than 1000 failed contracts per year, could have been allocated to other projects.

Keywords. Project management, machine learning, artificial intelligence, prediction, project evaluation

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

GLOSARIO

Alcance / Scope. “Suma de productos, servicios y resultados a ser proporcionados como un proyecto” (Project Management Institute PMI®, 2017)

Anaconda. “Una distribución de Python y R optimizada, descargable, gratuita, de código abierto, de alto rendimiento y con más de 250 paquetes incluidos automáticamente. Anaconda ofrece la opción de instalar fácilmente más de 7500 paquetes de código abierto adicionales para ciencia de datos, incluidos análisis científicos y avanzados” (Anaconda, Inc, 2021)

Aprendizaje automatizado / Machine Learning. Machine Learning es una forma de la IA que permite a un sistema aprender de los datos en lugar de aprender mediante la programación explícita. Sin embargo, Machine Learning no es un proceso sencillo. Conforme el algoritmo ingiere datos de entrenamiento, es posible producir modelos más precisos basados en datos. Un modelo de Machine Learning es la salida de información que se genera cuando entrena su algoritmo de Machine Learning con datos. Después del entrenamiento, al proporcionar un modelo con una entrada, se le dará una salida. Por ejemplo, un algoritmo predictivo creará un modelo predictivo. (IMB Colombia, 2020)

Aprendizaje supervisado. El aprendizaje supervisado comienza típicamente con un conjunto establecido de datos y una cierta comprensión de cómo se clasifican estos datos. El aprendizaje supervisado tiene la intención de encontrar patrones en datos que se pueden aplicar a un proceso de analítica. Estos datos tienen características etiquetadas que definen el significado de los datos. (IMB Colombia, 2020)

Aprendizaje no supervisado. El aprendizaje no supervisado se utiliza cuando el problema requiere una cantidad masiva de datos sin etiquetar. La comprensión del significado

detrás de estos datos requiere algoritmos que clasifican los datos con base en los patrones o clústeres que encuentra. El aprendizaje no supervisado lleva a cabo un proceso iterativo, analizando los datos sin intervención humana. (IMB Colombia, 2020)

Ciclo de Vida del Proyecto / Project Life Cycle. “Serie de fases que atraviesa un proyecto desde su inicio hasta su conclusión” (Project Management Institute PMI®, 2017)

Ciclo de Vida Predictivo / Predictive Life Cycle. “Forma de ciclo de vida del proyecto en la cual el alcance, el tiempo y el costo del proyecto se determinan en las fases tempranas del ciclo de vida” (Project Management Institute PMI®, 2017)

Contrato / Contract. “Un contrato es un acuerdo vinculante para las partes en virtud del cual el vendedor se obliga a proveer el producto, servicio o resultado especificado y el comprador a pagar por él” (Project Management Institute PMI®, 2017)

Jupyter Notebook: es una aplicación web de código abierto que le permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Los usos incluyen: limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos, aprendizaje automático y mucho más. (Jupyter, 2020)

Datos Abiertos Colombia. “Los datos abiertos son información pública dispuesta en formatos que permiten su uso y reutilización bajo licencia abierta y sin restricciones legales para su aprovechamiento. En Colombia, la Ley 1712 de 2014 sobre Transparencia y Acceso a la Información Pública Nacional, define los datos abiertos en el numeral sexto como “todos aquellos datos primarios o sin procesar, que se encuentran en formatos estándar e interoperables que facilitan su acceso y reutilización, los cuales están bajo la custodia de las entidades públicas o privadas que cumplen con funciones públicas y que son puestos a disposición de cualquier ciudadano, de forma libre y sin restricciones, con el fin de que terceros

puedan reutilizarlos y crear servicios derivados de los mismos" (Ministerio de Tecnologías de la Información y las Comunicaciones, 2021)

Deep Learning. Es un método específico de Machine Learning que incorpora las redes neuronales en capas sucesivas para aprender de los datos de manera iterativa. El Deep Learning es especialmente útil cuando se trata de aprender patrones de datos no estructurados. Las redes neuronales complejas de Deep Learning están diseñadas para emular cómo funciona el cerebro humano, así que las computadoras pueden ser entrenadas para lidiar con abstracciones y problemas mal definidos. Las redes neuronales y el Deep Learning se utilizan a menudo en el reconocimiento de imágenes, voz y aplicaciones de visión de computadora. (Ministerio de Tecnologías de la Información y las Comunicaciones, 2021)

Navegador Anaconda / Anaconda Navigator. "Una interfaz gráfica de usuario (GUI) de escritorio incluida en todas las versiones de Anaconda que le permite administrar fácilmente paquetes, entornos, canales y computadoras portátiles conda sin la necesidad de utilizar la interfaz de línea de comandos (CLI)" (Anaconda, Inc, 2021).

Proyecto / Project. "Esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único" (Project Management Institute PMI®, 2017).

Python. Python es un lenguaje de programación que le permite trabajar más rápidamente e integrar sistemas de manera más efectiva. (Python, 2020)

SECOP. "El Sistema Electrónico para la Contratación Pública – SECOP- es el medio de información oficial de toda la contratación realizada con dineros públicos. El SECOP es el punto único de ingreso de información para las entidades que contratan con cargo a recursos públicos" (Colombia Compra Eficiente, 2020)

INTRODUCCIÓN

El planteamiento de un modelo de evaluación de proyectos radica en su alta tasa de fracaso, lo que trae consigo el incumplimiento del alcance, tiempo y/o costo; por lo tanto, aplicar nuevas tecnologías en la evaluación de proyectos en la fase de planeación permitirá obtener una predicción del índice de éxito o fracaso, lo que se convierte en el objetivo del presente trabajo.

Con el fin de dar respuesta al fracaso de los proyectos se opta por diseñar un modelo de evaluación de proyectos a partir de herramientas y algoritmos existentes de Machine Learning o aprendizaje automatizado, que permita predecir el índice de éxito de estos. Dicho modelo, demanda una data histórica de proyectos y unos conocimientos previos en gestión de proyectos, ML y redes neuronales, de tal manera se asegura que dicho modelo será acorde a los lineamientos de la triple restricción y contribuiría positivamente a la evaluación de proyectos.

1 PERFIL DE LA INVESTIGACIÓN

1.1 Marco Referencial

Para apoyar el proceso de búsqueda y clasificación de la literatura disponible se utiliza Scopus, una de las bases de datos que provee la Escuela Colombiana de Ingeniería, conocida en el ámbito de la investigación a nivel internacional, por ser una herramienta confiable con un número significativo de publicaciones especializadas y actualizadas. (Scopus®, 2021)

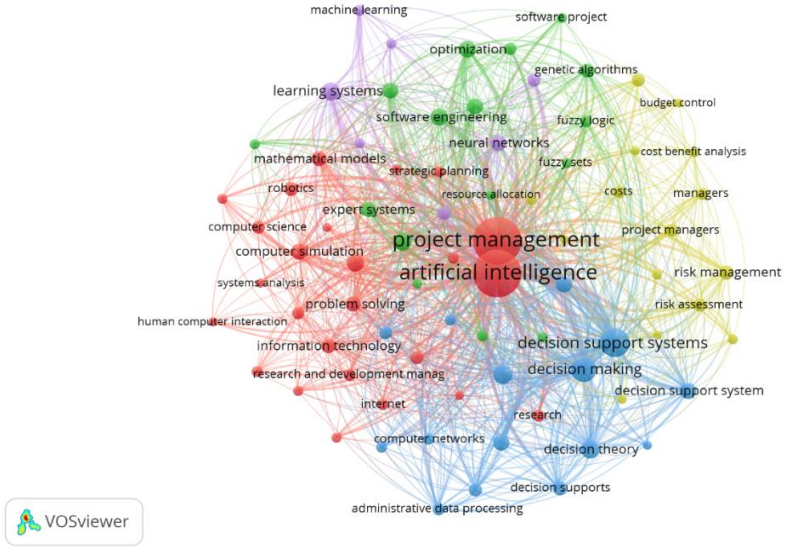
Esta investigación, se inicia con la construcción de un criterio de búsqueda con palabras clave, cuyos resultados se cargan en un metabuscador VOSViewer (Waltman, s.f.), herramienta que tiene la capacidad de construir mapas de visualización con la información obtenida de Scopus; estos mapas de interacción de los temas asociados a las palabras clave usadas en la búsqueda, permiten visualizar la relación de los temas encontrados, al igual que la interacción de las diferentes fuentes de información.

En este caso, se obtuvieron los siguientes mapas a partir de la búsqueda realizada en Scopus con las palabras clave “Project Management” y “Artificial Intelligence”; en primer lugar, se encuentra la **Figura 1**, que corresponde al mapa de interacción de temas asociados encontrados en la Búsqueda, esta gráfica evidencia la interacción que tienen las palabras clave con temas como el Machine Learning, toma de decisiones, redes neuronales, gerencia de proyectos, costos y riesgos. Luego se muestra en la

Figura 2, el mapa de interacción de las fuentes halladas para los diferentes temas asociados con la Búsqueda, dependiendo del año de publicación, en este caso predomina la información comprendida entre los años 2008 y 2010. Por último, en la **Figura 3** el Mapa de calor donde se visualizan los temas con hallazgos, coincidencias y mayor publicación de información.

Figura 1

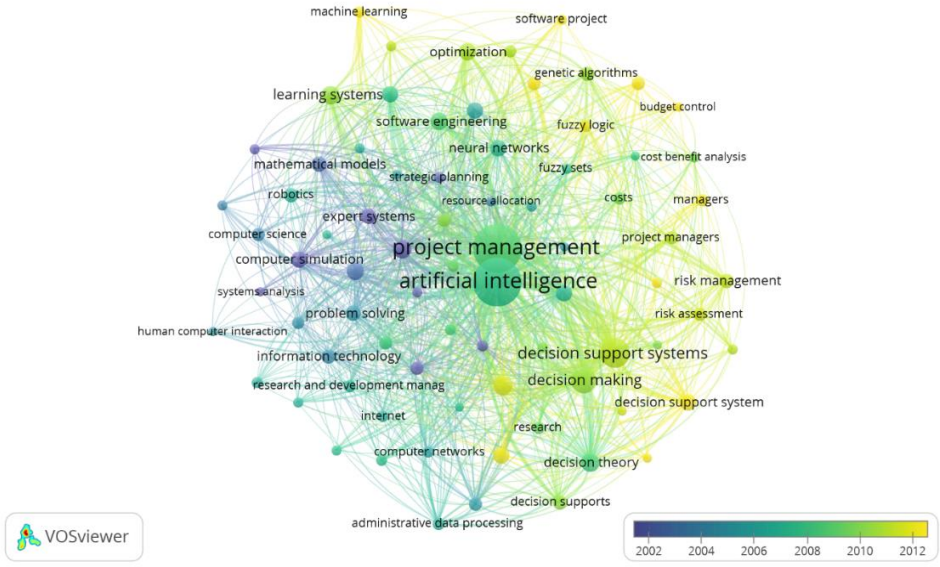
Mapa de Interacción.



Nota. Figura creada con el metabuscador VOSViewer.

Figura 2

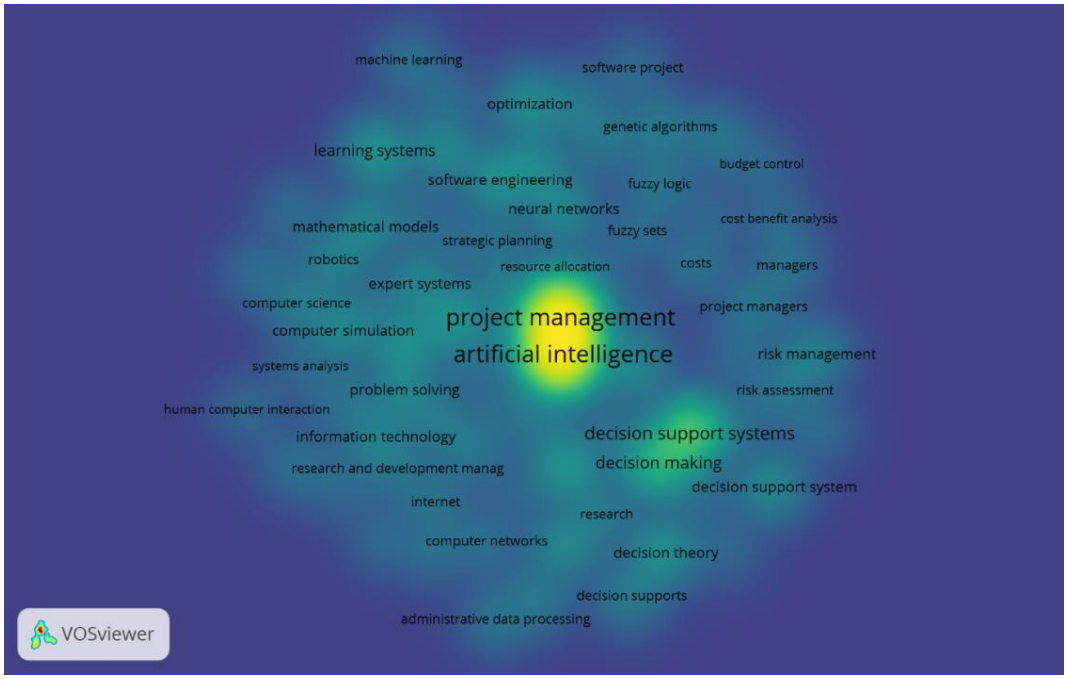
Mapa de Interacción por año.



Nota. Figura creada con el metabuscador VOSViewer.

Figura 3

Mapa de calor.



Nota. Figura creada con el metabuscador VOSViewer.

En consecuencia, con los resultados obtenidos en esta primera etapa de la investigación, se confirma la necesidad de profundizar en los temas de interacción; con lo cual se percibe que a pesar de que la historia neuronal a nivel computacional tiene indicios con las máquinas cibernéticas en el año 100 a.C., es en 1936 cuando se originan los estudios de computación neuronal y es en 1957 cuando el científico Frank Roseblatt inicia el desarrollo del perceptrón, la primera unidad neuronal, a partir de la cual nacen las redes neuronales artificiales que hoy se conocen. Sin embargo, antes de profundizar en el tema es necesario tener claridad sobre que son las redes neuronales artificiales, para lo cual se toma la definición de Hecht-Nielsen "… un sistema de computación hecho por un gran número de elementos simples, elementos de proceso muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas" (Hetch-Nielsen, 1988).

Su desarrollo está inspirado en las redes neuronales biológicas del cerebro, por lo cual su fin es lograr que se comporten de manera similar, cuya estructura tenga como unidad mínima la neurona, que es aquella que se encarga de procesar las señales de entrada de tal forma que generen salidas; es así, como se inicia una tecnología computacional cuyas aplicaciones en la actualidad van desde la biología, el medio ambiente, la medicina hasta las finanzas, la manufactura, empresas e incluso el ámbito militar.

Por otro lado, en lo que refiere a los antecedentes de la gestión de proyectos, se contempla que sus inicios son paralelos a los inicios de la civilización, un ejemplo de ello, es la construcción de las pirámides de Egipto que a pesar de ser un proyecto de gran magnitud no tenía gran complejidad; a medida que fue evolucionando la civilización lo hicieron los proyectos y fue en la revolución industrial donde adquirieron mayor complejidad y se evidenció la necesidad de una herramienta de planificación y control (Roberts & Wallace, 2014).

Es en los años 60 que se instituye el PMI, con la concepción de métodos más complejos de gestión de proyectos, pero realmente fue en los años 70 donde nos acercamos más a lo que se conoce hoy en día, en donde se avanza hacia el concepto de evaluación y medición de desempeño, teniendo en cuenta costo y tiempo, y lo más representativo el valor agregado de los proyectos (Roberts & Wallace, 2014). En consecuencia, actualmente desde la visión del PMBOK, se dice que: “La gestión de proyecto, entonces, es el uso del conocimientos, habilidades y técnicas para ejecutar proyectos de manera eficaz y eficiente. Se trata de una competencia estratégica para organizaciones, que les permite vincular los resultados de un proyecto con las metas comerciales para posicionarse mejor en el mercado” (Project Management Institute PMI®, 2013).

Así mismo, se tiene la concepción del proyecto como tal, el cual es definido por el PMBOK como “…un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o

resultado único.", el cual dentro de sus áreas de conocimiento abarca la gestión de alcance, tiempo (cronograma) y costo, las cuales se reconocen en la gestión de proyectos como la triple restricción, considerando que siempre debe existir un balance entre estas y que en el caso de que una de estas se vea afectada, otra o las dos restantes se debe modificar con el fin de mantener este equilibrio (Project Management Institute PMI®, 2017).

Teniendo en cuenta el informe "Pulse of Profession" presentado por el Project Management Institute en el año 2018, en el cual se evidencia la importancia del control del alcance, cuya corrupción y falta de claridad son producto de los tres motivos principales del fracaso de los proyectos, y que, en concordancia con lo anterior, genera desbalances en el costo y tiempo; según el informe se contempla que durante el 2018 el 52% de los proyectos presentaron corrupción del alcance, de tal manera que su control es sinónimo de disminución en la tasa de fracaso de los proyectos.

Finalmente, es el informe "Pulse of Profession" del año 2019 el que ratifica que el desempeño de los proyectos no está mejorando, y que es hora de incursionar en el ámbito tecnológico, como lo citan en el informe "… 85% de los participantes de la Encuesta CEO 2019 de PwC afirma que la IA "cambiará significativamente su manera de hacer negocios en los próximos cinco años". Y casi dos tercios de los CEO del mundo la consideran un disruptor que incluso supera a Internet" (Project Management Institute PMI®, 2019). Es aquí donde inicia este proyecto, en busca de una aplicación de estas nuevas tecnologías en la gestión de proyectos específicamente en la evaluación de su desempeño, cuyo fin, tal vez no sea descubrir la fórmula secreta para el éxito de los proyectos, pero donde la predicción del indicador de éxito será la base de la toma de decisiones informadas de los negocios.

1.2 Planteamiento del Problema y Justificación

Actualmente, las buenas prácticas en la gestión de proyectos plantean procesos de monitoreo y control con el fin de lograr que estos sean exitosos en términos de la triple restricción; sin embargo, a pesar de implementar dichos métodos, los proyectos presentan altas tasas de fracaso.

Es así que, organizaciones como The Standish Group, publican anualmente informes con estadísticas del porcentaje de éxito o fracaso de proyectos de TI (Tecnología Informática), para el año 2015, en su informe anual The Chaos Report (The Standish Group, 2015), se revela que el 19% de los proyectos fracasaron o fueron cancelados, y que, el 52% superaron en costo, tiempo y/o funcionalidad prometida, y que solo el 29% de los mismos, se consideraron exitosos. Para el año 2019, los porcentajes empeora pues aumenta a un 31.1% los proyectos que fracasaron o fueron cancelados, y al mantenerse en un 52,7% los que superaron en costo, tiempo y/o funcionalidad prometida, se disminuyen a un 16.2% los que se consideraron exitosos, estas cifras confirman la situación crítica en el desempeño de proyectos (OpenDoor Technology, 2019).

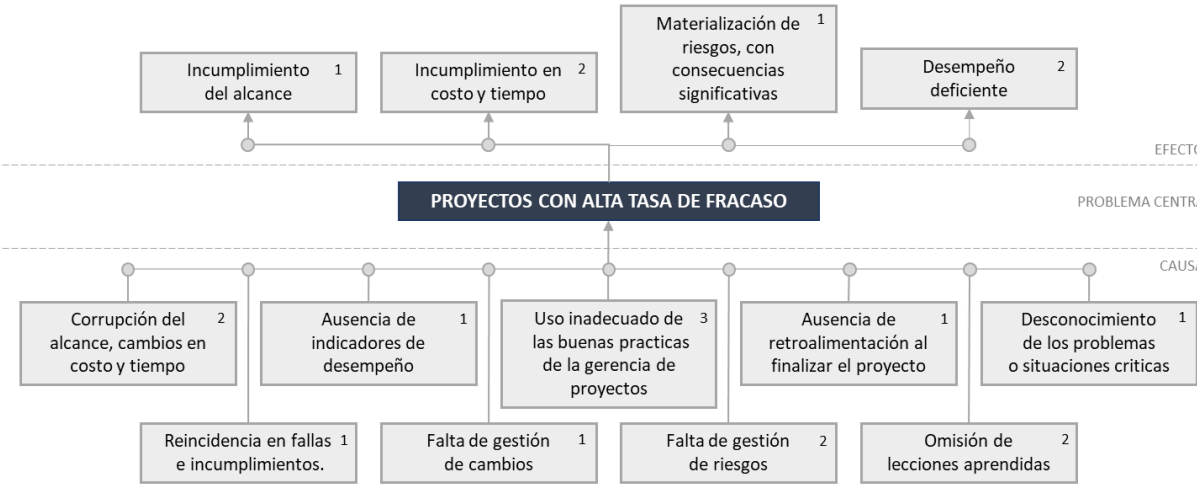
De igual manera, el Project Management Institute (PMI) en su informe anual del 2017, pública que el 14% de los proyectos fallaron o fracasaron; sin embargo, este valor solo representa los proyectos con fallas totales, de los proyectos que no fracasaron completamente, el 31% no cumplió con sus objetivos, el 43% excedió sus presupuestos iniciales y el 49% terminaron tarde (Project Management Institute PMI®, 2017). Un año más tarde, el Informe para el año 2018, revela que aumentaron al 15% los proyectos que fallaron o fracasaron completamente, mientras que los proyectos que no cumplieron sus objetivos o excedieron sus presupuestos iniciales mantuvieron los porcentajes de 31% y 43% respectivamente, pero aumentaron a un 52% los proyectos que terminaron tarde (Project Management Institute PMI®, 2018).

No obstante, Project Management Institute incluye en los resultados de su informe del año 2019, empresas que han estado implementando prácticas, en busca de mejores resultados en sus proyectos, estas empresas se denominan “PMTQ Innovators” mientras que las que siguen rezagadas “PMTQ Laggards”. Para estas últimas, los proyectos que fallaron o fracasaron en su totalidad superaban el 18% y los que no cumplieron con sus objetivos el 39%, sin embargo, disminuyeron su porcentaje aquellos proyectos que excedieron sus presupuestos iniciales (42%) en 1 punto porcentual y en 6 puntos porcentuales los proyectos que terminaron tarde (46%) (Project Management Institute PMI®, 2019).

Por otra parte, para Gartner, Inc. (Compañía de consultoría en TI) el aprendizaje profundo representa el principal impulsor de la inteligencia artificial (IA), puesto que ofrece capacidades superiores de fusión, combinación y análisis de datos en comparación con otras líneas de ML; por lo cual, pronostica que será un impulsor crítico para predicciones de demanda, fraude y fallas, con un rendimiento superior a los de su clase (Laurence Goasduff, 2017). Lo anterior, conduce a formular la siguiente pregunta: ¿Es posible, que mediante el uso de la funcionalidad de las herramientas disponibles en la actualidad de Machine Learning o aprendizaje automatizado y recopilando como insumo una data del mundo real, se logre diseñar un modelo de diagnóstico para los gerentes del proyecto en el cual se prediga la tasa de éxito de un proyecto?

Figura 4

Árbol de problemas.



1 - (Project Management Institute PMI®, 2017)
2 - (Project Management Institute PMI®, 2018)
3 - (Project Management Institute PMI®, 2019)

1.3 Objetivos

Para el desarrollo del Trabajo de Grado, se presentan un objetivo general y los objetivos específicos.

1.4 Objetivo general

Diseñar un modelo de evaluación de proyectos a partir de herramientas y algoritmos existentes de Machine Learning o aprendizaje automatizado, que permita predecir el índice de éxito de los mismos.

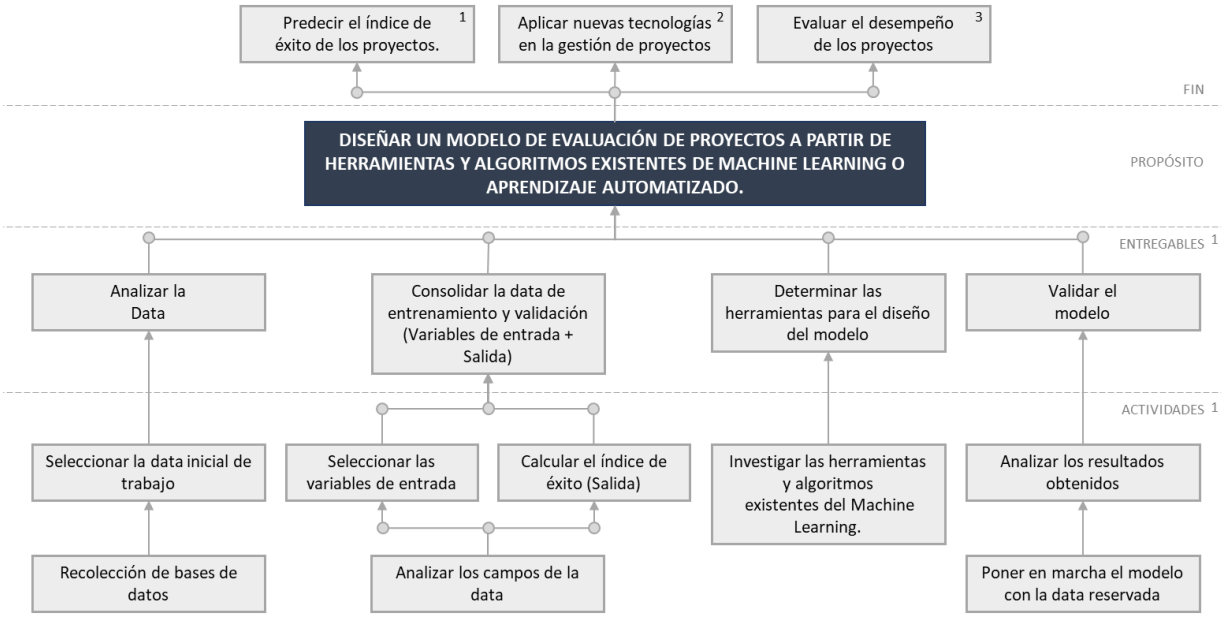
1.5 Objetivos específicos

- Analizar la data inicial, de manera que se asegure la eficacia de la información.
- Consolidar la data de entrenamiento y validación del modelo, incluyendo variables de entrada y cálculo de índice de éxito o salida.

- Determinar la alternativa más adecuada entre las herramientas y algoritmos existentes del Machine Learning, de acuerdo con el tamaño y la calidad de la muestra de la data obtenida.
- Validar el modelo con la data reservada para tal fin y el análisis correspondiente.

Figura 5

Árbol de objetivos.



1 - (Khepri, 2018)
2 - (Project Management Institute PMI®, 2019)
3 - (Project Management Institute PMI®, 2018)

2 DISEÑO METODOLÓGICO

Este trabajo de grado se desarrolla a partir de proyectos de desarrollo de software como unidad de análisis, con lo cual se pretende analizar los aspectos relacionados al desempeño de proyectos y su evaluación; en concordancia, se enfocará en una investigación aplicada cuyo objeto es resolver las malas prácticas de la gestión de proyectos enmarcados en la evaluación de desempeño de los proyectos.

La fase preliminar, está enfocada en la investigación a partir de literatura e informes publicados por organizaciones cuya trayectoria ha permitido obtener estadísticas de porcentajes de éxito y fracaso de proyectos, para esto se hará uso de bases de datos como Scopus y Mendeley, para identificar diferentes enfoques y objetivos en el campo de la Inteligencia Artificial aplicada a la gestión de proyectos, de tal manera, que se obtenga como resultado la justificación y los objetivos de este trabajo.

2.1 Marco Conceptual

La fase inicial, se basa en la investigación estructurada de herramientas de Machine Learning o Aprendizaje Automatizado, con lo cual se identifica la aplicación de algoritmos y herramientas de Inteligencia Artificial; de tal manera, se selecciona la mejor opción para diseñar el modelo.

- **Inteligencia Artificial.** Cualquier técnica que permita a las computadoras imitar la inteligencia humana.
- **Machine Learning.** un subconjunto de inteligencia artificial que incluye técnicas estadísticas profundas, que permitan a las máquinas mejorar en tareas específicas con experiencia.

- **Deep Learning.** Subconjunto de ML compuesto por algoritmos que permiten que el software se entrene a sí mismo para realizar tareas, como el reconocimiento de voz e imagen, al exponer redes neuronales de múltiples capas a grandes cantidades de datos.

2.2 RECOPILACIÓN Y ESTUDIO DE DATOS

La segunda fase, como su nombre lo indica consiste en la recopilación y estudio de los datos históricos, de proyectos ejecutados en Colombia por entidades públicas y privadas, cuyo fin es obtener información verídica y eficaz para evaluar el desempeño de los proyectos. Dicho proceso se lleva a cabo en 3 subfases.

- **Recolección de datos.** Se basa en la búsqueda de Fuentes o Bases de Datos de entidades públicas o privadas con el histórico de proyectos ejecutados en Colombia, que determina el sector de estudio a emplear.
- **Depuración de datos.** Una vez se establece el sector de estudio, se disponen los criterios de verificación y depuración de la data disponible.
- **Análisis de datos.** Por último, se analiza cada uno de los campos que incluye la data para cada proyecto, y se hace una selección previa de los campos que se infiere que pueden llegar a contribuir en el diseño del modelo.

  Selección de variables de entrada. En primer lugar, se analiza y determina cada uno de los campos que son considerados como las variables de entrada del modelo y posteriormente se normalizan los campos.

  Construcción de índice de éxito (salida). Por otro lado, de acuerdo con el análisis de datos de la segunda fase, se establecen las fórmulas a usar en la construcción y son seleccionados aquellos datos que son necesarios para la construcción del índice.

2.3 Diseño del Modelo

En la tercera fase, se realiza la consolidación de datos y el diseño del modelo, teniendo en cuenta la información y datos recopilados en las fases previas, esta implica:

- **Preparación de datos.** Esta subfase recopila los primeros pasos del modelo, como lo son la lectura del conjunto de datos obtenidos en fases previas, distribución y normalización de datos.
- **Desarrollo del modelo.** Una vez, se preparan los datos en el modelo, se procede a definir el modelo su configuración y entrenamiento, esta es una fase determinante en el diseño del modelo.
- **Entrenamiento del Modelo**. Finalmente, se procede con el proceso de entrenamiento del modelo, y la evaluación y predicción del mismo con la data destinada para este proceso.

2.4 Evaluación del Modelo

En la última fase, se pretende realizar la carga de la data de prueba reservada en la anterior fase, de tal manera que se determine la precisión de la predicción del modelo, para lo cual es necesario llevar a cabo los siguientes subfases:

- **Inferencia o predicción data de prueba.** Este contempla la predicción de la salida para el conjunto de datos de prueba.
- **Evaluación del modelo con la data de prueba.** A modo de cierre, esta subfase indica el porcentaje de precisión del modelo, como resultado de la predicción de la data de prueba.

3 MARCO CONCEPTUAL

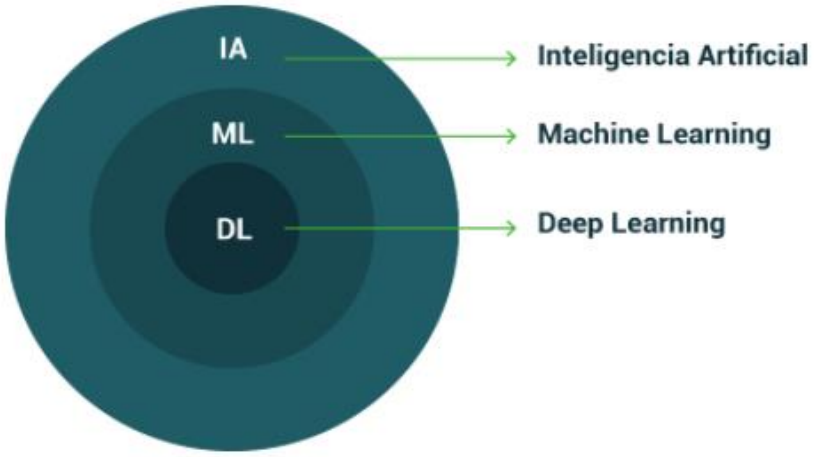
En el Marco Conceptual están las concepciones primordiales de la inteligencia artificial y sus tipos, la definición de Machine Learning y de Redes Neuronales y Deep Learning, esta última compuesta por las redes neuronales artificiales y redes profundas básicas.

3.1 Inteligencia Artificial

El concepto de Inteligencia Artificial (IA), surge inicialmente como una corriente académica para simular por medio de redes neuronales modelos de conocimiento, que puedan dar una respuesta de forma similar a cómo lo hace el cerebro humano.

Figura 6

Diagrama de relación.



Nota. Diagrama de relación entre Inteligencia Artificial, Machine Learning y Deep. Tomado de https://aml.stradata.co/inteligencia-artificial-machine-learning-y-deep-learning/

El núcleo de la IA es el denominado Deep Learning o Aprendizaje Profundo, que agrupa las técnicas del Machine Learning o Aprendizaje Automático, basadas en modelos de redes neuronales que tienen aplicabilidad principalmente en la academia y de investigación, pero que

han incursionado en la aplicabilidad a nivel global en la mayoría de los electrodomésticos, vehículos, consolas de video, aplicaciones de smartphone y en general, en todos los dispositivos. En la **Figura 6** se observa la relación que existe, donde el Deep Learning es un subconjunto del Machine Learning, que a su vez es una parte de la Inteligencia Artificial.

Se conocen avances en 3 áreas, que cuentan con una amplia gama de tecnologías usadas en la vida cotidiana, dentro de los cuales sobresalen:

- El reconocimiento de Voz Automático (Automated Speech Recognition - ASR) y la transcripción de texto, agentes que reciben instrucciones por voz y que tienden a eliminar los accesos por teclado.
- Procesamiento de Lenguaje Natural (Natural Language Processing - NLP)
- La aplicación más utilizada Google Translate, permite generar una traducción automática a varios idiomas. Sin embargo, existe una mayor relevancia en proyectos como el GPT-2, un generador de textos con IA, cuya liberación estuvo pospuesta, al ser considerada una poderosa herramienta para generar noticias falsas (Open AI, 2019).
- Visión por Computador (Computer Vision - CV), se han desarrollado aplicaciones para el reconocimiento de imágenes, que pueden ser usados en softwares de generación de rostros artificiales con alta aplicación en aplicaciones médicas y de seguridad para autenticación.

Dichos avances, han labrado mejoras en el campo de la robótica para la construcción de drones y automóviles sin conductor (autónomos) o que disponen de parqueo asistido, los cuales siguen avanzando gracias a los avances del Deep Learning.

Algunos de los CEOs y filántropos coinciden en que la IA tiene un impacto comparable con el fuego, la electricidad, y a finales del siglo pasado, con el Internet, y que, usándolos en un

sentido éticamente correcto, traerán beneficios para la humanidad tan importantes como la generación de vacunas y por qué no, la realidad virtual. En contraste, con sus opositores, quienes afirman que esta tecnología va a reemplazar la mano de obra humana y su consecuente eliminación de puestos de trabajo.

La Inteligencia Artificial abarca muchas áreas del conocimiento relacionadas con el aprendizaje automático, que van de la mano con el escalamiento computacional constante que ha venido entregando la industria del hardware, las herramientas matemáticas y una robusta comunidad de investigación; con resultados difundidos, que van desde el famoso encuentro en 1997 entre Deep Blue y el GM de ajedrez Gary Kasparov, dejando como vencedora a la máquina IBM, hasta el grupo de investigación de DeepMind, donde lograron desarrollar la máquina AlphaGo que venció a experimentados jugadores en el juego complejo Go (MIT, 2017).

3.1.1 Débil (Artificial Narrow Intelligence)

Existen dos clases de Inteligencia Artificial, en este tipo se incluyen los avances y aplicaciones revisadas anteriormente, relacionadas con el reconocimiento de voz automática, procesamiento de lenguaje natural, visión por computador y las aplicaciones e investigaciones del ámbito académico. Muchos coinciden en que la IA débil es la base de la IA fuerte, y es por esto que se reconocen mayores avances en ésta; sin embargo, algunos expertos mencionan que la IA débil terminará sirviendo como asistente para el ser humano en muchas de las actividades que cotidianamente realiza y, por ende, el abuso de este apoyo puede llevar a que el humano desaprenda ciertas actividades y con esto el atrofiamiento de las partes o músculos que se dejan de usar.

3.1.2 Fuerte (Artificial General Intelligence)

Esta incluye los avances y aplicaciones en máquinas o robots que están en capacidad de realizar las mismas actividades que el ser humano, o incluso, mejor y más rápido; es decir, máquinas con inteligencia que reemplacen al ser humano. Sin embargo, a pesar de que en la actualidad existen robots, que pueden mantener una conversación con un humano sobre temas en los cuales ha sido entrenada, aún está lejos la posibilidad de tener una máquina que cuente con las habilidades cognitivas normales y autonomía propia del ser humano. Por esa razón, a pesar de los grandes avances en la IA débil, se infiere que en la medida que se avanza en la débil la fuerte también lo hace.

3.1.3 Machine Learning

Es el subconjunto de la Inteligencia Artificial que proporciona a los computadores la capacidad de aprender sin ser explícitamente programados, es decir, sin que un programador proporcione las reglas para producir un resultado. Consiste en desarrollar para cada problema un algoritmo de predicción para cada uso o problema, estos algoritmos aprenden de los datos introducidos con el fin de encontrar patrones o tendencias, para analizarlos y construir un modelo para realizar la clasificación y predicción de estos elementos, cada uno de estos modelos utiliza una estructura algorítmica diferente para optimizar las predicciones basadas en los datos introducidos, estos algoritmos se pueden clasificar en 3 grandes categorías: aprendizaje supervisado (supervised learning), aprendizaje no supervisado (unsupervised learning) y aprendizaje por refuerzo (reinforcement learning).

El aprendizaje supervisado (supervised learning) se caracteriza por que los datos de entrenamiento de los modelos incluyen la solución deseada, llamada etiqueta (label), esto implica que un modelo aprenda a partir de una función que mapea una entrada a una salida; una vez entrenado el modelo, el mismo determina correctamente las etiquetas para los datos

no introducidos en el entrenamiento. Este modelo se construye con un algoritmo que iterativamente va afinado el modelo mediante predicciones sobre los datos utilizados en el entrenamiento y va comparando las predicciones con la respuesta correcta que el modelo conoce.

Dentro del aprendizaje supervisado se puede incluir una subclase como el aprendizaje auto supervisado (self-supervised learning), donde las etiquetas no son determinadas por humanos, sino por otros datos o heurísticas. En contraste, el aprendizaje no supervisado (unsupervised learning) es aquel cuyos modelos a entrenar no incluyen las etiquetas y, por lo tanto, el algoritmo es el que debe clasificar la información por sí mismo.

Se habla del aprendizaje por refuerzo (reinforcement learning), cuando el modelo se configura en forma de un agente que debe explorar un universo desconocido y debe determinar las acciones que debe predecir mediante prueba y error, por medio de unas calificaciones (positivas o negativas) que se dan a las acciones que realiza; el agente debe determinar la mejor ruta o estrategia, es decir, la que entregue la mejor calificación. Estos algoritmos combinados con el Deep Learning, son los utilizados en las áreas de reconocimiento de imágenes, automóviles autónomos o modelos para competir en juegos como Go o Startcraft.

3.2 Redes Neuronales y Deep Learning

Como se menciona al inicio de esta sección, el Deep Learning está inmerso en los avances en las tres áreas del Reconocimiento de Voz Automática (Automated Speech Recognition - ASR), el Procesamiento de Lenguaje Natural (Natural Language Processing - NLP) y la Visión por Computador (Computer Vision - CV), basados fundamentalmente en el desarrollo de modelos que simulan el funcionamiento del cerebro humano y específicamente, el funcionamiento de sus células: las neuronas. La agrupación de esta estructura en diferentes capas, son los modelos denominados como Redes Neuronales.

3.2.1 Redes neuronales artificiales.

Se define como la aproximación del Machine Learning, para imitar la actividad en capas de las neuronas de la neocorteza del cerebro humano, donde sucede el pensamiento, estas aprenden niveles de representación, abstracción y estructuras jerárquicas para determinar patrones de datos de diferentes fuentes como lo son el texto y las imágenes traducidas en un formato digital.

En otras palabras, es partir de la capacidad de aprender automáticamente la representación de las características en varios niveles de abstracción, por intermedio de funciones complejas configuradas desde el espacio de entrada al espacio de salida. En últimas, una neurona tiene una o más entradas y una salida, dependiendo del valor de las entradas, la neurona puede “determinar” la salida, con base a unas funciones de activación, ; es por esto, que la salida generalmente es un resultado binario: es o no es, por ejemplo: Una imagen digitalizada es un perro o es un gato, o, la palabra incluida es “uno” o es “dos”, o, la reproducción del sonido equivale a una “a” o a una “e”, o en este caso, un proyecto es exitoso o no lo es.

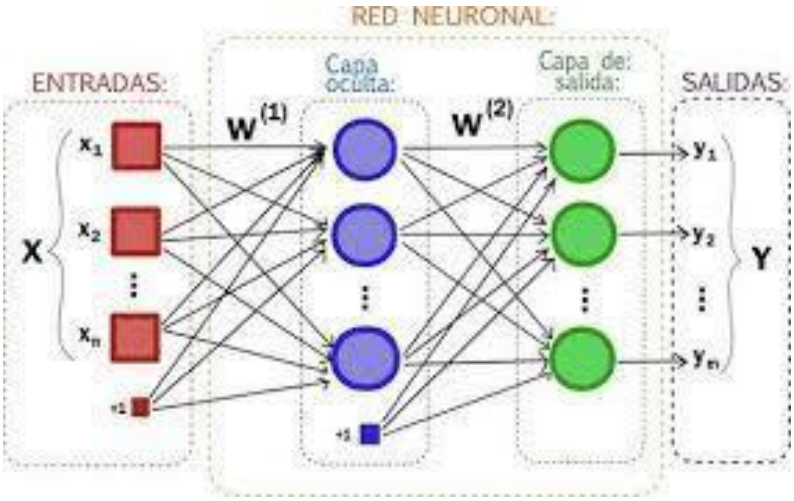
En el Deep Learning, las estructuras algorítmicas permiten modelos que están compuestos de múltiples capas de neuronas artificiales, para aprender representaciones de datos, con múltiples niveles de abstracción que realizan una serie de transformaciones lineales y no lineales para que, a partir de un conjunto de datos de entrada, generen una salida aproximada a la esperada. En el caso del aprendizaje supervisado, consisten en obtener parámetros de transformaciones cuyos resultados sean óptimos, es decir, que la diferencia entre el valor obtenido y el esperado, sea mínimo.

La **Figura 7** representa una red neuronal artificial con cuatro capas: Una capa de entrada (input layer) que recibe los datos de entrada, una(s) capa(s) intermedia(s) u oculta(s)

(hidden layers), una capa de neuronas de salida que procesan los resultados de la capa anterior y una capa de salida (output layer) que retorna la predicción realizada.

Figura 7

Red Neuronal Artificial con cuatro capas.



Nota. Tomado de https://sites.google.com/site/mayinteligenciartificial/unidad-4-redes-neuronales

Actualmente se manejan redes neuronales artificiales con múltiples capas, compuestas de n número de neuronas, que están apiladas una encima de la otra, y por ello se constituye el concepto de profundidad de la red (Deep), cada una, con sus respectivos parámetros que realizan una transformación simple de los datos que reciben de las neuronas de la capa anterior, para procesarlos y transferir sus resultados a la siguiente capa. La unión de todas las capas permite determinar patrones y tendencias a partir de los datos incluidos en la capa de entrada.

El proceso de configuración y parametrización de cada capa define el desafío de cada modelo y constituye el paradigma del Deep Learning, cada modelo tiene una configuración y determinación de número de capas, neuronas y profundidad.

3.2.2 Las Redes Profundas Básicas

En la práctica, todos los algoritmos del Deep Learning son redes neuronales que comparten algunas propiedades básicas comunes que consisten en un número de neuronas interconectadas en diferentes capas, su diferencia radica en la arquitectura de la red, o cómo se encuentran interconectadas en la red, y la forma en cómo se entrenan. A continuación, se listan los diferentes tipos de algoritmos aplicados a las redes neuronales actuales:

- **Perceptrón multi capa (Multi-Layer perceptron- MLP):** Tipo de red neuronales con capas densamente conectadas
- **Redes neuronales convolucionales (Convolutional Neural Networks – CNN):** Tipo de red neuronal con varios tipos de capas especiales, usadas principalmente en la visión por computador.
- **Redes neuronales recurrentes (Recurrent Neural Networks – RNN):** Tipo de red neuronal con una memoria interna, que se crea con los datos de entrada ya conocidos por la red. La salida de una red de este tipo es una combinación de su memoria interna y los datos de entrada. En la medida que van entrando nuevos datos, su memoria se va a actualizando. Este tipo de redes son candidatas preferentes para tareas que funcionan con datos secuenciales, como datos de tipo texto, y datos de series de tiempo (Goodfellow, Bengio, & Corville, 2016).

4 RECOLECCIÓN Y ESTUDIO DE DATOS

El capítulo de Recolección de datos contiene el proceso de búsqueda de las fuentes de información de datos a partir de la base de información del SECOP, por medio de la plataforma del Gobierno Nacional “Datos Abiertos”, la depuración de los datos y el análisis de datos el cual conformado por la selección de las variables de entrada y la construcción de la salida que es el índice de éxito

4.1 Recolección de Datos

Este proceso se inició con la búsqueda de fuentes que contarán con un histórico de proyectos ejecutados en los últimos 10 años, búsqueda que condujo a la plataforma de contratación de proyectos de carácter público, SECOP – Sistema Electrónico para la Contratación Pública.

SECOP, es la plataforma mediante la cual Entidades Estatales llevan a cabo procesos de contratación con cargo a recursos y dineros públicos, esta contempla la totalidad del proceso desde su planeación hasta su liquidación; en donde, para 2016 se habían publicado más de 1.023.981 procesos cuyo valor ascendía a los $83.747 mil millones, en diferentes sectores económicos (Colombia Compra Eficiente, 2020). Puesto que esta información es de carácter público, la plataforma Datos Abiertos da acceso a las bases de datos en formatos que permiten su uso sin ninguna restricción legal (Ministerio de Tecnologías de la Información y las Comunicaciones, 2021).

Como resultado, se selecciona esta plataforma como el medio para la obtención de la data teniendo en cuenta que se obtiene un número significativo de datos y que adicional a esto son proyectos reales; así mismo, se determina que a pesar de que el modelo está enfocado a proyectos en general, es necesario definir un sector de estudio, que para este caso será el

sector de la construcción. Cabe resaltar que esta decisión se basa en el aumento en el aporte del sector al PIB de Colombia como se evidencia en la **Figura 8**, lo que lleva a pensar que al lograr predecir el índice de éxito de los proyectos no solo contribuiría a la gestión de los mismos, sino que puede, llegar a aportar a la economía y a la obra civil e infraestructura pública del país.

Tabla 1

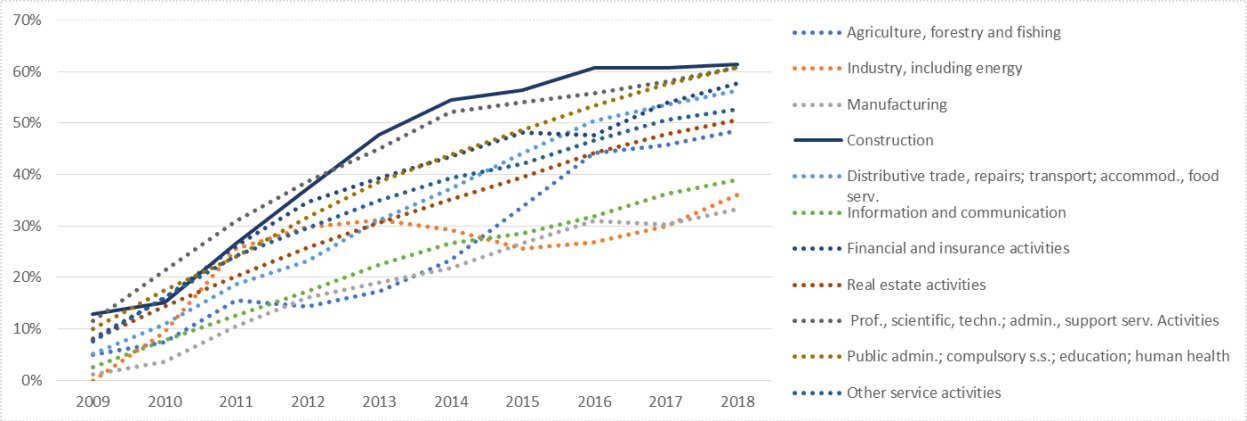
PBI Colombia por sectores económicos.

	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
B1GVA: Agriculture, forestry and fishing (ISIC rev4)	31,869	33,554	34,411	37,709	37,209	38,509	41,555	48,124	57,065	58,815	61,974
B1GVB_E: Industry, including energy (ISIC rev4)	127,834	127,708	140,903	171,686	182,265	185,528	180,824	172,015	174,849	182,349	200,018
B1GVB_E: Industry, including energy (ISIC rev4) B1GVC: of which: Manufacturing (ISIC rev4)	73,264	74,133	75,989	81,820	87,241	90,421	93,603	99,789	106,226	105,022	109,785
B1GVF: Construction (ISIC rev4)	25,288	29,019	29,776	34,462	40,385	48,320	55,568	58,042	64,325	64,474	65,509
B1GVG_I: Distributive trade, repairs; transport; accommod., food serv. (ISIC rev4)	75,636	79,703	84,926	93,024	98,508	109,807	120,677	135,429	152,684	163,052	173,447
B1GVJ: Information and communication (ISIC rev4)	17,124	17,580	18,572	19,612	20,702	22,092	23,336	23,961	25,122	26,821	28,070
B1GVK: Financial and insurance activities (ISIC rev4)	17,984	19,552	21,396	24,357	27,533	29,619	31,839	34,696	34,324	39,060	42,552
B1GVL: Real estate activities (ISIC rev4)	42,236	45,858	49,355	52,966	56,928	60,922	65,194	69,825	75,645	80,976	85,547
B1GVM_N: Prof., scientific, techn.; admin., support serv. activities (ISIC rev4)	26,411	29,840	33,567	38,275	43,078	48,022	55,216	57,392	59,643	62,898	67,522
B1GVO_Q: Public admin.; compulsory s.s.; education; human health (ISIC rev4)	57,489	63,821	69,620	75,682	84,164	93,615	102,459	112,077	123,511	135,235	146,959
B1GVR_U: Other service activities (ISIC rev4)	10,983	11,888	13,087	14,510	15,586	16,900	18,084	18,982	20,551	22,226	23,192

Nota. Adaptado de https://stats.oecd.org/viewhtml.aspx?datasetcode=SNA_TABLE1&lang=en (Organisation for Economic Co-Operation and Development, 2019).

Figura 8

Variación del aporte al PBI por sector económico desde el año 2008.



Nota. Figura de la variación del aporte al PBI por sector económico tomando como punto de partida el año 2008.

En concordancia, en la siguiente subfase se describe cada uno de los criterios usados para depurar la data inicial respecto a la establecido anteriormente.

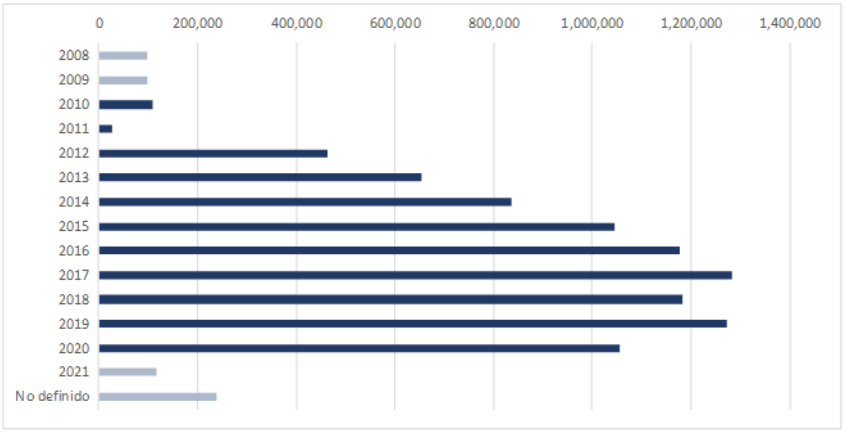
4.2 Depuración de Datos

La data recolectada, cuenta con más de nueve millones de procesos de contratación en Colombia, por lo cual el primer criterio para depurar es seleccionar los procesos cargados en la plataforma en los últimos 10 años; dicho criterio se establece teniendo en cuenta que la data perteneciente a los años 2008 y 2009 no contiene la información suficiente, y 355,525 procesos no tienen definido el año de cargue en la plataforma o el año es posterior a la fecha de recolección de la data.

Figura 9

Distribución de Data por Año de Cargue SECOP.

Anno Cargue SECOP	
2008	99,764
2009	98,531
2010	108,973
2011	26,441
2012	464,643
2013	654,324
2014	837,042
2015	1,045,435
2016	1,175,964
2017	1,282,875
2018	1,181,364
2019	1,272,663
2020	1,056,866
2021	117,556
No definido	237,969
Total	**9,660,410**



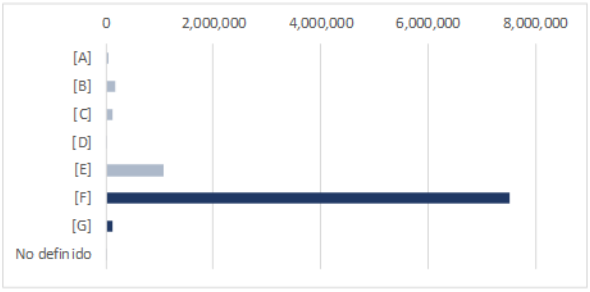
Nota. Figura de la distribución de la data por año de cargue en SECOP, los datos sombreados son los seleccionados.

Una vez realizado el primer filtro, se opta por depurar la data de acuerdo con el grupo y objeto a contratar; para este campo se determina cuáles son los grupos y objetos por contratar que están alineados con el sector de estudio seleccionado.

Figura 10

Distribución de Data por Grupo.

Grupo		
[A]	Material Vivo Animal y Vegetal	35,677
[B]	Materias Primas	161,707
[C]	Maquinaria, Herramientas, Equipo Industrial y Vehículos	131,865
[D]	Componentes y Suministros	30,046
[E]	Productos de Uso Final	1,083,666
[F]	Servicios	7,514,692
[G]	Terrenos, Edificios, Estructuras y vías	116,497
No definido		32,440
Total		**9,106,590**

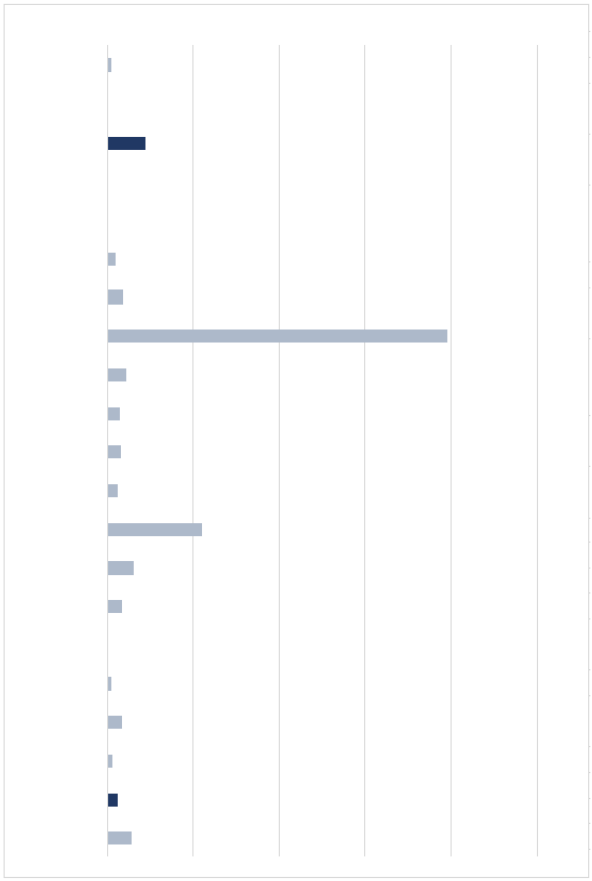


Nota. Figura de la distribución de la data por grupo, los datos sombreados son los seleccionados.

Figura 11

Distribución de Data por Objeto a Contratar.

Objeto a Contratar		
70000000	Servicios de Contratacion Agrícola,	48,977
71000000	Servicios de Minería, Petróleo y Gas	6,023
72000000	Servicios de Edificación, Construcción de Instalaciones y Mantenimiento	440,173
73000000	Servicios de Producción Industrial y Manufactura	13,266
76000000	Servicios de Limpieza, Descontaminación y Tratamiento de Residuos	1,002
77000000	Servicios Medioambientales	89,987
78000000	Servicios de Transporte, Almacenaje y Correo	193,653
80000000	Servicios de Gestion, Servicios Profesionales de Empresa y Servicios Administrativos	3,944,153
81000000	Servicios Basados en Ingeniería, Investigación y Tecnología	221,297
82000000	Servicios Editoriales, de Diseño, de Artes Graficas y Bellas Artes	142,669
83000000	Servicios Públicos y Servicios	152,869
84000000	Servicios Financieros y de Seguros	116,093
85000000	Servicios de Salud	1,088,162
86000000	Servicios Educativos y de Formación	319,778
90000000	Servicios de Viajes, Alimentación, Alojamiento y Entretenimiento	165,367
91000000	Servicios Personales y Domésticos	19,519
92000000	Servicios de Defensa Nacional, Orden Publico, Seguridad y Vigilancia	44,669
93000000	Servicios Políticos y de Asuntos Cívicos	171,925
94000000	Organizaciones y Clubes	60,241
95000000	Terrenos, Edificios, Estructuras y Vías	116,497
No definido		274,869
Total		**7,631,189**



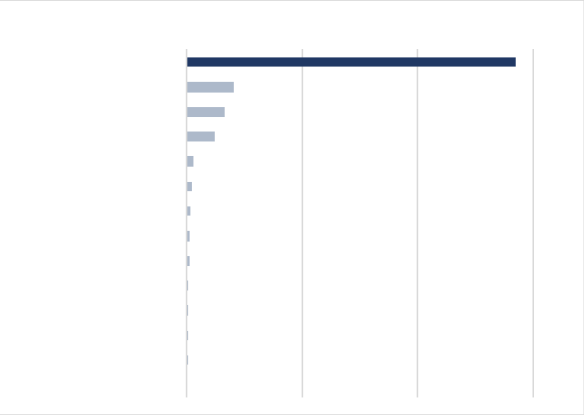
Nota. Figura de la distribución de la data por objeto a contratar, los datos sombreados son los seleccionados.

En esta etapa, se consolida una data con 556.670 procesos, no obstante, al realizar la distribución por tipo de contrato se evidencia que de ellos 140 no tienen definido este campo, por lo cual estos datos se desechan, y posteriormente se depura tomando como factor determinante la cuantía total de los procesos por tipo.

Figura 12

Distribución de Data por Tipo de Contrato.

Tipo de Contrato		
Obra	281,833	284,478,121
Otro Tipo de Contrato	40,724	40,358,434
Concesión	288	32,239,434
Prestación de Servicios	170,333	23,964,130
Compraventa	6,847	5,219,447
Consultoría	5,906	4,215,259
Interventoría	7,961	2,951,911
Arrendamiento	23,128	2,334,412
Crédito	188	1,387,388
Fiducia	51	1,091,853
Comodato	3,087	1,023,404
Suministro	15,838	810,801
Acuerdo Marco	342	504,628
Agregación de Demanda	4	15
Total	**556,530**	**400,579,236**



Nota. Figura de la distribución de la data por Tipo de Contrato - Recuento y cuantía total de los procesos, los datos sombreados son los seleccionados.

4.3 Análisis de Datos

En esta etapa de la fase de recolección y estudio de datos, se hace un análisis del contenido de cada uno de los campos y se determina si cumple 3 reglas: un 70% de los proyectos posee información, no es una constante y no es un dato variable en cada proyecto.

Tabla 2

Descripción de los campos de información de los procesos de compra pública registrados en la plataforma SECOP.

NOMBRE CAMPO	DESCRIPCIÓN	TIPO	CHECK
UID	Valor compuesto para identificar de manera individual cada registro	Número	X
Anno Cargue SECOP	Año en el que se hizo el registro del proceso en la plataforma	Número	✓
Anno Firma del Contrato	En caso de ser un contrato firmado, la fecha en que esta firma se hizo	Número	✓
Nivel Entidad	Determina el primer grado de caracterización de la entidad de acuerdo con su orden: Nacional o Territorial	Texto simple	✓
Orden Entidad	Detalla el orden de la Entidad, definiendo el tipo de Entidad Nacional o Territorial de acuerdo con su grado de centralización	Texto simple	✓
Nombre de la Entidad	Nombre de la Entidad del estado a la que corresponde el proceso	Texto simple	✓
NIT de la Entidad	NIT de la Entidad, tal como lo registró en la plataforma	Texto simple	✓

NOMBRE CAMPO	DESCRIPCIÓN	TIPO	CHECK
Código de la Entidad	Código de la Entidad, utilizado como identificador único en la plataforma SECOP I	Número	✓
ID Tipo de Proceso	El ID y Tipo de Proceso describen la modalidad a través de la cual se desarrolló el proceso de compra	Número	✓
Tipo de Proceso	El ID y Tipo de Proceso describen la modalidad a través de la cual se desarrolló el proceso de compra	Texto simple	✓
Estado del Proceso	El Estado del proceso a la fecha de publicación	Texto simple	✓
Causal de Otras Formas de Contratación Directa	En caso de ser un proceso desarrollado bajo la modalidad de contratación directa, este campo describe la causa por la cual se determinó ese tipo de contratación	Texto simple	X
ID Régimen de Contratación	ID del régimen bajo el cual la entidad desarrolla el proceso de compra pública	Número	✓
Régimen de Contratación	Descripción del régimen bajo el cual la entidad desarrolla el proceso de compra pública	Texto simple	✓
ID Objeto a Contratar	ID del Objeto de la contratación, basado en el catálogo de bienes y servicios UNSPSC, consultable desde https://www.colombiacompra.gov.co/clasificador-de-bienes-y-servicios	Número	✓
Objeto a Contratar	Descripción del Objeto de la contratación, basado en el catálogo de bienes y servicios UNSPSC, consultable desde https://www.colombiacompra.gov.co/clasificador-de-bienes-y-servicios	Texto simple	✓
Detalle del Objeto a Contratar	Adicional al código que define el objeto del contrato, se registra un detalle de la definición del bien o servicio que se adquirirá dentro del proceso	Texto simple	X
Tipo de Contrato	Tipo de Contrato que se realizará, ejemplos: Fiducia, Obra, entre otros.	Texto simple	X
Municipio Obtención	Municipio en el que se desarrolla el proceso de compra pública	Texto simple	✓
Municipio Entrega	Municipio en el que se hace la entrega del bien o servicio	Texto simple	✓
Municipios Ejecución	Municipios en los que se desarrollará el objeto del proceso de compra pública	Texto simple	✓
Fecha de Cargue en el SECOP	Fecha en la que se hizo el registro en la plataforma	Texto simple	✓
Numero de Constancia	Identificador del proceso de compra, generado por SECOP I	Número	X
Numero de Proceso	Identificador del proceso, de acuerdo con la nomenclatura de la entidad	Texto simple	X
Número del Contrato	Identificador del contrato, de acuerdo con la nomenclatura de la entidad	Texto simple	X
Cuantía Proceso	Valor por el cual se lanza el proceso de compra	Número	✓
ID Grupo	Categorización inicial del bien o servicio definido en el proceso de compra, de acuerdo con sus características principales	Texto simple	✓
Nombre Grupo	Categorización inicial del bien o servicio definido en el proceso de compra, de acuerdo con sus características principales	Texto simple	✓
ID Familia	Segundo nivel de detalle dentro de la caracterización del bien o servicio	Número	✓

NOMBRE CAMPO	DESCRIPCIÓN	TIPO	CHECK
Nombre Familia	Segundo nivel de detalle dentro de la caracterización del bien o servicio	Texto simple	✓
ID Clase	Tercer nivel de detalle dentro de la caracterización del bien o servicio	Número	✓
Nombre Clase	Tercer nivel de detalle dentro de la caracterización del bien o servicio	Texto simple	✓
ID Adjudicación	Identificador de la adjudicación o adjudicaciones hechas en el proceso de compra	Número	X
Tipo Identificación del Contratista	Tipo de Identificación del contratista seleccionado en la adjudicación	Texto simple	✓
Identificación del Contratista	Identificación del contratista seleccionado en la adjudicación	Número	✓
Nombre Raz Social Contratista	Nombre o Razón Social del contratista seleccionado en la adjudicación	Texto simple	✓
Dpto y Municipio Contratista	Departamento y Municipio en el cual opera el contratista seleccionado en la adjudicación	Texto simple	✓
Tipo Documento Representante Legal	En caso de ser una empresa, el tipo de identificación del representante legal de la empresa seleccionada en la adjudicación	Texto simple	✓
Identificación del Representante Legal	En caso de ser una empresa, identificación del representante legal de la empresa seleccionada en la adjudicación	Número	✓
Nombre del Representante Legal	En caso de ser una empresa, Nombre del representante legal de la empresa seleccionada en la adjudicación	Texto simple	✓
Fecha de Firma del Contrato	Fecha en la que se realiza la firma del contrato correspondiente a la adjudicación del registro	Texto simple	✓
Fecha Inicio Ejecución Contrato	Fecha en la que se inicia la ejecución del contrato correspondiente a la adjudicación del registro	Texto simple	✓
Plazo de Ejecución del Contrato	Valor y unidad en la que se mide el tiempo de ejecución del contrato, sean días o meses	Número	✓
Rango de Ejecución del Contrato	Valor y unidad en la que se mide el tiempo de ejecución del contrato, sean días o meses	Texto simple	✓
Tiempo Adiciones en Días	Extensión del contrato, fuera de la definición inicial, en días	Número	✓
Tiempo Adiciones en Meses	Extensión del contrato, fuera de la definición inicial, en meses	Número	✓
Fecha Fin Ejecución Contrato	Fecha de finalización de la ejecución del contrato	Texto simple	✓
Compromiso Presupuestal	En caso de tener registro presupuestal, el campo muestra el código correspondiente	Texto simple	X
Cuantía Contrato	Valor por el cual se firma el contrato	Número	✓
Valor Total de Adiciones	Valor de la suma de las adiciones hechas al contrato	Número	✓
Valor Contrato con Adiciones	Valor total del contrato, incluyendo las adiciones	Número	✓
Objeto del Contrato a la Firma	Objeto del contrato, registrado al momento de la firma	Texto simple	X
ID Origen de los Recursos	Identificador de la forma en que se consiguen los recursos con los que se va a pagar el contrato	Número	✓
Origen de los Recursos	Identificador de la forma en que se consiguen los recursos con los que se va a pagar el contrato	Texto simple	✓

NOMBRE CAMPO	DESCRIPCIÓN	TIPO	CHECK
Código BPIN	En caso de corresponder a un proceso financiado por el Banco de Programas y Proyectos de Inversión Nacional - DNP, aquí se registra el código	Número	X
Proponentes Seleccionados	Listado de los proponentes seleccionados en el proceso de compra	Texto simple	X
Calificación Definitiva	Calificación definitiva de los proponentes dentro del proceso de compra	Texto simple	X
ID Sub Unidad Ejecutora	Identificador y nombre de la Sub Unidad de ejecución de presupuesto asignada al proceso de compra	Texto simple	X
Nombre Sub Unidad Ejecutora	Identificador y nombre de la Sub Unidad de ejecución de presupuesto asignada al proceso de compra	Texto simple	X
Ruta Proceso en SECOP I	Ruta del proceso de compra en SECOP, para hacer consulta de la información detallada	URL del sitio web	X
Moneda	Moneda en la cual están registradas las cuantías del proceso de compra	Texto simple	X
EsPostConflicto	Marcador que indica si el proceso de compra pública está enmarcado en las acciones de postconflicto del acuerdo de paz del año 2017	Texto simple	X
Marcación Adiciones	Marcación de si el proceso tiene adiciones registradas	Texto simple	X
Posición Rubro	Posición (Identificador) del Rubro presupuestal	Texto simple	X
Nombre Rubro	Nombre del Rubro Presupuestal al que corresponde el proceso	Texto simple	X
Valor Rubro	Valor del Rubro presupuestal asignado al proceso	Número	X
Sexo Representante Legal Entidad	Sexo del Representante legal de la entidad compradora	Texto simple	✓
Pilar Acuerdo Paz	Pilar, del acuerdo de paz del año 2016, sobre el cual se desarrolla el proceso de compra	Texto simple	X
Punto Acuerdo Paz	Punto del acuerdo de paz del año 2016, al que se relaciona el objeto de contratación del proceso de compra	Texto simple	X
Municipio Entidad	Municipio en el que se encuentra registrada la entidad estatal compradora	Texto simple	✓
Departamento Entidad	Departamento en el que se encuentra registrada la entidad estatal compradora	Texto simple	✓
Ultima Actualización	Para suplir requerimientos de timestamp, se registra la última fecha de modificación del proceso de compra	Texto simple	X

Nota. Adaptado de (Ministerio de Tecnologías de la Información y las Comunicaciones, 2021).

4.3.1 Selección de variables de entrada.

Para llevar a cabo la selección de estas variables, primero fue necesario definir que el momento del Ciclo de Vida Predictivo del proyecto en el que se debe correr el modelo es antes de la adjudicación del contrato; una vez se define este momento, se hace el análisis de los 72 datos que arroja la data para cada uno de los proyectos, partiendo del hecho de que los datos

que sean seleccionados como variables de entrada deben ser aquellos de los cuales se tiene certeza antes de la adjudicación del contrato.

Como resultado de dicho análisis, se seleccionan 15 variables de entrada datos que son tipificados con los criterios que se determinan en la **Tabla 3**.

Tabla 3

Criterios de tipificación de variables de entrada.

VARIABLE DE ENTRADA	TIPIFICACIÓN DE DATA	TIPO
Código de la Entidad	Se verifica que el código no presente ningún tipo de carácter o espacio.	Número sin decimales
ID Tipo de Proceso	Se verifica que el código no presente ningún tipo de carácter o espacio.	
ID Régimen de Contratación	Se verifica que el código no presente ningún tipo de carácter o espacio.	
ID Objeto a Contratar	Se verifica que el código no presente ningún tipo de carácter o espacio.	
Cuantía Proceso	Se verifica que el código no presente ningún tipo de carácter o espacio.	
ID Familia	Se verifica que el código no presente ningún tipo de carácter o espacio.	
ID Clase	Se verifica que el código no presente ningún tipo de carácter o espacio.	
ID Origen de los Recursos	Se verifica que el código no presente ningún tipo de carácter o espacio.	
Anno Cargue SECOP	Se verifica que el código no presente ningún tipo de carácter o espacio.	
Fecha Cargue SECOP	Se cambia el formato a día-mes-año de carácter numérico.	
ID Orden Entidad	Se verifica que el código no presente ningún tipo de carácter o espacio.	
ID Departamentos Ejecución	Se tipifican los departamentos de ejecución y se les asigna un ID, de acuerdo con el ranking de transparencia.	
ID Departamento Entidad	Se tipifican los departamentos de ejecución y se les asigna un ID, de acuerdo con el ranking de transparencia.	
ID Grupo	Se verifica que el código no presente ningún tipo de carácter o espacio.	
Tiempo planeado	Con los campos de plazo de y rango de ejecución del contrato se unifican los criterios de tiempo a días para todos los proyectos,	

4.3.2 Construcción de índice de éxito (salida).

Para la construcción del índice de éxito (salida), se utilizaron los datos que influyen en términos de costo, tiempo y alcance, para cada factor se tiene los siguientes datos usados:

Tabla 4

Datos usados para determinar tiempo, costo y alcance.

TIPO	DATO 1	DATO 2	DATO 3
Tiempo	Fecha de inicio de contrato	Fecha fin de contrato	Plazo de ejecución del contrato
Costo	Cuantía del Proceso	Valor contrato con adiciones	
Alcance	Estado del Proceso		

En razón a que las herramientas de Machine Learning o Aprendizaje Automatizado trabajan con valores numéricos, las fechas se transformaron a un valor numérico, mientras que el dato del estado del proceso se les asigno un valor numérico para cada estado. Para determinar la salida del tiempo y posterior clasificación en el rango, se usaron las fórmulas de:

Para la duración real en días se tomó la diferencia entre las variables de fecha fin de contrato y fecha de inicio de contrato en valor numérico:

$$Duración\ Real = Fecha\ fin\ de\ contrato - Fecha\ inicio\ de\ contrato \quad (1)$$

Para la duración planeada en días se tomó la variable de plazo de ejecución del contrato en días, es decir, en caso de que la variable estuviera en meses o años se realizó la conversión a días.

$$Duración\ planeada = \ \ Plazo\ de\ ejecución\ del\ contrato \quad (2)$$

Con base al resultado de la Ecuación No. 1 y Ecuación No.2, se tiene la Ecuación No.3, la cual determina la fórmula de tiempo en la variación de la duración planeada y duración real.

$$Formula\ tiempo = \frac{Duración\ planeada - Duración\ Real}{Duración\ planeada} \quad (3)$$

Para determinar la salida del costo y posterior clasificación en el rango, se usó la fórmula de variación entre el valor de la cuantía del proceso y el valor total con adiciones acorde con la Ecuación No.4:

$$Formula\ costo = \frac{Valor\ contrato\ con\ adiciones - Cuantía\ del\ Proceso}{Valor\ contrato\ con\ adiciones} \quad (4)$$

Para determinar la salida de alcance se tomó la variable de estado del proceso, debido a que el estado del proceso es en formato texto, a cada estado se le asigno un ID.

Tabla 5

ID del Estado del Proceso.

ESTADO DEL PROCESO	ID DEL ESTADO DEL PROCESO
Liquidado	1
Celebrado	2
Terminado sin liquidar	3

Dependiendo del valor numérico positivo o negativo de las salidas para el tiempo, costo y alcance se asignaron calificaciones de 0 a 1 de acuerdo con el rango de variación de los respectivos valores.

Respecto a las salidas de tiempo y costo, se tiene los siguientes rangos y clasificación:

Tabla 6

Rangos y clasificación de la salida de tiempo y costo.

RANGO	CLASIFICACIÓN
Mayor o igual a 1	0
(0) - (0,25)	1
(0.26) - (0.50)	0.75
(0.51) - (0.75)	0.5
(0.76) - (0.99)	0.25
(0) - (-0.25)	-0.25
(-0.26) - (-0.50)	-0.5
(-0.51) - (-0.75)	-0.75
(-0,76) - (-0,99)	-1
Mayor o igual (-1)	0

Respecto a la salida de alcance, se tiene los siguientes rangos y clasificación:

Tabla 7

Rangos y clasificación de la salida de alcance.

RANGO	CLASIFICACIÓN
1	1
2	0.5
3	0

Con base a la clasificación de tiempo, costo y alcance, para la salida se usa los porcentajes definidos en la **Tabla 8.**

Tabla 8

Porcentajes para las variables de tiempo, costo y alcance.

VARIABLE	PORCENTAJE
Tiempo	40%
Costo	40%
Alcance	20%

Con los porcentajes de las variables, y acorde con el rango se obtiene la calificación del proyecto de 0 a 1, donde 0 es fracaso y 1 es éxito.

Tabla 9

Clasificación del proyecto.

CLASIFICACIÓN	RANGO MENOR	RANGO MAYOR
1.00	0.9	1
0.75	0.76	0.8999
0.50	0.51	0.7599
0.25	0.26	0.5099
0.00	0	0.2599

Finalmente, como resultado de la fase de recolección y estudio de datos, se obtiene un conjunto de datos depurado y tipificado con un numero de variables determinadas y una salida, este se considera como la materia prima de las etapas subsiguientes.

5 DISEÑO DE UN MODELO DE EVALUACIÓN DE PROYECTOS A PARTIR DE HERRAMIENTAS DE MACHINE LEARNING O APRENDIZAJE AUTOMATIZADO.

De acuerdo con el enfoque, aplicaciones y usos de la Inteligencia Artificial y Machine Learning, en los cuales se profundizó en el capítulo del Marco Conceptual, se determina hacer uso de un algoritmo de aprendizaje supervisado de la rama de la Inteligencia Artificial Débil, en razón a que el objetivo de este modelo es predecir en la etapa de planeación si un proyecto va a ser exitoso o no. Se entiende como aprendizaje supervisado, el enfoque sobre modelos que son entrenados a partir de una data histórica de proyectos, compuesta de unas entradas determinadas y la constitución de la salida a través de fórmulas que miden la triple restricción, que, para este caso, corresponde al índice de éxito o fracaso. y posteriormente, al presentarle al modelo las entradas seleccionadas, este predice la salida con base a los datos de entrenamiento.

En concordancia con lo referenciado en el capítulo 2.3 del Diseño Metodológico, el diseño del modelo se compone de las siguientes subfases, que serán profundizadas a continuación:

1. **Preparación de datos.** Lectura del conjunto de datos obtenidos en la fase de recopilación, su correspondiente distribución y normalización.
2. **Desarrollo del modelo.** Definición de la configuración del modelo.
3. **Entrenamiento del Modelo**. Proceso para el aprendizaje que permita obtener la mayor precisión posible en su fase de inferencia o predicción.

5.1 Preparación de Datos

En esta fase, como se definió en el diseño metodológico se dará lugar a la preparación de los datos obtenidos en la fase de recolección y estudio, para ello, se importan en el ambiente de desarrollo todas aquellas librerías que se requieren para el proceso de cargue de archivos, visualización y análisis de los mismos (Ver **Figura 13**).

Figura 13

Importe de las bibliotecas necesarias para la lectura del conjunto de datos desde un archivo Excel o un archivo CSV.

```
In [1]: # 1. Importación de las librerías neceserias para:
        #    lectura del Conjunto de Datos a partir de un Archivo de Excel o CSV para su visualización y análisis
        #    Configurción, entrenamiento , Evaluación y predicción de la Red Neuronal

        import math
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        import keras
        import keras.backend as kb
        import tensorflow as tf
        from datetime import datetime
        from sklearn.metrics import confusion_matrix
```

Nota. Tomada de MEP_20210218.

5.1.1 Lectura de datos

Una vez se selecciona el formato del archivo con la data a utilizar, se usa la librería **Pandas (Pandas)** que permite la manipulación y análisis de datos, para importar el archivo en el modelo; en este caso un archivo CSV, por lo que se utiliza la clase **pandas.read_csv** (**Pandas**), que recibe como parámetro, el nombre del archivo a leer (Ver **Figura 14**).

Figura 14

Lectura del conjunto de datos del archivo CSV

```
In [11]: # Lectura del conjunto de Datos desde un Archivo .CSV
         data_file='20210216_Data_95%_C40_T40_A20_3.csv'
         df_full = pd.read_csv(data_file)
```

Nota. Tomada de MEP_20210218.

Una buena práctica, es darle un vistazo rápido a la estructura de los datos, para lo cual se utiliza la clase **pandas.head**, que lista los primeros 5 registros con el encabezado de cada variable de entrada y/o salida incluidos en el archivo (Ver **Figura 15**).

Figura 15

Verificación del encabezado del conjunto de datos

```
In [11]: # lectura del conjunto de Datos desde un Archivo .CSV
         data_file='20210216_Data_95%_C40_T40_A20_3.csv'
         df_full = pd.read_csv(data_file)
         # Despliegue del encabezado de los variables de Entrada y de Salida del conjunto de Datos a introducir al modelo
         df_full.head()
```

Out[11]:

	Cod_Entidad	ID_Tipo_Proceso	ID_Regimen_Cont	ID_Objeto_a_Contratar	Cuantia_Proceso	ID_Familia	ID_Clase	ID_Origen_Recursos
0	217050011	12	3	72000000	8929900	0	0	0
1	217050011	12	3	72000000	8854060	0	0	0
2	217050011	12	3	72000000	12454000	0	0	0
3	217050011	12	3	72000000	12787000	0	0	0
4	217050011	12	3	72000000	11761100	0	0	0

Nota. Tomada de MEP_20210218.

5.1.2 Separación de datos.

Inicialmente se cuenta con un solo conjunto de datos consolidado, no obstante, para la configuración y evaluación en un modelo en Machine Learning, generalmente se dividen los datos obtenidos o levantados, en 2 conjuntos:

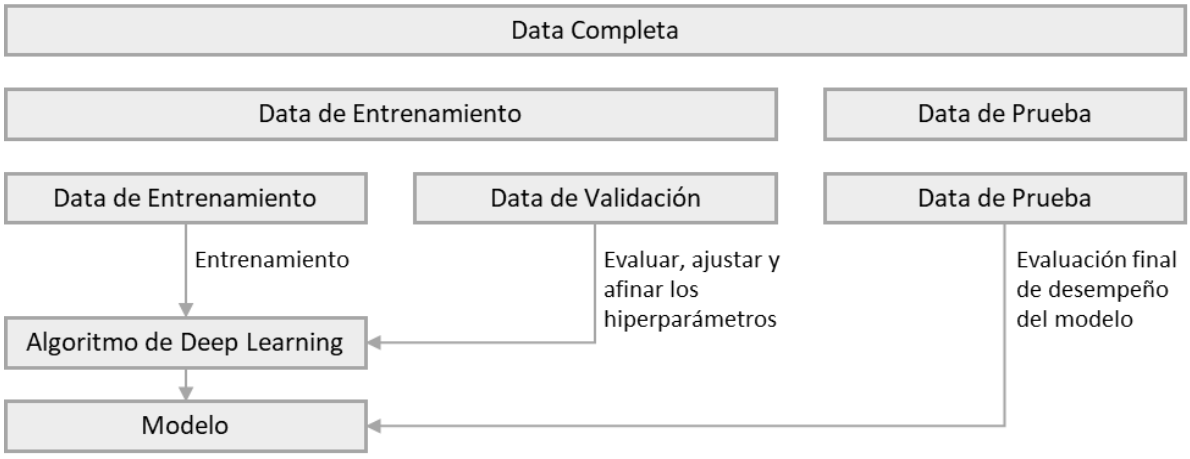
1. **Data de Entrenamiento (Training).** Corresponde al conjunto de datos con los cuales se entrena el modelo; a su vez, una porción de estos datos se reserva como **Data de Validación (V*alidation*).** En este caso la data de entrenamiento corresponde al 95% del conjunto total de datos.
2. **Data de Prueba (Test).** Es el conjunto de datos restante (5%) con el cual se prueba el modelo, a diferencia de la Data de entrenamiento este conjunto de datos no incluye la variable de salida.

En efecto, los datos de entrenamiento que resultan después reservar los de validación y prueba, son aquellos que se usan para que el algoritmo de aprendizaje calcule los parámetros del modelo, mientras que los de validación son los que se encargan de afinar los hiperparámetros; con algunas de las métricas obtenidas del modelo, junto con la precisión (accuracy), que se obtiene del conjunto de datos de validación, se toman decisiones para ajustar los hiperparámetros del algoritmo, antes de repetir el proceso de entrenamiento.

Es de resaltar, que el esfuerzo por mejorar el algoritmo, ajustando los hiperparámetros con respecto al comportamiento del modelo con los datos de validación, puede generar de cierta forma una incidencia en los resultados, a favor del conjunto de datos de validación. Es por esta razón, que se deben reservar unos datos de prueba, un conjunto de datos que el modelo no ha visto ni como conjunto de datos de entrenamiento, ni como conjunto de datos de validación, con lo cual se pretende obtener un comportamiento del algoritmo más objetivo; de tal manera, se garantiza que el modelo predice de forma correcta. En la **Figura 16**, se pretende esquematizar esta separación de conjunto de datos y el propósito de cada uno.

Figura 16

Esquema de distribución del conjunto de datos.



Esta estrategia de evaluación es la más utilizada; sin embargo, existen otras estrategias de separación de conjunto de datos, para validar los modelos cuando se presenta dificultad para obtenerlos, como por ejemplo la validación cruzada (cross-validation), donde se divide los datos en x particiones del mismo tamaño, y para cada partición i, el modelo es entrenado con las restantes x-1 particiones, y evaluando en cada partición i.

Para este modelo se utiliza el método **sample()** proporcionado por la librería Pandas, donde en el parámetro frac se configura con el valor 0.8, para indicar que se quiere separar de la data disponible, un 80% para el conjunto de Datos para entrenamiento y que se almacenan en la variable **train_df** , y el 20% restante de estos datos se destinarán para pruebas, y que serán almacenados en la variable **test_df.** Para ello se utiliza el método **drop()** proporcionado por la librería Pandas.

Figura 17

Separación del conjunto de datos.

```
In [16]: # 3. Separar el conjunto de Datos en datos de Entrenamiento y Pruebas, Extracción y Escalamiento de los datos


         # Separar el conjunto de Datos en datos de Entrenamiento y Pruebas
         train_df = df_full.sample(frac=0.8,random_state=0)
         test_df = df_full.drop(train_df.index)
```

Nota. Tomada de MEP_20210218.

5.1.3 Normalizar datos de Entrada

Una buena práctica es normalizar los datos que se introducirán a la red neuronal, es decir escalar los datos a valores más pequeños, en rangos como [-1, 1] o [0, 1], o si los datos presentan valores más grandes en diferentes valores que 2 binarios, es necesario normalizarlos.

Se pueden inspeccionar los datos de entrada mediante el método **describe()**, el cual ofrece el paquete Pandas para obtener datos estadísticos como el número de valores,

medianas y desviaciones, de cada una de las variables incluidas en el archivo; para desplegar esta información estadística en el eje horizontal, y los datos de las variables en el eje vertical, se requiere utilizar el método **transpose()**, como se evidencia en la **Figura 18**.

Figura 18

Visualización del conjunto de datos y métricas estadísticas.

```
In [12]: # 2. Visualización del conjunto de Datos y métricas estadísticas
         train_stats = df_full.describe()
         train_stats = train_stats.transpose()
         train_stats
```

Out[12]:

	count	mean	std	min	25%	50%	75%	max
Cod_Entidad	217589.0	2.138353e+08	6.543397e+07	1002001.0	205670011.0	223675011.0	254000022.0	5.130002e+08
ID_Tipo_Proceso	217589.0	9.955292e+00	4.650524e+00	1.0	11.0	12.0	13.0	2.200000e+01
ID_Regimen_Cont	217589.0	2.893933e+00	3.081776e-01	1.0	3.0	3.0	3.0	3.000000e+00
ID_Objeto_a_Contratar	217589.0	7.691491e+07	9.427992e+06	72000000.0	72000000.0	72000000.0	72000000.0	9.500000e+07
Cuantia_Proceso	217589.0	6.638612e+08	1.057117e+10	0.0	15000000.0	37000000.0	200000000.0	3.480000e+12
ID_Familia	217589.0	3.359240e+03	3.940744e+03	0.0	0.0	0.0	7211.0	9.514000e+03
ID_Clase	217589.0	3.359325e+05	3.940839e+05	0.0	0.0	0.0	721110.0	9.514190e+05
ID_Origen_Recursos	217589.0	4.181094e-01	1.257136e+00	0.0	0.0	0.0	0.0	6.000000e+00
Anno_cargue_SECOP	217589.0	2.014578e+03	2.461702e+00	2010.0	2013.0	2014.0	2017.0	2.019000e+03
Fecha_Cargue_SECOP3	217589.0	4.206019e+04	8.927559e+02	40179.0	41415.0	41961.0	42819.0	4.371100e+04
ID_Orden_Entidad	217589.0	8.923342e+00	3.752280e+00	1.0	6.0	11.0	12.0	1.200000e+01
ID_Depto_Ejecucion	217589.0	9.970706e+00	8.267028e+00	0.0	2.0	9.0	15.0	3.200000e+01
ID_Depto_Entidad	217589.0	9.208784e+00	8.419223e+00	0.0	1.0	9.0	14.0	3.200000e+01
ID_Grupo_Numerico	217589.0	1.213692e+00	4.099127e-01	1.0	1.0	1.0	1.0	2.000000e+00
Tiempo_Planeado	217589.0	7.156800e+01	1.281616e+02	0.0	15.0	31.0	90.0	1.179000e+04
SALIDA	217589.0	8.440856e-01	2.987332e-01	0.0	1.0	1.0	1.0	1.000000e+00

Nota. Tomada de MEP_20210218.

Con este resultado, se precisan las diferencias entre los rangos de cada variable, por lo cual el ideal es normalizar las variables que utilizan diferentes escalas y rangos, puede que el modelo llegue a converger sin normalización de estas variables; sin embargo, se ha demostrado que, de no hacerlo, el entrenamiento presenta dificultades, con lo que se determina que el resultado del modelo depende de la elección de las variables de entrada utilizadas.

Es importante resaltar que se debe normalizar de la misma forma el conjunto de datos de prueba, así como el conjunto de datos que se utilizarán para incluir al modelo para predecir el resultado. Existen varias maneras que permiten escalar los datos del modelo previo al momento de presentarlos, para que todos los datos tengan una escala lo más equivalente posible (normalización), para facilitar el proceso de aprendizaje de la red neuronal.

- En Python se puede crear una función norm, que reescale los datos de cada una de las variables de entrada en un rango [0,1] y las centre con respecto a una media 0 con desviación estándar 1 (estandarización), de forma que las columnas de todas las variables que se utilicen tengan los mismos parámetros que una distribución normal estándar (media cero y varianza unidad):

  def norm(x): return (x – train_stats[mean’]) / train_stats[‘std’]

  Luego ya se puede utilizar esta función aplicándola sobre los datos de entrenamiento y los datos de prueba, y al final, sobre los datos que no se le han presentado al modelo, y que se utilizarán para la etapa de predicción o inferencia.

  x_train = norm(train_data)

  x_test = norm(test_data)

- Luego de separar los datos de entrenamiento y de pruebas, se puede utilizar el método **to_numpy()** de la librería numpy, la cual permite extraer los datos en arreglos para cada variable, y de esta manera, invocar la función max , que obtiene el valor máximo de todas las variables, para luego dividir cada uno de los arreglos por este valor, de manera muy similar a como se realizó en el punto anterior:

Figura 19

Normalización del conjunto de datos.

```
In [16]: # 3. Separar el conjunto de Datos en datos de Entrenamiento y Pruebas, Extracción y Escalamiento de los datos


         # Separar el conjunto de Datos en datos de Entrenamiento y Pruebas
         train_df = df_full.sample(frac=0.8,random_state=0)
         test_df = df_full.drop(train_df.index)

         # Extracción en arreglos numpy
         train_data=train_df.to_numpy()
         test_data=test_df.to_numpy()
         train_data_shape= train_data.shape
         test_data_shape= test_data.shape
```

Nota. Tomada de MEP_20210218.

- Una opción adicional es utilizar el método **MinMaxScaler()** de la librería **sklearn.preprocessing**, la cual permite para obtener las escalas máximas y mínimas de un rango de datos, para luego invocar el método **fit(dataset**)que obtiene el valor máximo de un conjunto de datos entregado, y luego aplicar el método **transform(dataset),** donde finalmente quedaría normalizado el conjunto de datos de entrenamiento y de pruebas.

Figura 20

Normalización del conjunto de datos (Escalar).

```
# Selección de los datos de la datos de salida
y_train = train_data [:,15]
y_test = test_data[:,15]

# Determinación del valor máximo para poder escalar (normalizar) la data de entrenamiento y de pruebas
X_max=train_data[:,0:15].max(axis=0)
x_train= train_data[:,0:15]/X_max
x_test =test_data[:,0:15]/X_max
```

Nota. Tomada de MEP_20210218.

Para visualizar finalmente los datos como quedan normalizados se puede volver a utilizar los métodos **describe()** y **transpose()** para visualizar los datos y obtener las métricas estadísticas que confirmarían la normalización de todos los datos que serán utilizados para el entrenamiento y las pruebas.

5.2 DESARROLLO DEL MODELO

Esta etapa se constituye en la parte más interesante: Desarrollar el modelo que cumpla con las expectativas iniciales, que, en últimas, construir el modelo que permita dar respuesta a lo que queremos obtener del modelo. También es la parte menos mecánica, y más compleja, pues a parte de decidir la arquitectura del modelo, se debe determinar los hiperparámetros que constituyen su configuración

5.2.1 Definición del Modelo

Para establecer las restricciones que se requieren en la salida de la red neuronal, se debe decidir inicialmente la configuración de la última capa de la red, el tipo de activación a utilizar y la relación con la función de pérdida que se deben determinar para el modelo.

Para este caso, hemos optado por 2 capas densamente conectadas: 1 capa de entrada de los datos y 1 capa intermedia u oculta de 64 neuronas cada una; y una capa de salida, con 1 neurona, cuya configuración se declara utilizando el método sequential de la librería Keras , donde se le configura el número de neuronas de cada capa y el tipo de activación a utilizar

Figura 21

Creación y compilación de la Red Neuronal.

In [19]:
```
# 4. Creación y Compilación de la Red Neuronal usando el método Sequiential de la libreria Keras
# Este modelo tiene 3 capas: 1 de Entrada , 1 de Salida  y 1 Intermedia u oculta
# la arquitectura del mdelo (Numero de capas, neuronas, funciones de activación, etc) se establece despues de varias pruebas e intentos


model = keras.Sequential([

    # Capa de Entrada:
    keras.layers.Dense(64, activation='relu', input_shape=(x_train.shape[1],)),

    # Capa intermedia u oculta
    keras.layers.Dense(64, activation='relu'),

    # Capa de Salida:
    keras.layers.Dense(1, activation='sigmoid'),
])
```

Nota. Tomada de MEP_20210218.

Para generar esta configuración se debe utilizar el método compile de la librería Keras, determinando el optimizador, la relación con la función de pérdida y las métricas para determinar la precisión de la red neuronal:

Figura 22

Compilación del modelo.

```
# Compilar el Modelo
model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['accuracy'])
```

Nota. Tomada de MEP_20210218.

Finalmente, para inspeccionar la arquitectura con la cual se construyó la red neuronal se puede invocar el método **summary()**

Figura 23

Configuración del Modelo.

```
# Imprimir la configuración del Modelo
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 64)                1024
_________________________________________________________________
dense_2 (Dense)              (None, 64)                4160
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 65
=================================================================
Total params: 5,249
Trainable params: 5,249
Non-trainable params: 0
```

Nota. Tomada de MEP_20210218.

5.2.2 Configuración del Modelo

Otra de las decisiones clave en la construcción del modelo es la definición de la función de perdida y el optimizados a utilizar.

Función de Pérdida

La función de perdida a utilizar debe coincidir con el tipo de problema que trata de resolver el modelo, basado en una métrica de rendimiento. Para un problema de regresión La raíz del error cuadrático medio (RECM) o raíz de la desviación cuadrática media (RDCM) (en inglés: *root-mean-square deviation*, MSD, o *root-mean-square error*, MSE) es una medida de uso frecuente de las diferencias entre los valores (valores de muestra o de población) predichos por un modelo o un estimador y los valores observados. El MSE representa la raíz cuadrada del segundo momento de la muestra de las diferencias entre los valores previstos y los valores observados o la media cuadrática de estas diferencias. Estas desviaciones se denominan residuos cuando los cálculos se realizan sobre la muestra de datos que se utilizó para la estimación y se denominan errores (o errores de predicción) cuando se calculan fuera de la muestra. El MSE sirve para agregar las magnitudes de los errores en las predicciones para varias veces en una sola medida de poder predictivo. El MSE es una medida de precisión, para comparar errores de predicción de diferentes modelos para un conjunto de datos en particular y no entre conjuntos de datos, ya que depende de la escala.

$$\text{MSE} = \sqrt{1/m \sum \quad (Y\ estimada\ -\ Y\ real)^2}$$

Aunque la RCME es generalmente la métrica de rendimiento preferida para tareas de regresión en algunos casos puede utilizarse la función del error absoluto medio MAE (*Mean Absolute Error*): En Estadística, el error absoluto medio es una medida de la diferencia entre dos variables continuas. Considerando dos series de datos (unos calculados y otros observados) relativos a un mismo fenómeno, el error absoluto medio sirve para cuantificar la precisión de una técnica de predicción comparando por ejemplo los valores predichos frente a los observados, el tiempo real frente al tiempo previsto, o una técnica de medición frente a otra técnica alternativa de medición.

$$\text{MAE} = 1/m\sqrt{\sum \quad |Y\ estimada\ -\ Y\ real|}$$

Ambas métricas son usadas en modelos de regresión. MAE es más robusto para valores atípicos, ya que no al no utilizar el cuadrado, y debido a que el MSE si lo utiliza, los errores grandes tienen una influencia relativamente mayor que los errores más pequeños. MSE es más utilizada cuando los errores grandes son muchos más grandes que los errores equivalente pequeños.

Optimizador

Es uno de los argumentos del método compile(), que en TensorFLow, con la API de Keras se disponen del RMSProp, AdaGrad, Adadelta, Adma, Adamax, cuyo detalle de puede encontrar en la documentación de Keras, y se constituyen en variantes u optimizaciones del algoritmo de descenso del gradiente presentado; sin embargo, se debe considerarse que las particularidades de cada uno de estos optimizadores se adaptan mejor o peor para cada uno de los problemas que se pretendan abordar. Por ejemplo, el AdaGrad mejora el algoritmo de descenso del gradiente cuando se tienen varias dimensiones. Pero a su vez, aumenta el riesgo de no llegar al óptimo global.

Una mejora se presenta con el algoritmo RMSProp, que funciona mejor que el AdaGrad, y puede ser el más utilizado en *Deep Learning* hasta la llegada del optimizador Adam y unas variantes como el Nadam

5.3 ENTRENAMIENTO DEL MODELO

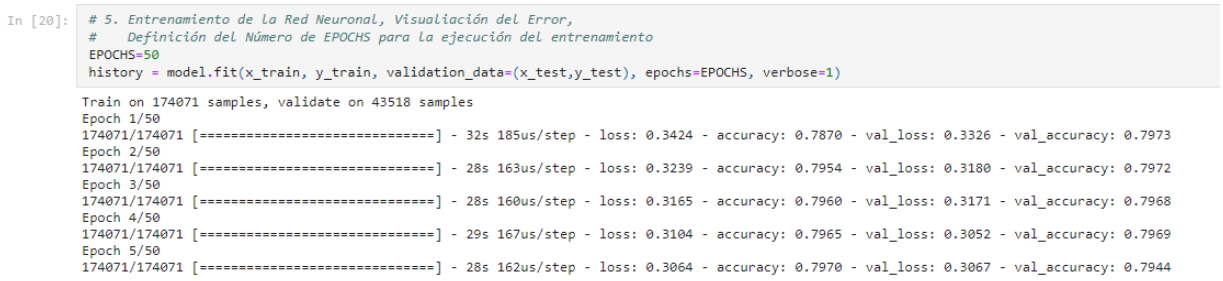
El hiperparámetro EPOCHS indica el número de veces que los datos de entrenamiento han de pasar por la red neuronal en el proceso de entrenamiento. El valor del hiperparámetro EPOCH es supremamente importante para la configuración de la red, pues entre más **EPOCHS**

se configuren, puede mostrar una mayor precisión la red, para los datos de entrenamiento. Pero a su vez, debe entenderse que, si este número es demasiado alto la red podría tener problemas de sobre ajuste (overfitting), por lo tanto, encontrar el valor óptimo para este hiperparámetro es muy relevante, y está directamente relacionado con los datos de validación que se hayan definido. Además, es directamente proporcional con el tiempo de entrenamiento del modelo, entre más EPOCHs mayor será el tiempo de entrenamiento de la red.

Para el proceso de entrenamiento se invoca el método **fit()** de la librería Keras, que recibe como argumentos, los conjuntos de datos de entrenamiento (entradas y salida), separados previamente en los **datasets x_train** y **y_train**, y los datos de prueba o validación en los **datasets x_test** y **y_test**, el número de EPOCHS a entrenar el modelo, y un parámetro que permite visualizar el proceso de entrenamiento (verbose) de la siguiente manera:

Figura 24

Entrenamiento de la Red Neuronal.



```
In [20]: # 5. Entrenamiento de la Red Neuronal, Visualiación del Error,
         #    Definición del Número de EPOCHS para la ejecución del entrenamiento
         EPOCHS=50
         history = model.fit(x_train, y_train, validation_data=(x_test,y_test), epochs=EPOCHS, verbose=1)

Train on 174071 samples, validate on 43518 samples
Epoch 1/50
174071/174071 [==============================] - 32s 185us/step - loss: 0.3424 - accuracy: 0.7870 - val_loss: 0.3326 - val_accuracy: 0.7973
Epoch 2/50
174071/174071 [==============================] - 28s 163us/step - loss: 0.3239 - accuracy: 0.7954 - val_loss: 0.3180 - val_accuracy: 0.7972
Epoch 3/50
174071/174071 [==============================] - 28s 160us/step - loss: 0.3165 - accuracy: 0.7960 - val_loss: 0.3171 - val_accuracy: 0.7968
Epoch 4/50
174071/174071 [==============================] - 29s 167us/step - loss: 0.3104 - accuracy: 0.7965 - val_loss: 0.3052 - val_accuracy: 0.7969
Epoch 5/50
174071/174071 [==============================] - 28s 162us/step - loss: 0.3064 - accuracy: 0.7970 - val_loss: 0.3067 - val_accuracy: 0.7944
```

Nota. Tomada de MEP_20210218.

Si se observa con detenimiento el progreso del entrenamiento, que se refleja en la consola de salida del modelo, se puede apreciar, se va informando para cada EPOCH, tanto las métricas obtenidas con los datos de entrenamiento (pérdida (loss), el error absoluto medio (mae) y el error cuadrático medio (mse)), como las métricas obtenidas con los datos de validación (pérdida (val_loss), el error absoluto medio (val_mae) y el error cuadrático medio (val_mse)), que se estudiarán con más detalle en la siguiente sección.

5.3.1 Visualización del proceso de Entrenamiento

Luego de que el proceso de entrenamiento ha finalizado, se dispone de una alternativa muy útil, que permite visualizar el resumen de la ejecución del proceso de entrenamiento por medio del objeto historial, que es retornado por el método **fit()** y especificando el número de EPOCHS finales a desplegar con el método **tail.**

Figura 25

Creación de Dataframe para desplegar el resultado de los EPOCHS.

```
In [21]: # Creación de un DataFrame de la librería Pandas para desplegar el resultado de las últimas 10 EPOCHS
         # con su respectiva pérdida y precisión por cada EPOCH

         hist = pd.DataFrame(history.history)
         hist['epoch'] = history.epoch
         hist.tail(10)
```

Out[21]:

	val_loss	val_accuracy	loss	accuracy	epoch
40	0.296747	0.798107	0.307498	0.797669	40
41	0.296123	0.796636	0.308512	0.797652	41
42	0.296134	0.798612	0.308318	0.797514	42
43	0.296985	0.798084	0.309065	0.797623	43

Nota. Tomada de MEP_20210218.

La diferencia del comportamiento de la métrica mse de los datos de entrenamiento contra los de validación val_mse, indica el “sobreajuste” (*ovefitting*) que está sufriendo el modelo. En otras palabras, cuando se presenta este “sobreentrenamiento” del modelo, redunda en un incremento del ajuste a los datos de entrenamiento. Cuando el modelo detecta datos que no han sido usados para entrenar, el modelo se empezará a comportar de una manera ineficiente.

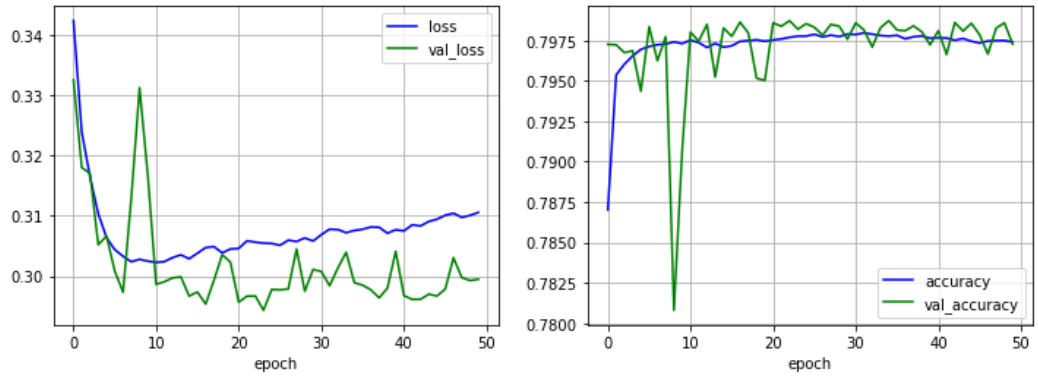
Para ver cómo van evolucionando la pérdida del error y el nivel de precisión del modelo, entre la data de entrenamiento y la data de validación o pruebas, se puede utilizar el siguiente código para visualizar gráficamente:

Figura 26

Visualización de la perdida y precisión del proceso de entrenamiento.

```
In [22]: # Visualización Gráfica de la pérdida y precisión del proceso de Entrenamiento

fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2,figsize=(12,4))
hist.iloc[0:200].plot(x='epoch', y=['loss','val_loss'],ax=ax1,marker="",color=['blue','green'])
plt.grid()
hist.iloc[0:200].plot(x='epoch', y=['accuracy','val_accuracy'],ax=ax2,marker="",color=['blue','green'])
ax1.grid()
ax2.grid()
plt.show()
```



Nota. Tomada de MEP_20210218.

En la figura anterior se puede apreciar que más o menos a partir de la EPOCH número 10, la pérdida del error de la data de entrenamiento empieza a ser mayor, que la pérdida del error de la data de validación, y cada vez empieza crecer más, mientras que la pérdida del error para los datos de validación empieza a ser menor. Es decir, el modelo se va ajustando a los datos que se usan para entrenar, y, por lo tanto, generaliza en menor proporción para los datos de validación.

5.3.2 Overfitting

El concepto de sobreajuste de un modelo se presenta cuando el modelo obtenido se ajusta tanto a los datos de entrenamiento que no puede realizar las predicciones correctas en datos nuevos que nunca se le hayan presentado al modelo. En resumen, es el fenómeno que se presenta en un modelo que modela los datos de entrenamiento demasiado bien, aprendiendo detalles de estos que no son generales, debido a que se sobreentrenan al modelo

y empieza a considerar como válidos solo los datos idénticos de los datos de entrenamiento, incluyendo sus defectos, que se denomina también como “ruido” en este contexto. Es decir, es una situación en la que el modelo puede tener una baja tasa de error de clasificación para los datos de entrenamiento, pero no generaliza bien a la población de datos nuevos en los que se está interesados en predecir con el modelo.

En general, es evidente que esta situación presenta un impacto negativo en la eficiencia del modelo, cuando se usa para inferencia de datos nuevos, por lo que es muy importante evitar caer en este sobre ajuste. De aquí, la importancia de reservar una parte de los datos de entrenamiento, como datos de validación, debido a que el modelo los utiliza para probar y evaluar diferentes opciones de hiperparámetros para minimizar este sobre ajuste, haciendo una puesta a punto o *tunning* de estos hiperparámetros, como lo son los *epochs*, el *ratio* de aprendizaje o los datos de la arquitectura de la red neuronal, para disminuir este sobreajuste.

5.3.3 Predicción Data de Entrenamiento

Para predecir el conjunto de datos de entrenamiento, almacenado en la variable x_train, se utiliza el siguiente código:

```
y_predict_train=model.predict(x_train)
```

Figura 27

Predicción del modelo vs. Data de entrenamiento.

```
In [23]: # Predicción del modelo vs Data de Entrenamiento y desescalamiento
         # Visualización de la Salida de la data de Entrenamiento VS la variable de Entrada Cuantía del Proceso

         y_predict_train=model.predict(x_train)
```

Nota. Tomada de MEP_20210218.

Para visualizar gráficamente el resultado de la respuesta que entrega el modelo sobre la variable **y_predict_train** contra la variable de salida ingresada para este conjunto de datos,

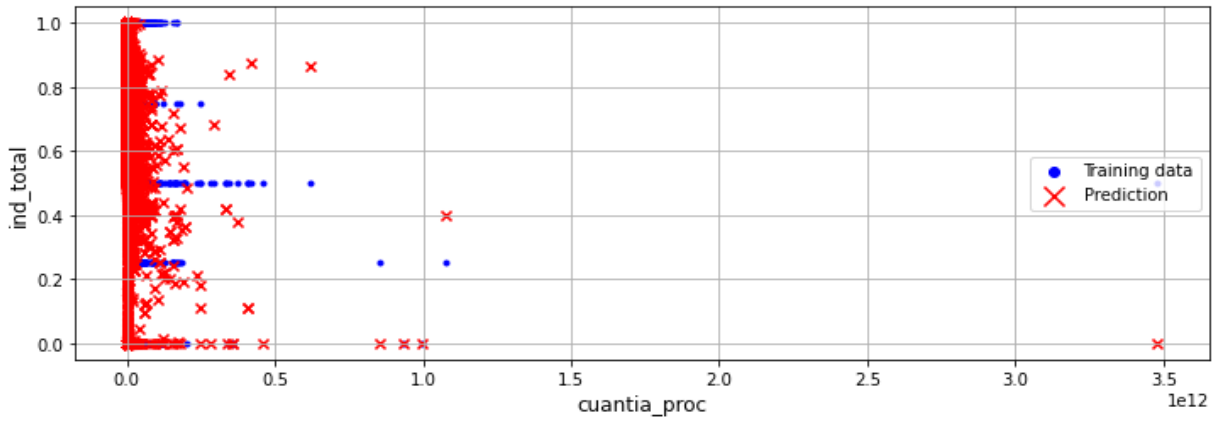
puede resultar un poco difícil de entender, sobre todo, cuando existen varias variables de entrada al modelo. Por ello, y para facilidad del proceso de inferencia, se opta por mostrar el proceso de predicción contra una de las variables de entrada utilizadas. Es claro que mezclarlas todas en un gráfico sería bastante caótico para entender. Para visualizar gráficamente la respuesta del modelo contra una variable de salida se usa el código como se muestra en la **Figura 28.**

Figura 28

Visualización de la Salida de la data de Entrenamiento vs. la variable de Entrada Cuantía del Proceso

```
# Plot ind_total
fig, ax1 = plt.subplots(nrows=1, ncols=1,figsize=(12,4))
ax1.scatter(train_cuantia_proc,y_train, marker=".",color='blue')
ax1.scatter(train_cuantia_proc,y_predict_train,marker="x",color='red')
plt.sca(ax1)
plt.legend(['Training data','Prediction'],loc="center right",markerscale=2)
plt.ylabel('ind_total', fontsize=12)
plt.xlabel('cuantia_proc', fontsize=12)
ax1.grid()

plt.show()
```



Nota. Tomada de MEP_20210218.

5.3.4 Evaluación del modelo con Data de Entrenamiento

Para evaluar lo bien que generaliza el modelo o lo bien que evalúa un modelo sobre unos datos de prueba que no se le han presentado al *modelo* durante el entrenamiento, para

tener una clara suposición de que el modelo puede predecir una respuesta lo más cercana posible, con respecto a estos datos de verificación no mostrados el modelo.

Para ello se puede utilizar el método evaluate(),el cual recibe como argumentos el conjunto de datos que se quiere evaluar. Por lo tanto, se puede evaluar la pérdida de error y la precisión del modelo con un conjunto de datos entregado.

Dado que se separaron el conjunto de datos para entrenar el modelo en un conjunto de datos de entrenamiento y un conjunto de datos de pruebas, se puede evaluar el resultado de la eficiencia del modelo con cada uno de estos conjuntos de datos; como se muestra en la **Figura 29**, se hace uso del código **tVal_accuracy=model.evaluate**, para evaluar la precisión del modelo con los datos de entrenamiento almacenados en las variables x_train y y_train.

Figura 29

Evaluación del modelo a partir de la data de entrenamiento.

```
In [24]: # Evaluación del modelo a aprtir de la data de Entrenamiento
         # Impresión de la estimación del Error, Pérdida y Precisión del Modelo

         tLoss, tVal_accuracy  = model.evaluate(x_train,y_train)

         print('Train Data Errors in:')
         print('Loss:', tLoss)
         print('Accuracy:', tVal_accuracy )

         174071/174071 [==============================] - 12s 67us/step
         Train Data Errors in:
         Loss: 0.30851112641865175
         Accuracy: 0.7971574664115906
```

Nota. Tomada de MEP_20210218.

Como conclusión del proceso de evaluación del modelo planteado, si los resultados de pérdida y precisión tanto para el conjunto de datos de Entrenamiento como para el conjunto de datos de validación convergen en los mismos valores respectivamente, se infiere que el modelo está preparado para la fase final de inferencia o predicción, de un conjunto de datos de prueba los cuales no se le han presentado al modelo.

6 EVALUACIÓN DEL MODELO

En concordancia con lo referenciado en el capítulo 2.4 del Diseño Metodológico, la evaluación del modelo pretende realizar la carga de la data de prueba reservada, de manera que se determine la precisión de la predicción del modelo, para lo cual es necesario llevar a cabo los siguientes subfases:

1. **Inferencia o predicción data de prueba.** Este contempla la predicción de la salida para el conjunto de datos de prueba.
2. **Evaluación del modelo con la data de prueba.** A modo de cierre, esta subfase indica el porcentaje de precisión del modelo, como resultado de la predicción de la data de prueba.

6.1 Inferencia o Predicción Data de Prueba

La fase de inferencia o predicción es aquella, en donde se recogen los frutos del trabajo realizado sobre la puesta a punto del modelo, y se comprueba que está preparado para predecir una respuesta a la pregunta planteada inicialmente, con un nivel de precisión adecuado. En efecto, es poner a trabajar el modelo, recordando que lo que motiva el diseño del mismo, es dar una respuesta al problema planteado con la mayor confiabilidad posible.

Al igual que, con la data de entrenamiento como se observa en la **Figura 30**, se utiliza el método de la librería Keras: **predict()** V cabe resaltar que a este método se le debe presentar el conjunto de datos que nunca se le han presentado al modelo para lo cual la data de prueba (test) fue reservada desde el inicio, conjunto de datos que solo contiene las variables de entrada, de tal manera que el modelo infiera o prediga la respuesta; para ello, se debe leer o

cargar el conjunto de datos con las variables de entrada, usando la clase **pandas.read_csv (Pandas)**, la cual recibe como parámetro el nombre del archivo donde está la data reservada .

Figura 30

Predicción del modelo vs. Data de prueba.

```
In [25]: # Predicción del modelo vs Data de prueba
         # Visualización de la Salida de la data de prueba VS la variable de Entrada Cuantía del Proceso
         y_predict_test=model.predict(x_test)
```

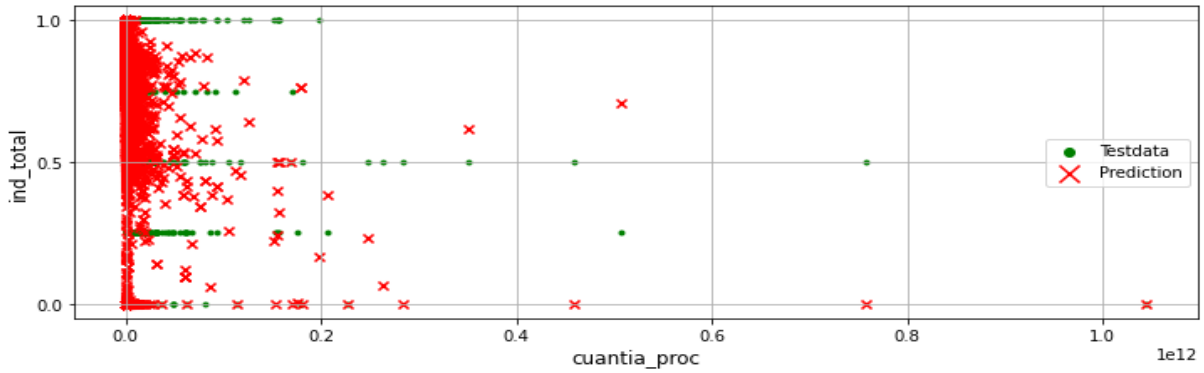
Nota. Tomada de MEP_20210218.

Es preciso señalar que, en aras del proceso de investigación se opta por invocar este método con los datos de entrenamiento y de prueba, para luego confirmar con el conjunto de datos reservado la confiabilidad de la predicción.

Figura 31

Visualización de la Salida de la data de prueba vs la variable de Entrada Cuantía del Proceso.

```
# Plot ind_total vs test_cuantia_proc
fig, ax1,  = plt.subplots(nrows=1, ncols=1,figsize=(12,4))
ax1.scatter(test_cuantia_proc,y_test,marker=".",color='green')
ax1.scatter(test_cuantia_proc,y_predict_test,marker="x",color='red')
plt.sca(ax1)
plt.legend(['Testdata','Prediction'],loc="center right",markerscale=2)
plt.yticks([0, 0.5, 1])
plt.ylabel('ind_total', fontsize=12)
plt.xlabel('cuantia_proc', fontsize=12)
ax1.grid()
plt.show()
```



Nota. Tomada de MEP_20210218.

Finalmente, para visualizar gráficamente la respuesta del modelo se puede hacer uso de opciones gráficas como las anteriormente presentadas o con un simple print, que permite

imprimir la interacción entre la respuesta y una variable de entrada, de ser tabulados estos datos contra las salidas reales se puede medir el porcentaje de acierto de la respuesta del modelo vs. la realidad.

6.2 Evaluación del Modelo con la Data de Prueba

Finalmente, se puede evaluar el resultado de la eficiencia del modelo con el conjunto de datos de prueba o validación almacenados en las variables x_test y y_test, el cual se invoca con el código **loss, val_accuracy=model.evaluate(x_test,y_test),** como se observa en la **Figura 32;** además, se encuentra el código usado para retornar el resultado de perdida y precisión del modelo con este conjunto de datos.

Figura 32

Evaluación del modelo a partir de la data de prueba.

In [26]:
```
# Evaluación del modelo a aprtir de la data de Pruebas
# Impresión de la estimación del Error, Pérdida y Precisión del Modelo
loss, val_accuracy  = model.evaluate(x_test,y_test)

print('\nTest Data Errors in:')
print('Loss:', loss)
print('Accuracy:', val_accuracy )

print('\nThe confusion matrix is:')
y_predict=(y_predict_test>0.5)
y_predict=y_predict.astype(int)
y_predict=y_predict.reshape(1,test_data_shape[0])
y_predict=y_predict[0,:]
y_test=y_test.astype(int)
print(y_predict.shape)
print(y_test.shape)

cm= confusion_matrix(y_test,y_predict)
print(cm)
```

```
43518/43518 [==============================] - 3s 69us/step

Test Data Errors in:
Loss: 0.2994621049715474
Accuracy: 0.7972792983055115

The confusion matrix is:
(43518,)
(43518,)
[[ 2271  8449]
 [   89 32709]]
```

Nota. Tomada de MEP_20210218.

7 HALLAZGOS

7.1 Hallazgos de la Data Usada en el Modelo

La calidad de la información de las variables de entrada seleccionadas en la data del disponible del SECOP, para evaluar el cumplimiento del tiempo y costo, puede ser considerada como confiable, puesto que dispone de plazos de ejecución, cuantías de procesos previstas, y a su vez, fechas de inicio y fin de ejecución y cuantías reales de los proyectos; lo cual, asegura la confiabilidad de la salida construida.

A pesar de la cantidad de la información disponible en el SECOP para cada uno de los proyectos, se evidencia que no cuenta con un dato que permita conocer la desviación en el cumplimiento del alcance. En concordancia, para este modelo se infiere que el estado del proyecto es el único campo de la data, que está directamente relacionado con el cumplimiento del alcance.

Luego del análisis realizado sobre los 72 campos de información que existen para cada proyecto en la base de datos obtenida de la plataforma del SECOP, se concluye que únicamente 15 de ellos aportan a la selección del conjunto de datos de entrenamiento, pruebas y verificación. Los datos usados están directamente relacionados con el costo, tiempo y la entidad, tanto en la fase de planeación como en la ejecución real. Entre los datos están código de la entidad, ID Tipo de Proceso, ID Régimen de Contratación, ID Objeto a Contratar, Cuantía Proceso, ID Familia, ID Clase, ID Origen de los Recursos, Año Cargue SECOP, Fecha Cargue SECOP, ID Orden Entidad, ID Departamentos Ejecución, ID Departamento Entidad, ID Grupo y Tiempo planeado.

7.2 Hallazgos de la Configuración del Modelo

La precisión de la predicción del modelo con la data de entrenamiento que corresponde al 80% del total de la data, fue del 79.31% y para el 20% correspondiente a la data de prueba fue del 79.72%, como resultado general de esta investigación se obtiene que el modelo de evaluación de proyectos presenta un nivel de precisión o accuracy del 79.51%, que para los modelos predictivos presenta un nivel aceptable de confianza.

El factor relevante en la configuración de la Red Neuronal, por encima del número de neuronas, o el número de capas intermedias u ocultas, o los hiperparámetros de la función de pérdida y el optimizador, expuestos en el capítulo 5.2.2; es el número de datos que se le presentan al modelo para su proceso de entrenamiento y aprendizaje.

En el proceso de calibración del hiperparámetro epochs, el cual indica el número de veces que los datos de entrenamiento han de pasar por la red neuronal en el proceso de entrenamiento, se puede determinar que a un mayor número, puede presentar una mejor precisión del modelo; sin embargo, al utilizar un valor cada vez más alto, la red puede presentar problemas de sobre ajuste (overfitting) y mayores tiempos de entrenamiento, por lo tanto encontrar el valor óptimo para este hiperparámetro es relevante, al estar directamente relacionado con los datos de validación que se han definido y recopilado.

La configuración del modelo de 2 capas densamente conectadas (1 capa de entrada de los datos y 1 una capa intermedia de 64 neuronas cada una) se selecciona luego de los resultados obtenidos en las pruebas realizadas, con diferentes configuraciones en cuanto al número de capas y al número de neuronas de cada una, usando el mismo conjunto de datos de entrenamiento y pruebas; al ser la configuración con el tiempo de ejecución y nivel de precisión de la predicción óptimo.

La selección de 1 capa de salida con 1 una sola neurona junto con la utilización de la función de activación no lineal sigmoid, permite reducir los valores extremos o atípicos en datos válidos sin ser excluidos, convirtiendo variables independientes de rango casi infinito en probabilidades simples entre 0 y 1. Esto fue corroborado con una prueba en la cual se configuró la capa de salida con dos neuronas, dando como resultado un nivel de precisión de la predicción del 30%.

Al calcular la salida del conjunto de datos mediante el promedio de las micro fórmulas de evaluación del Costo, Tiempo y Alcance, definidas en el numeral 4.3 de este documento, se obtiene un accuracy del 35%; sin embargo, al configurar el cálculo de la salida con un factor del 40% para las micro fórmulas de Costo y Tiempo, y un 20% para el Alcance, manteniendo la misma configuración de la Red Neuronal y conjunto de datos, se incrementa su precisión a un 79%. Es decir, la información fuente utilizada para la construcción de la salida, infiere en el grado de precisión del modelo.

Al ejecutar el modelo con diferentes configuraciones, se detectó una tendencia marcada cuando el resultado de la red neuronal es una salida binaria: 0 o 1; determinando que para optimizar el acurracy del modelo, se debe aumentar la cardinalidad de la salida en 5 rangos (0.00, 0.25, 0.50, 0.75 y 1.00), lo cual minimiza que se presenten tendencias y aumenta la precisión del modelo.

8 CONCLUSIONES

El modelo producto de esta investigación hace posible predecir el éxito de los proyectos haciendo uso de herramientas y algoritmos del Machine Learning y data histórica de proyectos ejecutados.

El modelo desarrollado es una herramienta que proporciona a la evaluación de proyectos, un indicador diferente a las métricas financieras tradicionales (TIR, VPN, Rentabilidad del Negocio, ROI y B/C (Costo - beneficio)) para la toma de decisiones, con el fin de minimizar la tasa de proyectos fracasados.

En la última década, más de 1000 contratos de obra por año, para la construcción de infraestructura en Colombia con una ejecución de más de 5000 millones de pesos fueron fallidos. De haber utilizado la predicción del modelo, el Estado podría haber invertido estos recursos en otros proyectos.

De haber sido utilizado este modelo por parte del Estado, más de 45,000 proyectos no exitosos, con costos superiores a 60 mil millones de pesos, pudieron haber sido reestructurados o rechazados previo a su inicio.

9 RECOMENDACIONES Y TRABAJOS FUTUROS

De acuerdo con el Trabajo de Grado desarrollado, se presentan las recomendaciones para tener en cuenta en trabajos futuros de evaluación de proyectos a partir de herramientas de Machine Learning.

Con base al valor del modelo y al resultado de esta investigación se puede incursionar en la creación de negocio enfocado en la implementación de herramientas de Machine Learning en la gestión de proyectos, mediante la creación de modelos de diagnóstico y predicción, de acuerdo con las necesidades de las organizaciones que estén alineadas con las Innovación.

Se puede indagar con las entidades ejecutorias sobre información adicional a la almacenada en la plataforma, que pueda ser usada como fuente confiable para plantear una nueva micro fórmula que mida el cumplimiento del alcance de un proyecto; o por qué no, incluir en el conjunto de datos de la plataforma para cada proceso, un campo en el cual la entidad pueda consignar un índice de cumplimiento del alcance, lo cual no solo mejoraría la precisión del índice de éxito sino la transparencia de los procesos de contratación.

Con este proyecto de grado se ha aprendido que se puede utilizar las diferentes aplicaciones del Machine Learning para poder brindar un servicio para las empresas donde se detecten pérdidas o deficiencias en su cadena de producción, y una herramienta muy poderosa y atractiva que con base a un levantamiento de datos que permitan modelar una RN que permita predecir resultados para poder adoptar mejoras en los flujos donde se detecten estas diferencias.

A medida que el número de datos resultado de la fase de recopilación y análisis aumenta, o que se mejoran las configuraciones de la RN, se requieren capacidades

computacionales mayores, mejorando la predicción de situaciones que repercutan en la cadena de cualquier modelo productivo.

Las variables de entrada del modelo se deben adaptar de acuerdo con la información que se obtenga del histórico de proyectos ejecutados y evaluados, al igual que la construcción del índice de salida.

Evaluar el modelo con otro sector económico, ratificará el nivel de precisión del modelo, para lo cual se hace necesario extender la recopilación o adquisición de bases de datos históricas o acervos de organizaciones del sector seleccionado para dicha evaluación.

Dentro de la base de datos obtenida de la plataforma de Datos Abiertos de Colombia alimentada por el SECOP I y SECOP II, es importante que se incluya el parámetro de calificación del proyecto respecto al Alcance, es decir, una métrica que establezca si el objeto/alcance del proyecto contratado cumple con las obligaciones contractuales definidas.

10 BIBLIOGRAFÍA

Anaconda, Inc. (2021). *Anaconda. Documentation*. Obtenido de https://docs.anaconda.com/anaconda/navigator/glossary/

Colombia Compra Eficiente. (2020). *Colombia Compra Eficiente*. Obtenido de https://www.colombiacompra.gov.co/secop/secop-i

Escuela Colombiana de Ingenieria Julio Garavito. (s.f.). *Escuela Colombiana de Ingenieria Julio Garavito*. Obtenido de Biblioteca/Base de datos: https://escuelaing.metaproxy.org/subjects/databases.php?letter=S

Goodfellow, I., Bengio, Y., & Corville, A. (2016). *MIT Press.* Obtenido de Deep Learning : http://www.deeplearningbook.org

Hetch-Nielsen, R. (1988). Neurocomputing: Picking the Human Brain. *IEEE Spectrum*, 13-18.

IBM. (11 de 05 de 1997). *IBM100.* Obtenido de Icons of Progress: https://www.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/

Khepri, W. (2 de Noviembre de 2018). *Redes Neuronales, ¿qué son?. Introducción a las redes neuronales*. Obtenido de Medium: https://medium.com/@williamkhepri/redes-neuronales-que-son-a64d022298e0

Laurence Goasduff. (20 de Septiembre de 2017). *Gartner.* Obtenido de https://www.gartner.com/en/newsroom/press-releases/2017-09-20-gartner-says-deep-learning-will-provide-best-in-class-performance-for-demand-fraud-and-failure-predictions-by-2019

Ministerio de Tecnologías de la Información y las Comunicaciones. (2021). *Datos Abiertos Colombia*. Obtenido de https://www.datos.gov.co/Gastos-Gubernamentales/SECOP-I/xvdy-vvsk

MIT. (07 de 06 de 2017). *MIT Technology Review.* Obtenido de https://www.technologyreview.es/s/7927/el-dia-que-la-humanidad-fue-derrotada-por-una-inteligencia-artificial

Open AI. (Febrero de 2019). *Better Language Modellis and their applications .* Obtenido de https://openai.com/blog/better-language-models/

OpenDoor Technology. (20 de Febrero de 2019). Obtenido de https://www.opendoorerp.com/the-standish-group-report-83-9-of-it-projects-partially-or-completely-fail/

Organisation for Economic Co-Operation and Development. (2019). Obtenido de https://stats.oecd.org/viewhtml.aspx?datasetcode=SNA_TABLE1&lang=en

Pacheco, G. G. (2019). *Gerencia Fundamental de Proyectos.*

Pacheco, G. G. (2020). *MS Project para la Gerencia Fundamental de Proyectos.*

Project Management Institute. (2017). *A Guide to the Project Management Body of Knowledge.* Pennsylvania: Project Management Institute.

Project Management Institute PMI®. (2013). *La guía de los fundamentos para la dirección de proyectos (Guía del PMBOK).* EE.UU: Project Management Institute.

Project Management Institute PMI®. (2017). *La guía de los fundamentos para la dirección de proyectos (Guía del PMBOK).* EE.UU: Project Management Institute.

Project Management Institute PMI®. (2018). *Pulse of the Profession. El éxito en tiempos de disrupción: Ampliación del panorama de entrega de valor para abordar el alto costo de un bajo desempeño.* EE.UU.

Project Management Institute PMI®. (2019). *Pulse of the Profession. El futuro del trabajo: Liderar con PMTQ.* EE.UU.

Roberts, A., & Wallace, W. (2014). *Gestión de Proyectos.* Gran Bretaña.

Scopus®. (2021). *Scopus® Base de datos de citas y resúmenes curada por expertos*. Obtenido de https://0210a188u-y-https-www-elsevier-com.escuelaing.metaproxy.org:9443/solutions/scopus

The Standish Group. (2015). *The Chaos Report.* The Standish Group International, Inc. Obtenido de https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

Waltman, N. J. (s.f.). *VOS Viewer*. Obtenido de Leiden University's Centre for Science and Technology Studies (CWTS): https://www.vosviewer.com/