

Diseño de un modelo de evaluación de proyectos a partir de herramientas de *machine learning* o aprendizaje automatizado

Laura Tatiana Nova Barreto

Arquitecta y magíster en Desarrollo y Gerencia Integral de Proyectos
laura.nova@mail.escuelaing.edu.co

Jorge Sebastián Caro Mesa

Ingeniero civil y magíster en Desarrollo y Gerencia Integral de Proyectos
jorge.caro@mail.escuelaing.edu.co

Camilo Andrés Cruz Rodríguez

Ingeniero de sistemas, magíster en Desarrollo y Gerencia Integral de Proyectos
camilo.cruz-r@mail.escuelaing.edu.co

Revista **IDGIP**

ISSN 2619-1830 (en línea)

Volumen 1, N.º 4

Enero-diciembre de 2021,

pp. 32-44

Recibido: 4/7/2021

Aceptado: 2/9/2021

Disponible en <http://revistas.escuelaing.edu.co/index.php/idgip>

Resumen: El Project Management Institute asegura en el informe del 2019 que el 18 % de los proyectos fracasaron y el 39 % no logró los objetivos; sin embargo, aquellas empresas que han implementado nuevas prácticas, conocidas como "PMTQ Innovators", disminuyeron en un 42 % los proyectos que han excedido su presupuesto inicial y un 46 % superaron el tiempo planeado. Por lo anterior, las estadísticas fortalecen la necesidad de nuevas herramientas que contribuyan a la disminución de las tasas de fracaso de los proyectos con el uso de tecnología en la evaluación. Es ahí donde la investigación demuestra que es posible predecir el éxito de los proyectos usando herramientas y algoritmos de 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 de infraestructura de la última década en Colombia, a los cuales se les calcula su éxito con la triple restricción (costo, tiempo y alcance). De esta forma se determinó que, si se hubiera usado este modelo, más de 45.000 proyectos no exitosos con costos superiores a 60.000 millones de pesos se habrían podido reestructurar o rechazar antes del inicio.

Palabras claves: gestión de proyectos, aprendizaje automático, inteligencia artificial, predicción, evaluación de proyectos. Design of a model project evaluation based on Machine Learning tools or automated learning

Inglés

Abstract: The Project Management Institute, in 2019, reported that 18% of projects failed and 39% did not achieve their objectives. However, companies that have implemented new practices, known as "PMTQ Innovators", decreased the number of projects that have exceeded their initial budget by 42%, and those exceeding their planned time by 46%. Therefore, the statistics strengthen the need for new tools that contribute to failure rate reduction of projects with the use of technology in project evaluation. This is where the research shows that it is possible to predict a project's success using Machine Learning tools and algorithms, through a supervised learning system, modelled by means of an artificial neural network trained from historical databases of infrastructure projects of the last decade in Colombia. Their success is calculated under the triple constraint (budget, time, and scope), where it was determined that if this model had been used, more than 45,000 unsuccessful projects, with costs exceeding 60,000 million pesos, could have been restructured or rejected prior to their start.

Keywords: project management, machine learning, artificial intelligence, prediction, project evaluation

1. INTRODUCCIÓN

Debido a la alta tasa de fracasos de los proyectos en general, lo cual genera incumplimientos o sobrecostos, se considera necesario contar con un modelo de evaluación que se adopte desde la fase de planeación de cada proyecto. Por ello, el presente documento propone la aplicación de nuevas tecnologías para la evaluación de proyectos.

Con el fin de responder al fracaso de los proyectos, se opta por diseñar un modelo de evaluación a partir de herramientas y algoritmos de Machine Learning o aprendizaje automatizado, que permita predecir el índice de éxito. Dicho modelo demanda una data histórica de proyectos y conocimientos previos en gestión de proyectos, Machine Learning y redes neuronales; de tal manera, se asegura que el modelo esté acorde con los lineamientos de la triple restricción y contribuya positivamente a la evaluación de proyectos.

2. PROBLEMA, JUSTIFICACIÓN Y OBJETIVOS

2.1 Justificación

2.1.1 Pregunta de investigación

Mediante el uso de las herramientas disponibles de Machine Learning o aprendizaje automatizado, y recopilando como insumo una data del mundo real, ¿es posible que se logre diseñar un modelo de diagnóstico para los gerentes de proyectos en el cual se prediga la tasa de éxito de un proyecto?

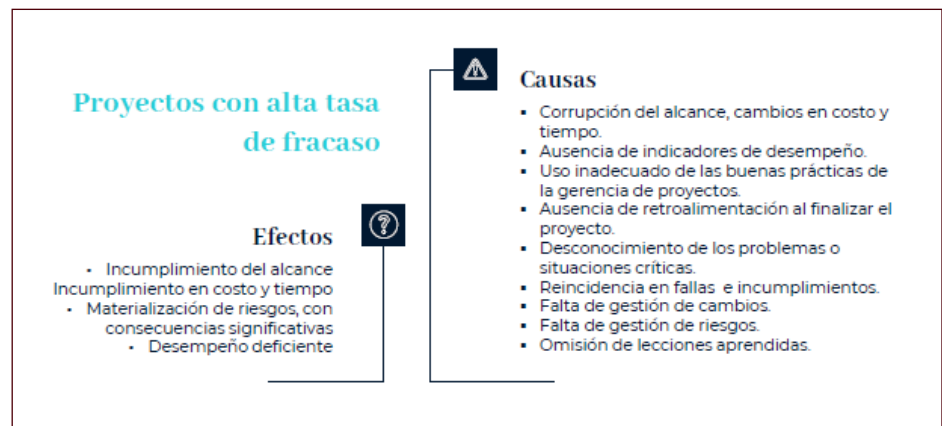


Figura 1. Problema, efectos y causas.

2.1.2 Oportunidad

“... el 85 % de los participantes de la Encuesta CEO 2019 de PwC afirman 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).

3. METODOLOGÍA

La metodología se basó en el marco conceptual, la recopilación y el estudio de datos, diseño del modelo y evaluación del modelo.



Figura 2. Diseño metodológico.

3.1 Marco conceptual

La primera fase 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 les permita a los computadores imitar la inteligencia humana.
- Machine Learning. Subconjunto de inteligencia artificial que incluye técnicas estadísticas profundas, que les permiten a las máquinas mejorar en tareas específicas con experiencia.
- Deep Learning. Subconjunto de Machine Learning 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.

3.1.1 Inteligencia artificial

Corriente académica para simular, por medio de redes neuronales, modelos de conocimiento que puedan dar una respuesta similar a como lo hace el cerebro humano. Existen dos tipos de inteligencia artificial: la débil y la fuerte.

- La IA débil es un asistente para el ser humano en las actividades que cotidianamente realiza. El uso excesivo de este apoyo puede llevar a que el humano desaprenda ciertas actividades y, con esto, se atrofién las partes o músculos que se dejan de usar.
- La IA fuerte 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 remplacen al ser humano.

3.1.2 Machine Learning

Subconjunto de la IA que les proporciona a los computadores la capacidad de aprender, sin que un programador establezca las reglas para producir un resultado. Hay tres tipos de aprendizaje: supervisado, no supervisado y por refuerzo.

- Aprendizaje supervisado: los datos de entrenamiento de los modelos incluyen la solución deseada. Esto implica que un modelo aprenda a partir de una función que mapea una entrada a una salida.
- Aprendizaje no supervisado. Los modelos por entrenar no incluyen etiquetas; por lo tanto, el algoritmo es el que clasifica la información.
- Aprendizaje por refuerzo. Su configuración tiene un agente que explora un universo desconocido y determina las acciones que debe predecir mediante prueba y error, con el apoyo de calificaciones del usuario.

3.1.3 Deep Learning

Basado fundamentalmente en el desarrollo de modelos que simulan el funcionamiento del cerebro humano y, específicamente, el de sus células: las neuronas. La agrupación de esta estructura en diferentes capas constituye los modelos denominados redes neuronales.

- Redes neuronales artificiales: se definen como la aproximación del Machine Learning para imitar la actividad en capas de las neuronas de la neocorteza cerebral.
- Redes profundas básicas: son redes neuronales que comparten algunas propiedades básicas y consisten en un número de neuronas interconectadas en varias capas. Su diferencia radica en la arquitectura de la red, cómo se encuentran interconectadas y su forma de entrenamiento.

3.2 Recopilación y estudio de datos

La segunda fase, como su nombre lo indica, consistió en la recopilación y estudio de los datos históricos de proyectos ejecutados en Colombia por entidades públicas, con el fin de obtener información verídica y eficaz para evaluar el desempeño de los proyectos. Dicho proceso se llevó a cabo con los siguientes pasos:



Figura 3. Recopilación y estudio de datos.

3.2.1 Recolección de datos

Se basó 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 por emplear.

3.2.2 Depuración de datos

Una vez establecido el sector de estudio, se dispusieron los criterios de verificación y depuración de los datos disponibles entre los años 2010 y 2020, que en total fueron 9.106.590. La depuración se realizó con base en el grupo y el objeto por contratar de servicios de edificaciones, construcción de instalaciones y mantenimiento y terrenos, edificios, estructuras y vías.



Figura 4. Depuración de datos.

3.2.3 Análisis de datos

Se analizó cada uno de los campos que incluye los datos de cada proyecto, y se hizo la selección previa de los campos que se infiere que pueden llegar a contribuir al diseño del modelo.

- Selección de variables de entrada: en primer lugar, se analizó y se determinó cada uno de los campos considerados como las variables de entrada del modelo; posteriormente se normalizaron los campos.

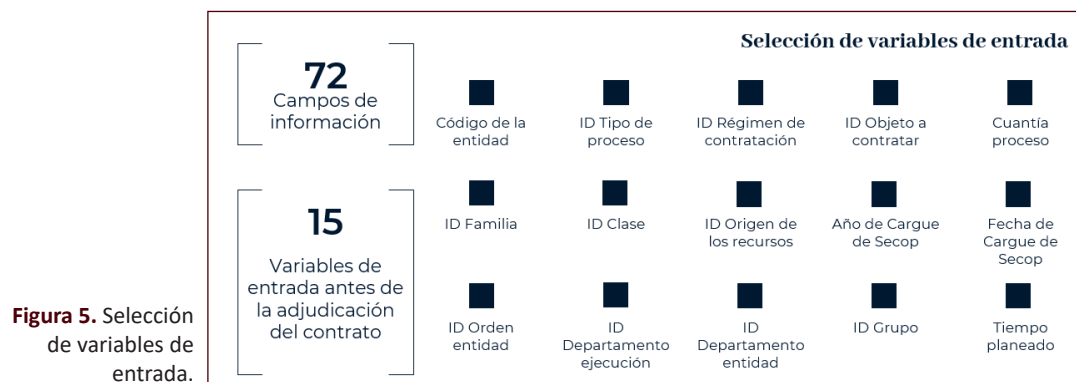


Figura 5. Selección de variables de entrada.

- Construcción de índice de éxito (salida): de acuerdo con el análisis de datos de la segunda fase, se estableció las fórmulas por usar en la construcción de tiempo, costo y alcance y se seleccionaron aquellos datos necesarios para la construcción del índice.

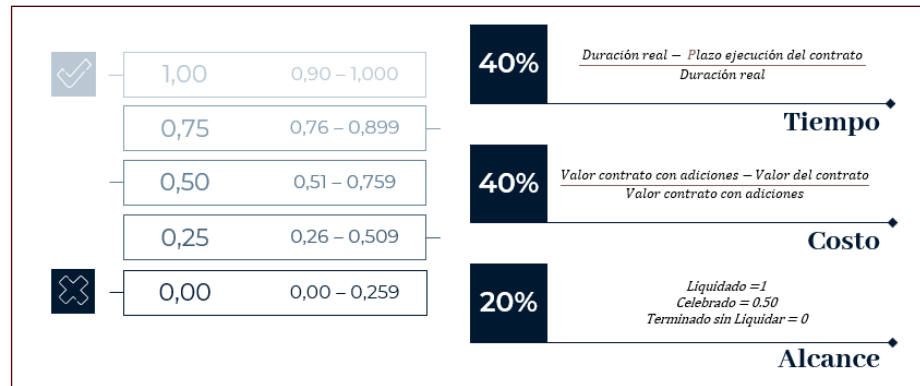


Figura 6. Construcción de índice de éxito (salida).

3.3 Diseño del modelo

De acuerdo con el enfoque, las aplicaciones y los usos de la inteligencia artificial y *Machine Learning*, en los cuales se profundizó en el capítulo 2.1 del marco conceptual, se determina el uso de un algoritmo de aprendizaje supervisado de la rama de la inteligencia artificial débil, en razón de que el objetivo del 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 entrenados a partir de unos datos históricos de proyectos, compuestos por unas entradas determinadas y la constitución de la salida a través de fórmulas que miden la triple restricción que, en este caso, corresponde al índice de éxito o fracaso. Posteriormente, al presentarle al modelo las entradas seleccionadas, éste predice la salida con base en los datos de entrenamiento.

El diseño del modelo se compone de la preparación de datos, y de su desarrollo y entrenamiento.

3.3.1 Preparación de datos

En la preparación de los datos se lee el conjunto de los que se obtienen en la fase de recopilación, su correspondiente distribución y normalización, de acuerdo con este proceso:

- Selección del formato del archivo (.csv) con la información recopilada.
- Uso de la librería Pandas, que permite manipular y analizar los datos. Clase `pandas.read_csv`.
- Uso de la librería Pandas, que permite visualizar el encabezado. Clase `pandas.head`.

En la separación de datos se tienen los de entrenamiento (95 %) y los de prueba (5 %).

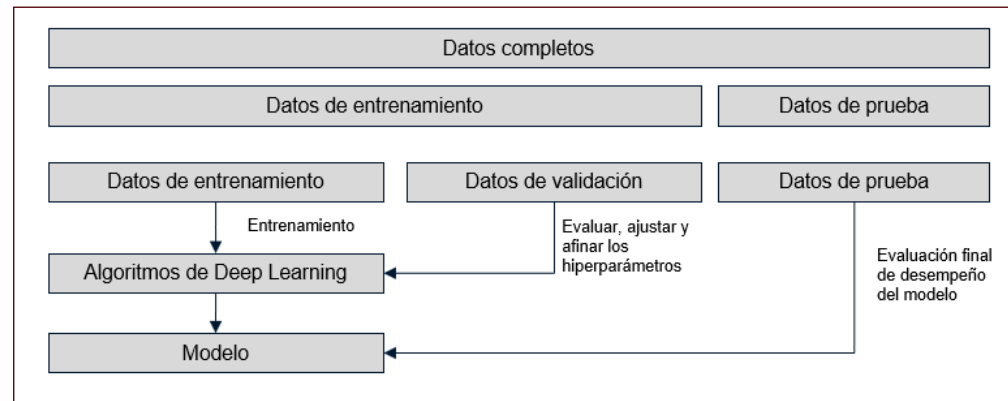


Figura 7. Separación de datos.

La normalización de datos consiste en escalar los datos a valores más pequeños, en rangos como $[-1, 1]$ o $[0, 1]$. Si presentan valores más grandes diferentes de dos binarios, es necesario normalizarlos; para esto se desarrolla:

- Uso del método `to_numpy()` de la librería Numpy, la cual permite extraer los datos en arreglos para el conjunto de datos de entrenamiento y almacenarlos en la variable `train_data`, y la data de validación en la variable `test_data`.
- Uso de la función `max`, que obtiene el valor máximo de todas las variables, para luego dividir cada uno de los arreglos por este valor, y dejar los datos normalizados de las entradas en dos variables: `x_train` (datos de entrenamiento) y `x_test` (datos de validación).

3.3.2 Desarrollo del modelo

Esta etapa constituye la parte más interesante: desarrollar el modelo que cumpla con las expectativas iniciales que, en últimas, significa responder a lo que se quiere obtener. También es la parte menos mecánica, y más compleja, pues aparte de decidir la arquitectura del modelo, se deben determinar los hiperparámetros que constituyen su configuración.

En la definición del modelo, se tiene la creación, la configuración y la visualización.

- Creación: método `sequential` de la librería Keras (número de neuronas de cada capa y tipo de activación). Tres capas densamente conectadas: una capa de entrada de los datos y una intermedia u oculta de 64 neuronas cada una, y una capa de salida, con una neurona.
- Configuración: método `compile` de la librería Keras (optimizador, la relación con la función de pérdida y las métricas para determinar la precisión de la red neuronal).
- Visualización de la arquitectura de la red: método `summary` de la librería Keras.

```
# 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
```

Figura 8. Visualización de arquitectura.

Fuente: MEP_20210218.

La configuración del modelo es la definición de la función de pérdida y el optimizador por utilizar.

- Optimizador: variantes u optimizaciones del algoritmo de descenso del gradiente. {RMSProp, AdaGrad, Adadelta, Adma, Adamax}. Primer argumento del método compile().
- Función de pérdida: función de pérdida del cálculo de error, basado en la métrica de rendimiento que se seleccione {binary_crossentropy}. Segundo argumento del método compile().
- Métricas de rendimiento: métrica de rendimiento dependiendo del tipo de problema que trata de resolver el modelo {accuracy, MSE, MAE}. Tercer argumento del método compile().
- MSE representa la raíz cuadrada del segundo momento de la muestra de las diferencias entre los valores previstos y los observados, o la media cuadrática de estas diferencias.

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

- MAE representa el error absoluto medio de la diferencia entre dos variables continuas, para cuantificar la precisión de una técnica de predicción, comparando los valores predichos frente a los observados, tiempo previsto.

$$\text{MAE} = \sqrt{\frac{1}{m} \sum |Y_{\text{estimada}} - Y_{\text{real}}|} \quad (2)$$

3.3.3 Entrenamiento del modelo

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 por entrenar el modelo, y un parámetro que permite visualizar el proceso de entrenamiento (*verbose*) de la siguiente manera:

```
In [20]: # 5. Entrenamiento de La Red Neuronal, Visualización 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
Epoch 6/50
```

Figura 9. Entrenamiento de la red neuronal.

Fuente: MEP_20210218.

La visualización del proceso de entrenamiento permite apreciar el resumen de la ejecución del proceso de entrenamiento por medio del objeto historial, retornado por el método `fit()` y especificando el número de Epochs finales por desplegar con el método *tail*.

```
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

Figura 10. Creación de Dataframe para desplegar el resultado de los Epochs.

Fuente: MEP_20210218.

En la predicción del proceso de entrenamiento se invoca el método `predict()` de la librería Keras. La predicción del conjunto de datos de entrenamiento se almacena en el dataset `y_predict_train`.

En la evaluación del proceso 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 durante el entrenamiento, y así 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 la evaluación se usó el método `evaluate()` de la librería Keras, el cual recibe como argumentos el conjunto de datos que se quiere evaluar: entradas y salida. En

este caso se refiere a los conjuntos de entrada y salida de los datos de entrenamiento. El resultado de la evaluación lo retorna en las variables tLoss y tValAccuracy.

```
In [24]: # Evaluación del modelo a partir 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
```

Figura 11. Evaluación del modelo a partir de los datos de entrenamiento.

Fuente: MEP_20210218.

En el proceso de evaluación del modelo, si los resultados de pérdida y precisión tanto para el conjunto de datos de entrenamiento como para el 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.

3.4 Evaluación del modelo

La evaluación del modelo se desarrolla a partir de datos de prueba, datos 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 que no se le han mostrado.

Para ello se puede utilizar el método `evaluate()`, que 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 los conjuntos de datos para entrenar el modelo en uno de entrenamiento y otro 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 13, se usa el 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`.

```
In [24]: # Evaluación del modelo a partir 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
```

Figura 12. Evaluación del modelo a partir de los datos de entrenamiento.

Fuente: MEP_20210218.

4. RESULTADOS

La evaluación del modelo pretende cargar los datos de prueba reservados, de manera que se determine la precisión de la predicción del modelo, para lo cual es necesario llevar a cabo las subfases de inferencia o predicción (datos de prueba) y de evaluación del modelo con estos últimos.

4.1 Inferencia o predicción data de prueba

Contempla la predicción de la salida para el conjunto de datos de prueba, es decir, se recogen los frutos del trabajo 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, se trata de poner a trabajar el modelo, recordando que lo que motiva su diseño es dar una respuesta al problema planteado con la mayor confiabilidad posible.

4.2 Evaluación del modelo con los datos de prueba

Identifica el porcentaje de precisión del modelo, como resultado de la predicción de los datos de prueba, es decir, evalúa el resultado de la eficiencia del modelo.

```
In [26]: # Evaluación del modelo a partir 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: 0.29340210913974
Accuracy: 0.7972792983055115

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

Figura 13. Evaluación del modelo a partir de los datos de prueba.

Fuente: MEP_20210218.

5. CONCLUSIONES

1. El modelo producto de esta investigación hace posible predecir el éxito de los proyectos usando herramientas y algoritmos del Machine Learning y datos histórica de proyectos ejecutados.
2. El modelo desarrollado es una herramienta que proporciona a la evaluación de proyectos un indicador diferente de las métricas financieras tradicionales, como

- TIR, VPN, rentabilidad del negocio, ROI y C-B (costo-beneficio), para la toma de decisiones, con el fin de minimizar la tasa de proyectos fracasados.
3. En la última década, más de 1000 contratos de obra para la construcción de infraestructura en Colombia, con una ejecución de más de 5000 millones de pesos resultaron fallidos. De haber utilizado la predicción del modelo, el Estado podría haber invertido estos recursos en otros proyectos.
 4. De haber sido utilizado este modelo por parte del Estado, más de 45.000 proyectos no exitosos, con costos superiores a los 60.000 millones de pesos hubieran sido reestructurados o rechazados antes de iniciarlos.

6. RECOMENDACIONES Y TRABAJO FUTURO

De acuerdo con el documento 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 en el valor del modelo y el resultado de esta investigación, se puede incursionar en la creación de un negocio enfocado en la implementación de herramientas de Machine Learning en la gestión de proyectos, mediante la creación de modelos de diagnóstico y predicción, según las necesidades de las organizaciones que estén alineadas con las innovaciones.
- Se puede indagar en las entidades ejecutorias sobre información adicional a la almacenada en la plataforma, que se pueda usar como fuente confiable para plantear una nueva microfórmula 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. Esto no sólo 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 pueden utilizar las aplicaciones del Machine Learning para brindar un servicio a las empresas, de modo que se detecten pérdidas o deficiencias en su cadena de producción; y una herramienta muy poderosa y atractiva que, con base en un levantamiento de datos, permita modelar una RN para predecir resultados y adoptar mejoras en los flujos donde se detecten estas diferencias.

REFERENCIAS

- Anaconda Inc. (2021). *Anaconda. Documentation*. <https://docs.anaconda.com/anaconda/navigator/glossary/>
- Colombia Compra Eficiente. (2020). <https://www.colombiacompra.gov.co/secop/secop-i>
- Escuela Colombiana de Ingeniería Julio Garavito. (s.f.). Biblioteca/Base de datos. <https://escuelaing.metaproxy.org/subjects/databases.php?letter=S>
- Goodfellow, I., Bengio, Y., & Corville, A. (2016). *MIT Press*. <http://www.deeplearningbook.org>
- Hetch-Nielsen, R. (1988). Neurocomputing: Picking the human brain. *IEEE Spectrum*, 13-18.
- IBM. (1997, 11 de mayo). *IBM100*. Icons of Progress. <https://www.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/>
- Khepri, W. (2018, 2 de noviembre). *Redes Neuronales, ¿qué son? Introducción a las redes neuronales*. Medium. <https://medium.com/@williamkhepri/redes-neuronales-que-son-a64d022298e0>
- Laurence Goasduff. (2017, 20 de septiembre). *Gartner*.

<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*. <https://www.datos.gov.co/Gastos-Gubernamentales/SECOP-I/xvdy-vvsk>

MIT. (2017, 7 de junio). *MIT Technology Review*. <https://www.technologyreview.es/s/7927/el-dia-que-la-humanidad-fue-derrotada-por-una-inteligencia-artificial>

Open AI. (2019, febrero). *Better language models and their applications*. <https://openai.com/blog/better-language-models/>

OpenDoor Technology. (2019, 20 de febrero). <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). 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*. Project Management Institute.

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

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

Project Management Institute PMI®. (2019). *Pulse of the profession. El futuro del trabajo: liderar con PMTQ*.

Roberts, A., & Wallace, W. (2014). *Gestión de proyectos*.

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

The Standish Group. (2015). *The chaos report*. https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

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