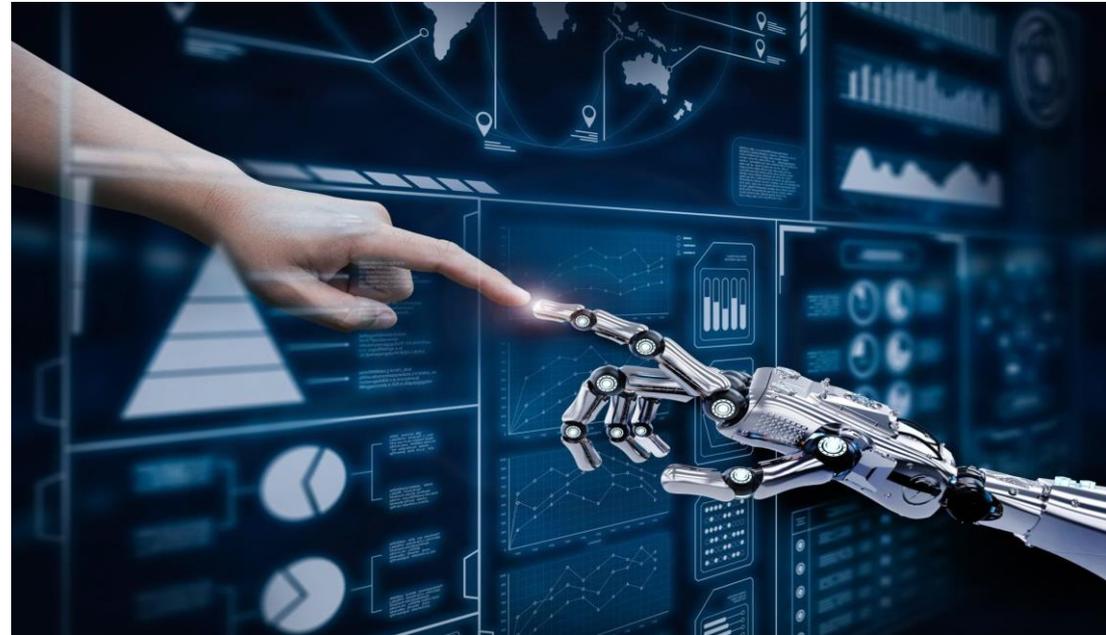


DISEÑO DE UN MODELO DE EVALUACIÓN DE PROYECTOS A PARTIR DE HERRAMIENTAS DE MACHINE LEARNING O APRENDIZAJE AUTOMATIZADO.

Maestría en Desarrollo y Gerencia
Integral de Proyectos

Ing. M.Sc. Rodrigo Buzeta

Ing. Jorge Sebastian Caro Mesa
Ing. Camilo Andrés Cruz Rodríguez
Arq. Laura Tatiana Nova Barreto



Contenido

01

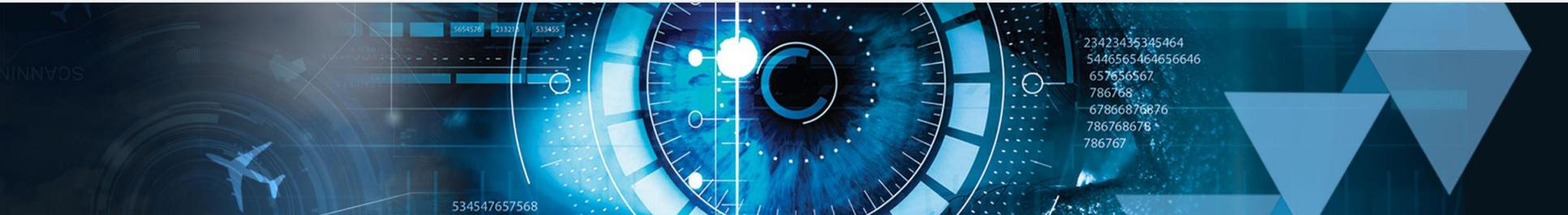
Contexto

Evaluación de proyectos y Machine Learning.

02

Justificación

Planteamiento del problema, justificación y objetivos.



03

Diseño metodológico

Desarrollo y enfoque de la investigación.

04

Gerencia del Proyecto

Definición, control y seguimiento y cierre.

01

Contexto

Evaluación de proyectos
y Machine Learning.



Evaluación de Proyectos



“

Para el año 2019, un 31,1% de los proyectos fracasaron o fueron cancelados, un 52,7% superaron en costo, tiempo y/o funcionalidad prometida, y solo el 16,2% se consideraron exitosos. Estas cifras confirman la situación crítica en el desempeño de proyectos”

—OpenDoor Technology, 2019

Machine Learning



“

... 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



02

Justificación

Planteamiento del problema, justificación y objetivos.

Problema

Proyectos con alta tasa de fracaso

Efectos

- Incumplimiento del alcance
- Incumplimiento en costo y tiempo
 - Materialización de riesgos, con consecuencias significativas
 - Desempeño deficiente



Causas

- Corrupción del alcance, cambios en costo y tiempo.
- Ausencia de indicadores de desempeño.
- Uso inadecuado de las buenas prácticas de la gerencia de proyectos.
- Ausencia de retroalimentación al finalizar el proyecto.
- Desconocimiento de los problemas o situaciones críticas.
- Reincidencia en fallas e incumplimientos.
- Falta de gestión de cambios.
- Falta de gestión de riesgos.
- Omisión de lecciones aprendidas.

Justificación



Pregunta de investigación

¿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 de proyectos en el cual se prediga la tasa de éxito de un proyecto?

Oportunidad



“... 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)

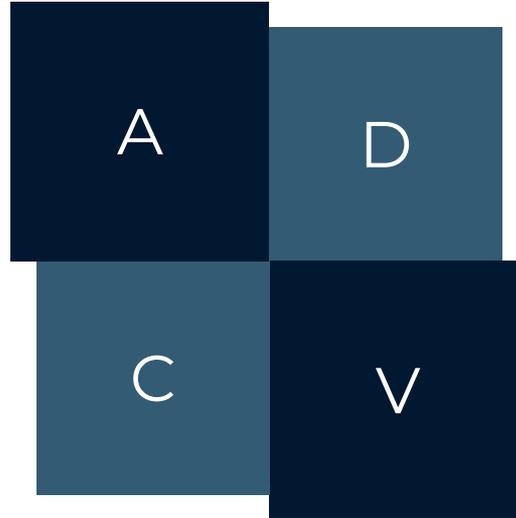
Objetivos

Analizar

Analizar la data inicial, de manera que se asegure la eficacia de la información.

Consolidar

Consolidar la data de entrenamiento y validación del modelo, incluyendo variables de entrada y cálculo de índice de éxito o salida.



Determinar

Determinar la alternativa más adecuada entre las herramientas y algoritmos existentes del Machine Learning.

Validar

Validar el modelo con la data reservada para tal fin y el análisis correspondiente.

Objetivo general

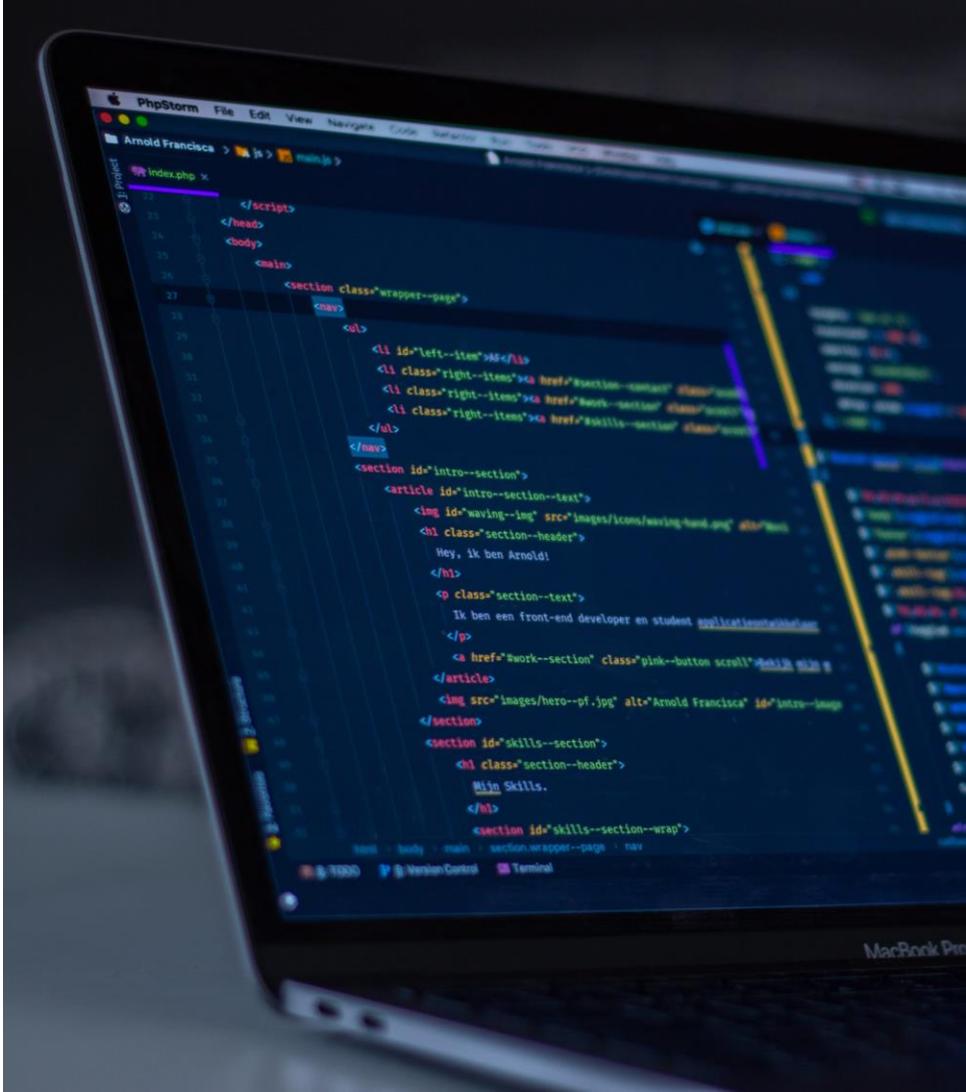
Diseñar un modelo de evaluación de proyectos a partir de herramientas y algoritmos existentes de Machine Learning o aprendizaje automatizado, que permita predecir el índice de éxito de los mismos.



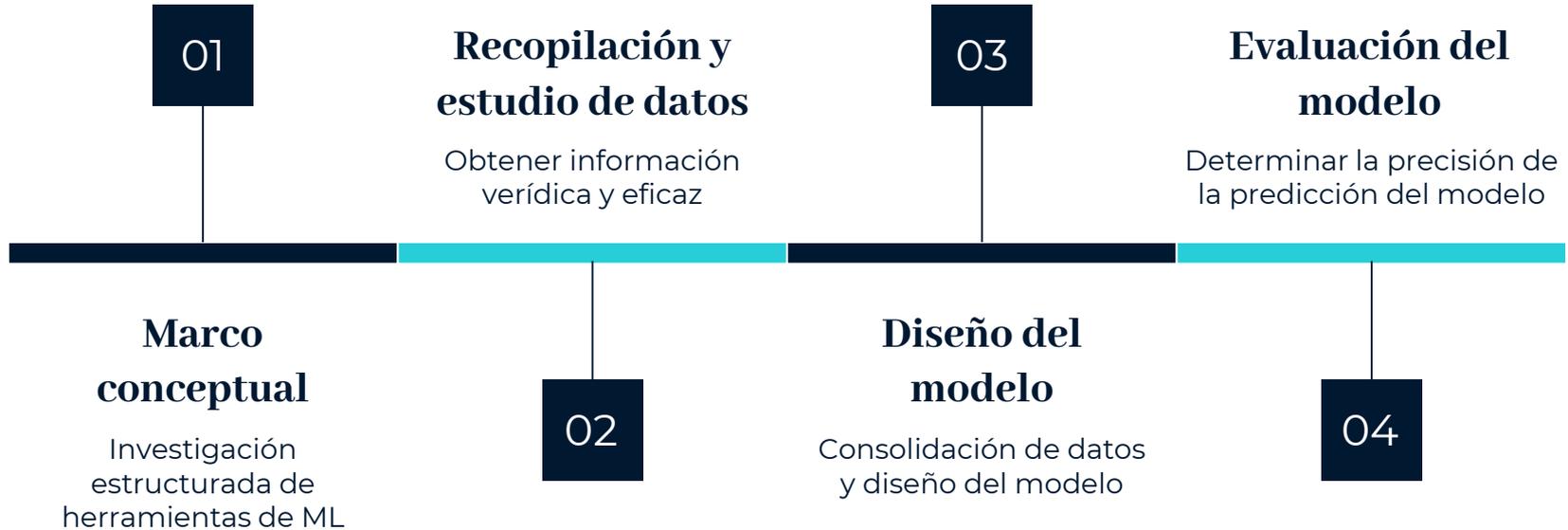
03

Diseño metodológico

Desarrollo y enfoque de
la investigación.



Diseño metodológico



Marco conceptual

Inteligencia Artificial

Técnicas que permiten imitar la inteligencia humana.



Machine Learning

Técnicas que permiten mejorar en tareas específicas con experiencia.



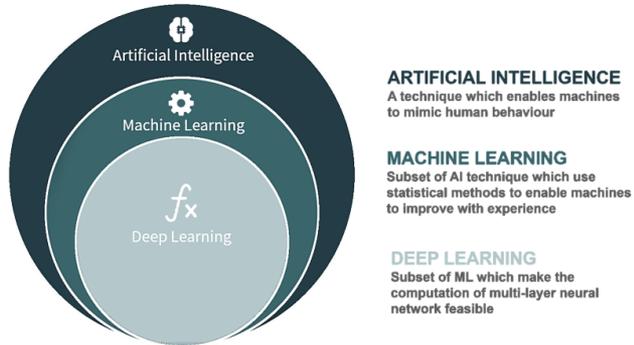
Deep Learning

Algoritmos que permiten que el software se entrene a sí mismo para realizar tareas

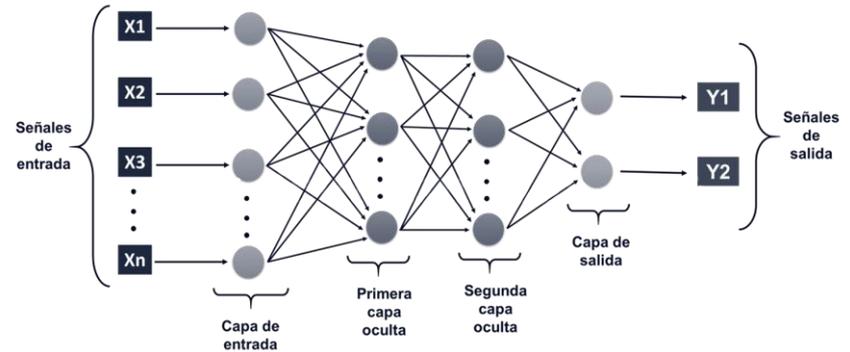


Marco conceptual

Diagrama de relación entre Inteligencia Artificial, Machine Learning y Deep Learning.



Red Neuronal, con dos capas ocultas en donde cada uno de los círculos de color representa un nodo o neurona.





Inteligencia Artificial

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.

IA D

IA Débil

Asistente para el ser humano en las actividades que cotidianamente realiza . 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.

IA F

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 reemplacen al ser humano





Machine Learning

Subconjunto de la IA que proporciona a los computadores la capacidad de aprender, sin que un programador proporcione las reglas para producir un resultado.

AS

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.

ANS

Aprendizaje no supervisado

Los modelos a entrenar no incluyen etiquetas, por lo tanto, el algoritmo es el que clasifica la información.

AR

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.





Deep Learning

Basado 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.

RNA

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.

RPB

Redes Profundas Básicas

Redes neuronales que comparten algunas propiedades básicas y consisten en un número de neuronas interconectadas en diferentes capas. Su diferencia radica en la arquitectura de la red, cómo se encuentran interconectadas y su forma de entrenamiento.



Recopilación y estudio de datos



Recolección de datos

Búsqueda de fuentes o bases de datos históricas de proyectos ejecutados



Depuración de datos

Criterios de verificación y depuración de la data disponible.



Análisis de datos

Selección previa de los campos que se infiere que pueden llegar a contribuir en el diseño

Recolección de datos



9.660.410
Procesos de contratación



Construcción
Sector económico con
mayor crecimiento.



Depuración de datos

Año de Cargue SECOP

2010 - 2020

9.106.590

Objeto a Contratar

Servicios de edificación, construcción de instalaciones y mantenimiento



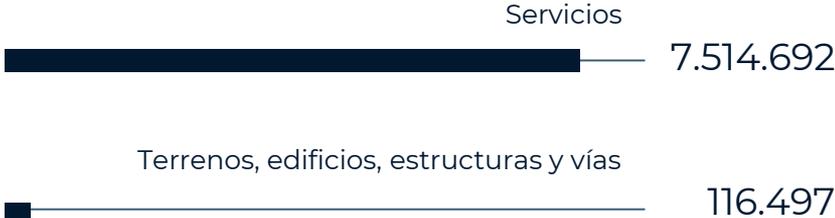
440.173

Terrenos, edificios, estructuras y vías



116.497

Grupo



Tipo de Contrato



Análisis de datos

72
Campos de
Información

Selección de variables de entrada

■
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

Antes de la
adjudicación del
contrato

■
ID Orden
Entidad

■
ID
Departamento
Ejecución

■
ID
Departamento
Entidad

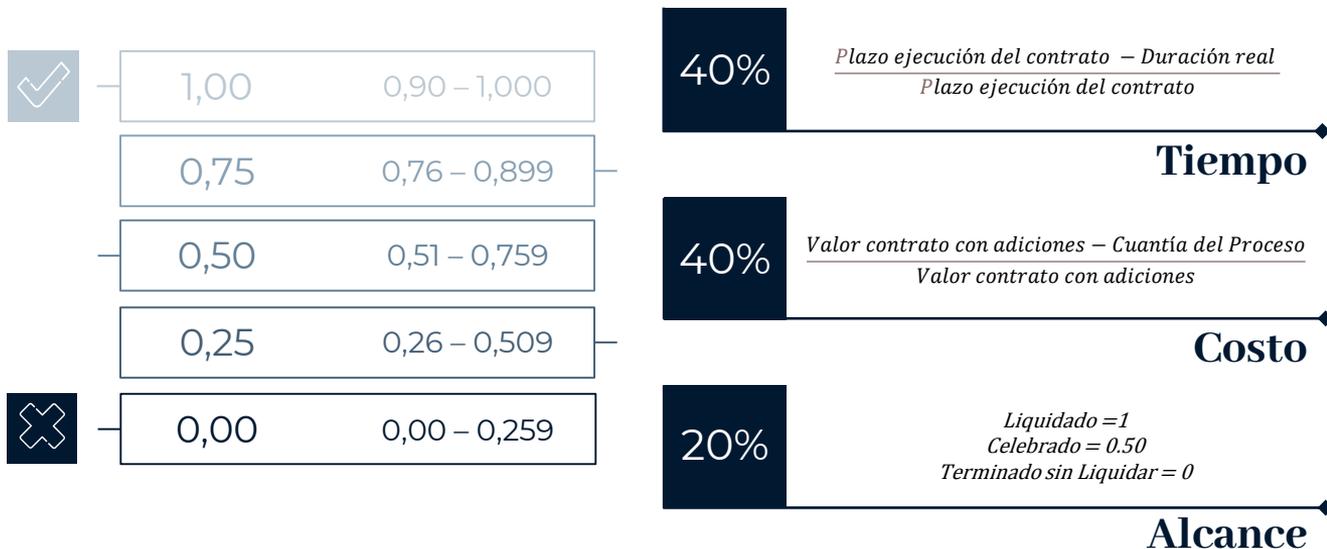
■
ID Grupo

■
Tiempo
planeado



Análisis de datos

Construcción de índice de éxito (salida)



Diseño del modelo

Preparación de datos

Lectura del conjunto de datos,
distribución y normalización.



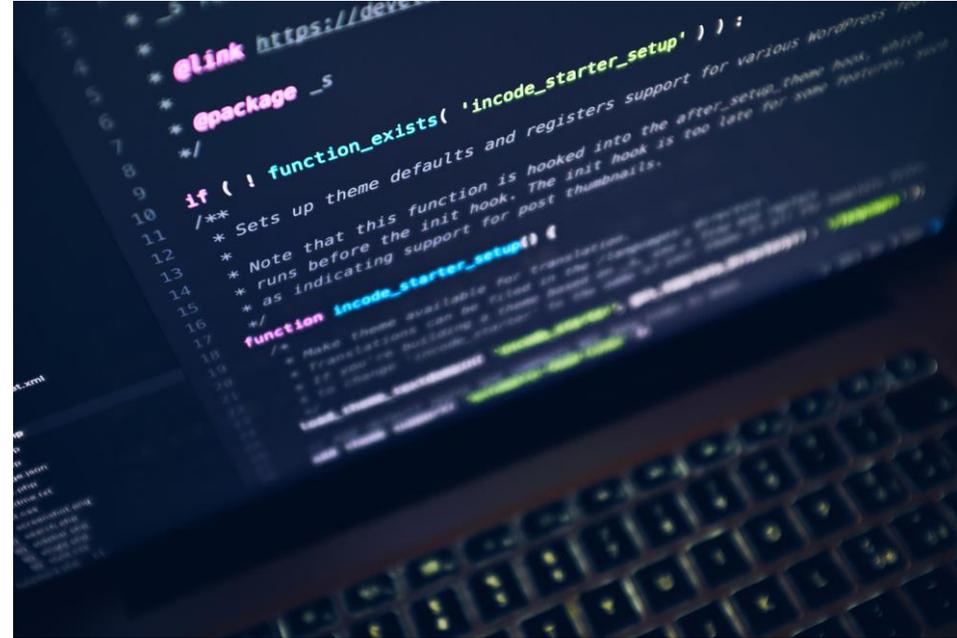
Desarrollo del modelo

Definición del modelo, configuración
y entrenamiento



Entrenamiento

Entrenamiento del modelo y
predicción con la data destinada



Preparación de datos

01

Lectura de Datos

- Seleccionar el formato del archivo (.CSV)
- Uso de la librería **Pandas** (Pandas) que permite la manipulación y análisis de datos. Clase `pandas.read_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)
```

- Uso de la librería **Pandas** (Pandas) que permite visualización del encabezado. Clase `pandashead`

```
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	Cuantidad_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

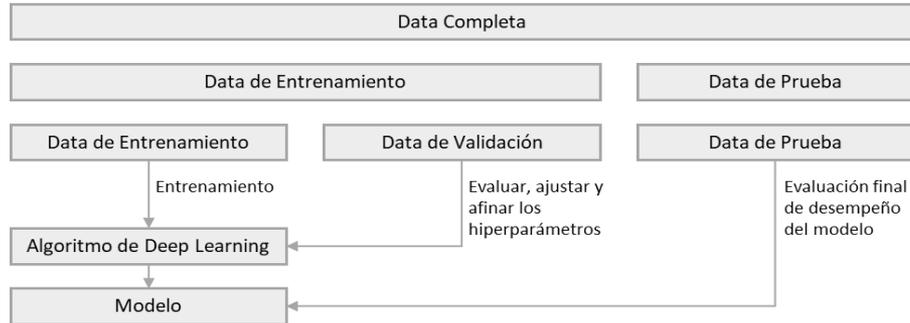


Preparación de datos

02

Separación de Datos

Data Completa = Data de Entrenamiento (95%) + Data de prueba (5%)



Data de Entrenamiento (95%)
=
Data de Entrenamiento (80%)
+
Data de Validación (20%)

- Uso del método **sample** de la librería **Pandas** (Pandas) que separa ($\text{frac} = 0.8$) `df_full` -> `train_df`
- Uso del método **drop** de la librería **Pandas** (Pandas) para obtener el complemento `df_full` -> `test_df`

```
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)
```



Preparación de datos

03

Normalización de Datos

- Se utiliza el método `to_numpy()` de la librería Numpy, la cual permite extraer los datos en arreglos para el conjunto de Datos Entrenamiento y almacenarlos en la variable `train_data`, y la Data de Validación en la variable `test_data`

```
# 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
```

- Invocar 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 2 variables: `x_train` (Data Entrenamiento) y `x_test` (Data de Validación)

```
# 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
```



Desarrollo del modelo

01

Definición del modelo

- **Creación** = Método **sequential** de la librería **Keras** (número de neuronas de cada capa y el tipo de activación). 2 capas densamente conectadas: **1 capa de entrada** de los datos y **1 capa intermedia u oculta** de 64 neuronas cada una; y **1 capa de salida**, con 1 neurona.

```
# 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'),
```

- **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).

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

- **Visualización 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
```



Desarrollo del modelo

02

Configuración del modelo

- **Optimizador** = variantes u optimizaciones del algoritmo de descenso del gradiente. {**RMSProp**, **AdaGrad**, **Adadelata**, **Adma**, **Adamax**}. Primer argumento del método **compile()**.

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

- **Función de Pérdida** = Función de Pérdida del calculo 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()**.

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

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

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

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



Entrenamiento del modelo

01

Visualización del proceso de Entrenamiento

El proceso de **entrenamiento** invoca el método **fit()** de la librería **Keras**, que recibe como argumentos:

- 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**.
- Número de EPOCHS a entrenar el modelo.
- Parámetro que permite visualizar el proceso de entrenamiento (**verbose**).

```
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
```

```
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

Afinamiento *Tuning* para evitar el sobreentrenamiento *overfitting*



Entrenamiento del modelo

02

Predicción del proceso de Entrenamiento

El proceso de predicción o inferencia de la salida del modelo. 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`

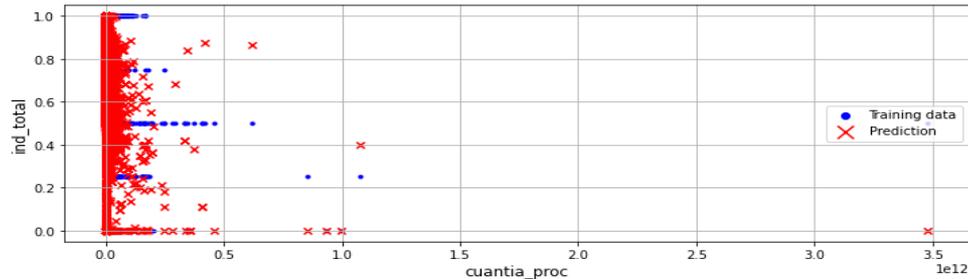
```
In [23]: # Predicción del modelo vs Data de Entrenamiento y desescalamiento
# Visualización de La Salida de La data de Entrenamiento VS La variable de Entrada Cuantía del Proceso

y_predict_train=model.predict(x_train)
```

Visualización Gráfica de las predicciones del Modelo.

```
# 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()
```



Entrenamiento del modelo

03

Evaluación del proceso de Entrenamiento

Se invoca el método `evaluate()` de la librería **Keras**, el cual recibe como argumentos el conjunto de datos que se quiere evaluar: Entradas y Salida; para este caso, los conjuntos de entrada y salida de la data 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
```



Evaluación del modelo



Inferencia o predicción data de prueba

Contempla la predicción de la salida para el conjunto de datos de prueba.



Evaluación del modelo con la data de prueba

Indica el porcentaje de precisión del modelo, como resultado de la predicción de la data de prueba.

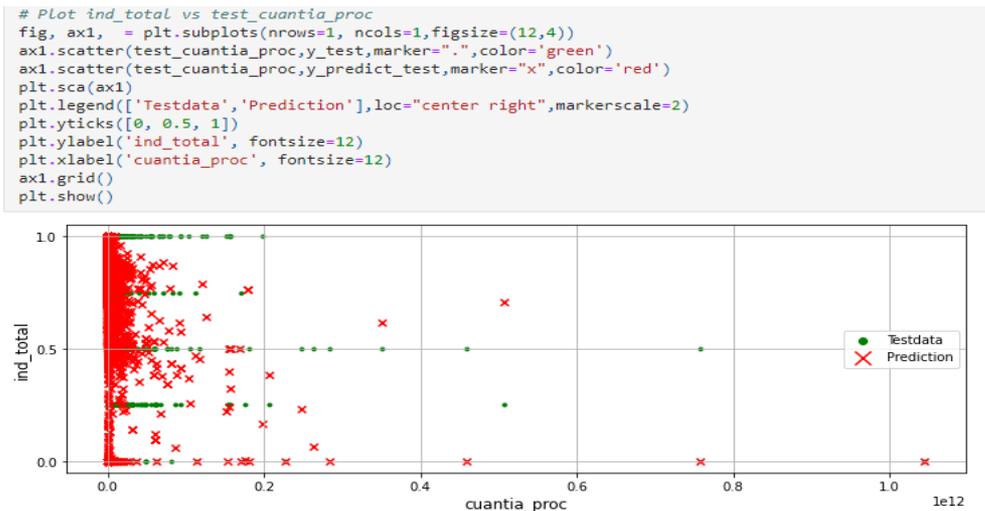
Evaluación del modelo



Inferencia o predicción de data de prueba

La fase de inferencia o predicción es aquella en donde, sobre la puesta a punto del modelo, se comprueba que está preparado para predecir una respuesta a la pregunta planteada inicialmente, con un nivel de precisión adecuado.

Visualización Gráfica de las predicciones del modelo.



Evaluación del modelo



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`.

```
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]]
```



Entorno de desarrollo

Anaconda Navigator

Ambiente de trabajo

Jupyter Notebook

Entorno de trabajo

Numpy - Pandas -

Keras

Librerías

The screenshot displays the Anaconda Navigator web interface. At the top, the logo 'ANACONDA NAVIGATOR' is visible on the left, and a 'Sign in to Anaconda Cloud' button is on the right. Below the header, there's a navigation sidebar on the left with options: Home, Environments, Learning, and Community. The main content area shows a grid of application cards under the heading 'Applications on base (root) Channels'. Each card includes an icon, the application name, version number, a brief description, and a button to either 'Launch' or 'Install' the application. The applications shown are:

- jupyterlab** (0.32.1): An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. (Launch)
- jupyter notebook** (5.5.0): Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. (Launch)
- qtconsole** (4.3.1): PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. (Launch)
- spyder** (3.2.8): Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. (Launch)
- vscode** (1.27.1): Streamlined code editor with support for development operations like debugging, task running and version control. (Launch)
- glueviz** (0.13.3): Multidimensional data visualization across files. Explore relationships within and among related datasets. (Install)
- orange3** (3.13.0): Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. (Install)
- rstudio** (1.1.423): A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks. (Install)

At the bottom of the sidebar, there are links for 'Documentation', 'Developer Blog', and 'Feedback', along with social media icons for Twitter, YouTube, and GitHub.



Conclusiones



Identificar proyectos que pueden llegar a **generar valor**, disminuyendo las tasas de fracaso y aumentando la posibilidad de lograr un mayor retorno de la inversión.



Es posible predecir el éxito de los proyectos haciendo uso de herramientas y algoritmos del **Machine Learning** y **data histórica** de proyectos ejecutados.



Herramienta

que proporciona a la **evaluación de proyectos** un indicador diferente a las métricas financieras tradicionales, para la toma de decisiones.

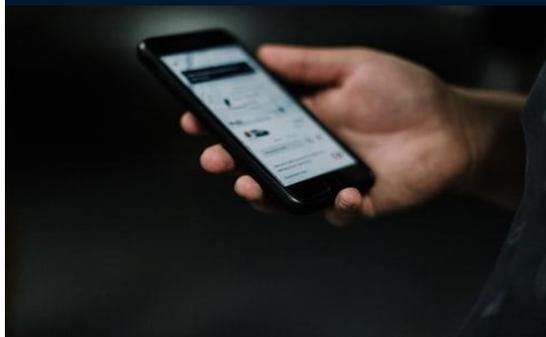
Conclusiones

Si el estado hubiera usado este modelo en la última década...



Se hubieran podido destinar a otros proyectos, **recursos superiores a los 5.000 millones de pesos**, que fueron ejecutados en **más de 1.000 contratos por año** que resultaron **fracasados** en la construcción de infraestructura en Colombia.

Más de **45.000 proyectos no exitosos**, con costos superiores a **60.000 millones de pesos**, se hubieran podido reestructurar o rechazar.



Trabajos futuros

Crear indicadores de evaluación del cumplimiento del alcance, lo que aportará a la creación y precisión de la tasa de éxito.

Evaluar el modelo con otro sector económico ratificará el nivel de precisión del modelo.

Incentivar la creación de bases de datos, que contengan información de calidad y, a su vez, indicadores de evaluación de proyectos históricos.



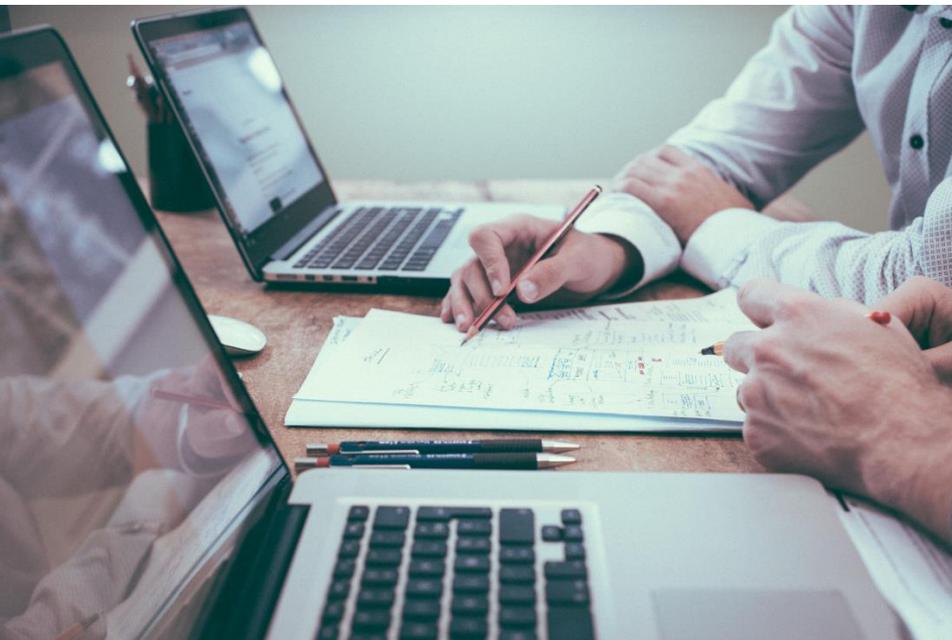


04

Gerencia del Proyecto

Definición, control y
seguimiento y cierre.

Libro de Gerencia del Proyecto



Inicio



Planeación



Seguimiento y control



Cierre

Inicio

Alcance

Diseño de un modelo de evaluación de proyectos, a partir de herramientas y algoritmos existentes de Machine Learning o aprendizaje automatizado, que permita obtener una predicción del éxito del proyecto.

Estimados



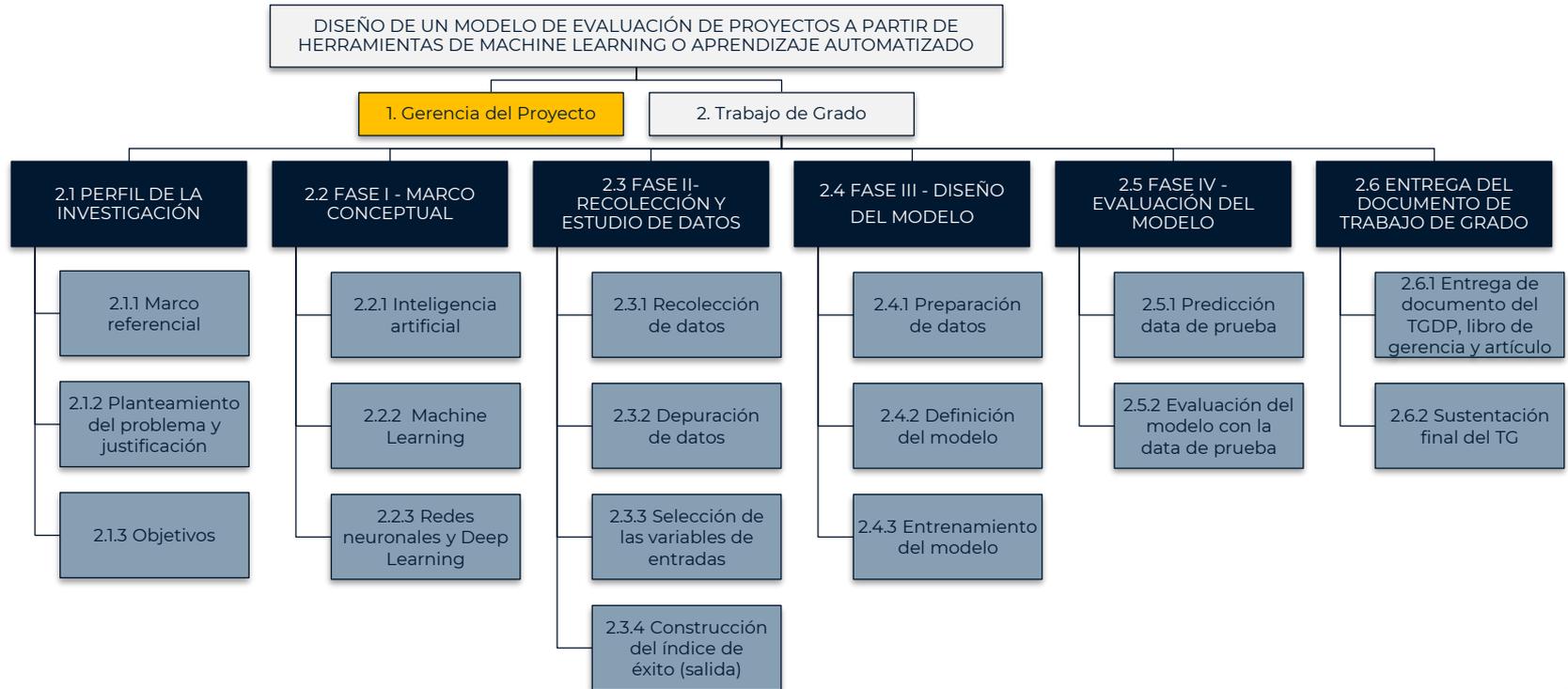
325 días



\$89.860.823



Planeación - WBS



Seguimiento y control



Comunicaciones

Costo



Calidad

Cronograma



Comunicaciones

17

Reuniones virtuales de seguimiento



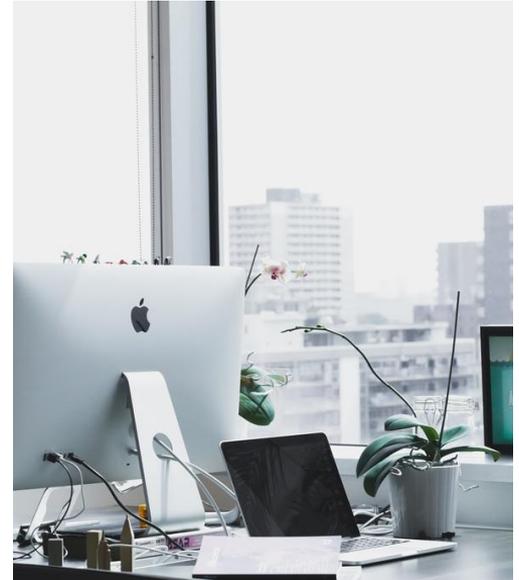
3

Reuniones virtuales con experto



8

Correos electrónicos



Control de Costo

Costo Estimado:

\$89.860.823

Costo Ejecutado:

\$83.860.823

93%

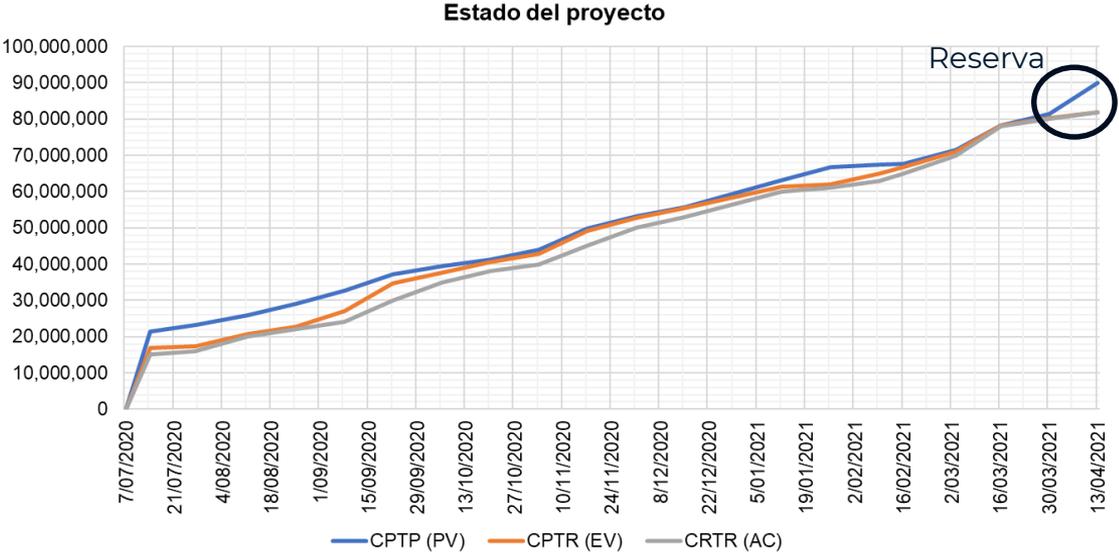


% Completado

Nombre de tarea	Valor planeado: PV (CPTP)	Valor acumulado: VA (CPTR)	% físico
▸ DISEÑO DE UN MODELO DE EVALUACIÓN DE PROYECTOS A PARTIR DE HERRAMIENTAS DE MACHINE LEARNING O APRENDIZAJE AUTOMATIZADO	\$89,860,823	\$83,860,823	93%
▸ GERENCIA DEL PROYECTO	\$16,894,805	\$16,894,805	100%
Gerencia del Proyecto	\$16,894,805	\$16,894,805	100%
▸ TGDP	\$72,966,018	\$66,966,018	92%
▷ PERFIL DE LA INVESTIGACIÓN	\$10,455,016	\$10,455,016	100%
▷ FASE I - MARCO CONCEPTUAL	\$13,419,280	\$13,419,280	100%
▷ FASE II - RECOLECCIÓN Y ESTUDIO DE DATOS	\$9,738,496	\$9,738,496	100%
▷ FASE III - DISEÑO DEL MODELO	\$10,491,176	\$10,491,176	100%
▷ FASE IV - EVALUACIÓN DEL MODELO	\$8,549,640	\$8,549,640	100%
▷ ENTREGA DEL DOCUMENTO DE TRABAJO DE GRADO	\$20,312,410	\$14,312,410	70%



Control de Costo



\$8.000.000

Reserva

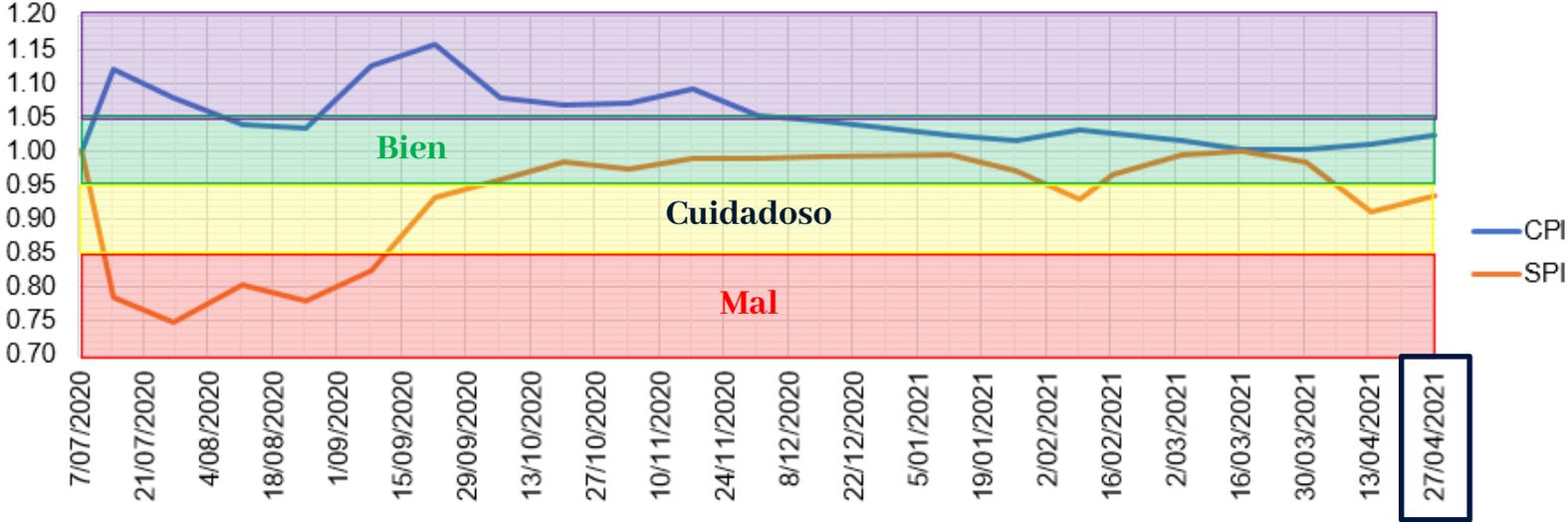
\$2.000.000

Reserva Usada



Control de Calidad

Indices de desempeño



CPI 1.01

SPI\$ 0.93



Control de Cronograma

Fecha inicio:

17 enero de 2020

Fecha fin:

14 mayo de 2021

22

Controles de seguimiento al cronograma

Nombre de tarea	Comienzo	Fin	Duración
▸ DISEÑO DE UN MODELO DE EVALUACIÓN DE PROYECTOS A PARTIR DE HERRAMIENTAS DE MACHINE LEARNING O APRENDIZAJE AUTOMATIZADO	17/01/2020	14/05/2021	325 días
▸ GERENCIA DEL PROYECTO	17/01/2020	8/04/2021	299 días
Gerencia del Proyecto	17/01/2020	8/04/2021	299 días
▸ TGDP	17/01/2020	14/05/2021	325 días
▸ PERFIL DE LA INVESTIGACIÓN	17/01/2020	17/07/2020	123 días
▸ FASE I - MARCO CONCEPTUAL	9/06/2020	8/09/2020	60 días
▸ FASE II - RECOLECCIÓN Y ESTUDIO DE DATOS	9/09/2020	5/11/2020	40 días
▸ FASE III - DISEÑO DEL MODELO	6/11/2020	24/12/2020	33 días
▸ FASE IV - EVALUACIÓN DEL MODELO	28/12/2020	26/01/2021	20 días
▸ ENTREGA DEL DOCUMENTO DE TRABAJO DE GRADO	19/02/2021	14/05/2021	57 días



Cierre

Lecciones aprendidas



Generar informes de desempeño, con la misma frecuencia, durante los periodos de descanso entre los semestres.



Verificar que los documentos de trabajo desarrollados en la nube, en este caso Google Drive, estén debidamente cargados con la última actualización.



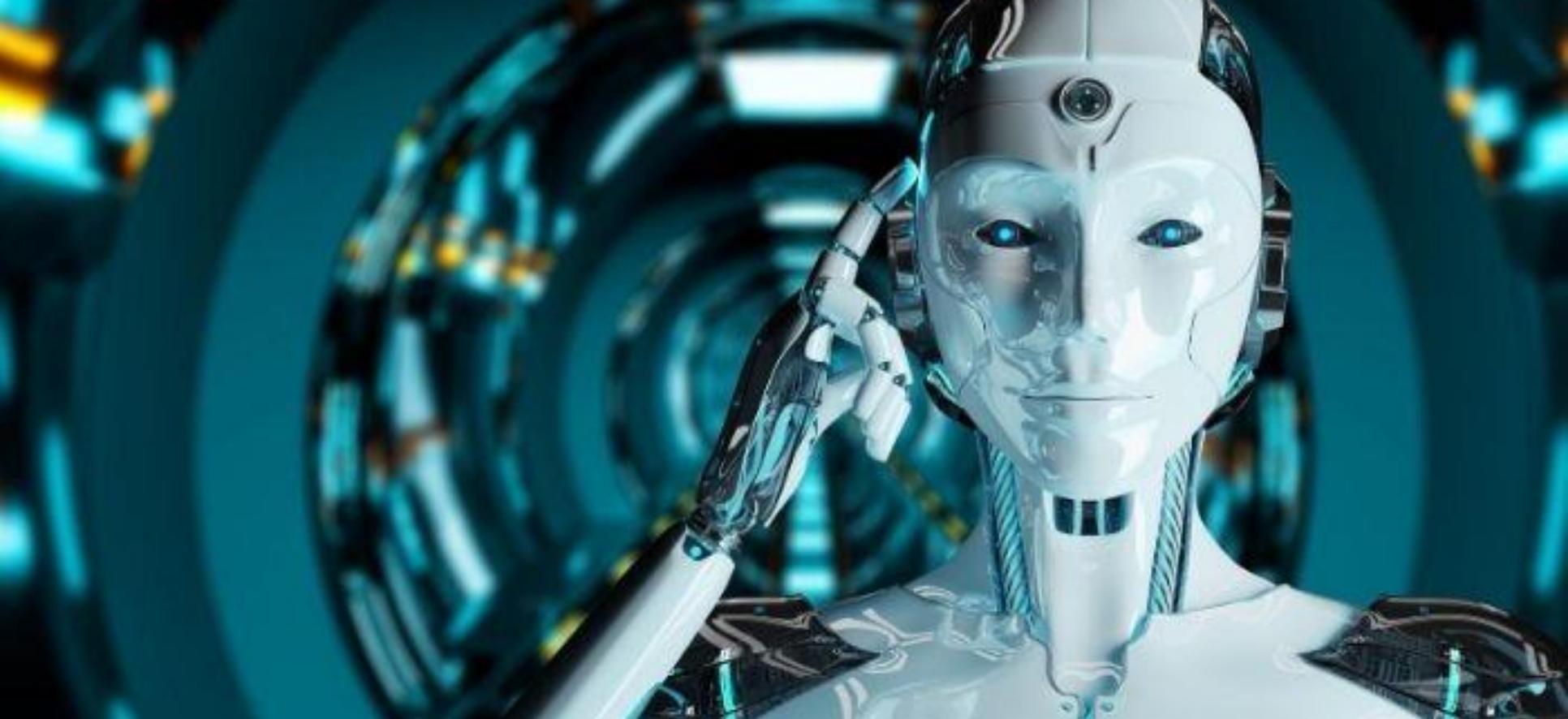
De acuerdo con el grado de complejidad del modelo de Machine Learning y la cantidad de datos a procesar, es importante obtener una configuración óptima de computador.

Acta de cierre



Se formaliza con la aceptación del trabajo de grado mediante la firma de acta de cierre.





Gracias

Maestría en Desarrollo y Gerencia
Integral de Proyectos

Referencias

-
- 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>
- 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>
- 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>
- 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
- 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.
- 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
- Unsplash.(2021). Obtenido de Unsplash. Fotos para todos.: <https://unsplash.com/collections>