



**Tecnológico  
de Monterrey**

ITESM- Campus Puebla

**NAATIK | CRISP DM**

**Inteligencia artificial avanzada para la ciencia de datos  
II**

**Integrantes Equipo 1:**

Myroslava Sánchez Andrade A01730712  
José Antonio Bobadilla García A01734433  
Karen Rugerio Armenta A01733228  
Alejandro Castro Reus A01731065

Fecha: 07/10/2022

# Índice

<b>1 Business Understanding</b>	<b>4</b>
1.1 Determine Business Objectives	4
Background	4
Objetivos de negocio	4
Criterios de éxito	4
1.2 Assess Situation	5
Inventory of Resources	5
Requirements, Assumptions and Constraints	6
Risks and Contingencies	6
Terminology	7
Costs and Benefits	8
1.3 Determine Data Mining Goals	9
Data Mining Goals	9
Data Mining Success Criteria	9
1.4 Produce Project Plan	10
<b>2 Data Understanding</b>	<b>10</b>
2.1 Collect Initial Data	10
Initial Data Collection	10
2.2 Collect Initial Data	11
Data Description	11
2.3 Explore Data	12
Data Exploration	12
2.4 Verify data	13
Data Quality	13
<b>3 Data Preparation</b>	<b>14</b>
3.1 Selecting Data	14
Dataset Description	14
3.2 Cleaning Data	14
3.3 Constructing New Data	15
3.4 Formatting Data	15
<b>4 Modeling</b>	<b>16</b>
4.1 Selecting Modeling Techniques	16

4.2 Building and Assessing the model	16
4.3 Clustering	25
<b>5 Evaluation</b>	<b>30</b>
5.1 Evaluating the results	30
5.1 Review process	30
5.1 Determining the next steps	30
<b>6 Deployment</b>	<b>30</b>
6.1 Planning for deployment	31
6.2 Planning monitoring and maintenance	35
6.3 Producing a final report	35
6.4 Conducting a final project review	36
<b>Referencias</b>	<b>37</b>

# 1 Business Understanding

## 1.1 Determine Business Objectives

### ***Background***

Como parte de nuestro proyecto final de la materia *Inteligencia Artificial Avanzada para la Ciencia de Datos*, se nos asignó una problemática dada por la empresa NAATIK. Esta es una empresa enfocada en el desarrollo y aplicación de Inteligencia Artificial y Ciencia de Datos para brindar soluciones.

La problemática planteada consiste en el análisis de un conjunto de datos de una compañía telefónica que contiene los siguientes datos: servicios contratados de clientes, información de la cuenta de los clientes, información demográfica de los clientes, y la permanencia del cliente en el último mes (booleano).

La entrega para esta problemática consiste en la predicción de la permanencia de un grupo cliente dados sus datos de una empresa de telefonía, y análisis de clientes para el desarrollo de un modelo de retención.

### ***Objetivos de negocio***

Dada la problemática de nuestro cliente, podemos identificar 4 objetivos:

- Predicción de la permanencia de un cliente (modelo de machine learning).
- Identificación de los clientes con alto riesgo de abandono (segmentación).
- Análisis de clientes con alto riesgo de abandono y sus características.
- Interpretación de los modelos y el análisis.

### ***Criterios de éxito***

Para determinar el éxito y satisfacción del proyecto, se deben de cumplir con los siguientes puntos:

- Obtener un porcentaje mínimo del 80% de precisión en el modelo de predicción.
- Hacer una segmentación de los clientes por grupos con características afines.
- Correcto análisis e interpretación de la segmentación de clientes.

## 1.2 Assess Situation

### *Inventory of Resources*

En este apartado se enlistan los recursos que están disponibles (o que fueron brindados por la institución educativa) para el desarrollo del proyecto.

- **Expertos:**
  - Dr. Benjamín Valdés Aguirre (área: Inteligencia Artificial).
  - Dr. Ismael Solís Moreno (áreas: BigData y Cloud Computing).
  - Dr. Carlos Alberto Dorantes Dosamantes (área: Estadística).
  - Dr. Juan Manuel Ahuactzin Larios (área: Inteligencia Artificial).
- **Datos:**
  - Archivo “telecom\_churn\_me.csv” extraído del sitio web: <https://www.kaggle.com/datasets/mark18vi/telecom-churn-data>
- **Recursos de cómputo:**
  - **Procesador:** Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz, 1498 Mhz, 4 Core(s), 8 Logical Processor(s).  
**Memoria RAM:** 12 GB  
**Almacenamiento:** 1 TB  
**Tarjeta Gráfica:** Intel(R) Iris(R) Plus Graphics  
**Sistema operativo:** Windows 10
  - **Procesador:** Arm M1  
**Memoria RAM:** 8GB  
**Almacenamiento:** 500 GB  
**Tarjeta Gráfica:**  
**Sistema operativo:** iOS
  - **Procesador:** Intel core i5 10th generation.  
**Memoria RAM:** 16 GB  
**Almacenamiento:** 2TB  
**Tarjeta Gráfica:** NVIDIA GTX 1650  
**Sistema operativo:** Windows 10
- **Software:**
  - Data Engineering (PySpark, Pandas, Numpy)
  - Machine learning libraries (TensorFlow, Keras, MLlib)

## ***Requirements, Assumptions and Constraints***

Este documento se puede encontrar en el repositorio de Github del proyecto como **"Requerimientos One Page.pdf"**.

## ***Risks and Contingencies***

Es necesario realizar la identificación de los riesgos, para poder proponer una solución en caso de que ocurran.

Los riesgos se evaluarán siguiendo una métrica del 1 al 4 en cuanto a probabilidad y gravedad. Una probabilidad 1 significa que es poco probable que ocurra, mientras que una probabilidad de 4 significa que las posibilidades de que ocurran son altas, de igual manera, tener un 1 como gravedad significa que el riesgo es bajo y no tendrá un gran impacto en la entrega del proyecto, sin embargo una gravedad de 4 significa que si el riesgo ocurre el proyecto ya no será viable como se había planteado. A continuación se presentará una tabla del análisis de riesgos que pueden presentarse durante este proyecto.

El nivel de riesgo se puede calcular haciendo uso de una matriz de evaluación de riesgo, en el cual se multiplica la probabilidad de que el evento suceda por la gravedad del mismo y se le asigna un nivel de prioridad. Los riesgos con una prioridad de 1-3 se aceptan, los riesgos con prioridad entre 4-8 se mitigan con un plan de acción y los riesgos con un nivel mayor de 9 se evitan totalmente.

Riesgo	Probabilidad	Gravedad	Nivel de prioridad
Los requerimientos no cubren las necesidades del socio formador	1	4	4
Un desarrollador no puede trabajar debido a una fuerza mayor	3	2	6
El modelo elegido no cumple con el porcentaje de	2	4	8

accuracy estipulado			
Los perfiles de los clientes no son fácilmente distinguibles.	2	4	8
La interfaz web no es lo que los clientes esperan	2	4	8
Una librería o tecnología no cumple con las características necesarias y debe ser cambiada	2	2	4
Se asigna un nuevo dataset cuando ya se llevaba un avance en el ETL	4	3	12
Las variables del modelo están poco correlacionadas con el target	1	4	4
Más de un miembro del equipo se encuentra ausente al mismo tiempo	2	4	8

### ***Terminology***

**Target** - Resultado obtenido que se encuentra en el set de datos o resultado esperado al momento de realizar un modelo de predicción. En este caso el target se considera como la permanencia del cliente.

**Inteligencia Artificial** - Combinación de algoritmos planteados con el propósito de crear programas que puedan realizar aprendizaje similar al ser humano.

**Ciencia de Datos** - Metodología utilizada para unificar estadísticas, análisis de datos, aprendizaje automático, y sus métodos relacionados, a efectos de comprender y analizar fenómenos que ocurren en alguna situación determinada.

**Machine learning** - Es la subdivisión de la inteligencia artificial que se centra en desarrollar sistemas que aprenden, o mejoran el rendimiento, con base en los datos que consumen y su procesamiento.

**Procesador** - Es un componente del hardware que se puede encontrar dentro de un ordenador, teléfonos inteligentes, y otros dispositivos programables. Su función es interpretar las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas, y externas.

**RAM** - La memoria de acceso aleatorio es utilizada en un sistema de cómputo para cargar los programas o archivos que se están siendo utilizados.

**Almacenamiento** - Un dispositivo de almacenamiento de datos es un conjunto de componentes electrónicos habilitados para leer o grabar datos en el soporte de almacenamiento de datos de forma temporal o permanente.

**Tarjeta Gráfica** - Es una tarjeta de expansión de la tarjeta madre o motherboard del computador que se encarga de procesar los datos provenientes del procesador y transformarlos en información comprensible y representable en el dispositivo de salida. Mayormente utilizada para representar gráficos en una pantalla, pero puede ser utilizada para otras tareas, como por ejemplo, resolver matrices de manera eficiente.

**Sistema operativo** - Es el programa que se encarga de administrar los recursos de la computadora.

**Data mining** - Es un campo de la estadística y las ciencias de la computación que se enfoca al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos.

## ***Costs and Benefits***

- 4 Laptops Asus:  $\$25,199 * 4 = \mathbf{\$100,796 \text{ MXN}}$ 
  - 15.6 Pulgadas
  - Full HD
  - Intel Core i5
  - NVIDIA GeForce RTX 3050
  - 8 GB RAM
  - 512 GB SSD
- Sueldo promedio de ingeniero de ML al mes en México: \$43, 218 MXN (Glassdoor: [https://www.glassdoor.com.mx/Sueldos/machine-learning-engineer-sueldo-SRCH\\_KO0.25.htm](https://www.glassdoor.com.mx/Sueldos/machine-learning-engineer-sueldo-SRCH_KO0.25.htm))
- $\$43,218 \text{ MXN} * 4 \text{ ingenieros} = \mathbf{\$172,872 \text{ MXN}}$
- Costo de procesamiento en la nube (Google Cloud):  $\$252.46 \text{ dólares} * \text{mes} = \mathbf{\$5,034 \text{ MXN}}$



Estimate

Compute Engine

1 x

Region: Iowa

304,167 total hours per month

Provisioning model: Regular

Instance type: n1-standard-8 USD 115.58

Operating System / Software: Free

GPU dies: 1 NVIDIA TESLA K80 USD 136.88

Estimated Component Cost: USD 252.46 per 1 month

Total Estimated Cost: USD 252.46 per 1 month

Estimate Currency

USD - US Dollar

EMAIL

COPY SAVED URL

DOWNLOAD\*

## 1.3 Determine Data Mining Goals

### *Data Mining Goals*

- Implementar un modelo de árbol de decisión para realizar predicciones sobre la permanencia de un cliente con base en su historial desde 1998 hasta 2018.
- Realizar una segmentación de clientes para identificar a los clientes con alto riesgo de abandono.
- Realizar un análisis de los perfiles clasificados como clientes con alto riesgo de abandono, tomando en cuenta las variables que determinan esta condición.
- Plasmar en un documento la interpretación del modelo y los resultados obtenidos.

### *Data Mining Success Criteria*

- El accuracy obtenido del decision tree debe ser mayor al 85% para ser considerado un modelo confiable.
- Realizar una matriz de confusión para evaluar el accuracy del modelo de clasificación.
- Los clientes considerados de alto riesgo comparten variables similares.

## 1.4 Produce Project Plan

### *Project plan*

El plan del proyecto se encuentra estructurado en Project Libre y se puede encontrar en el repositorio de Github del proyecto como **ProjectPlan.pod**, en él, se definen las etapas en las que se realizará el proyecto, que en este caso son semanales, ya que se trabaja con una metodología Ágil, así mismo se define la duración de cada una de las actividades a realizar, la persona o personas responsables de las actividades y las dependencias que se presentan a lo largo del proyecto.

### *Initial Assessment of Tools, and Techniques*

Este documento se puede encontrar en el repositorio de Github del proyecto como **“Mapecto de recursos y herramientas disponibles.pdf”**.

## 2 Data Understanding

### 2.1 Collect Initial Data

#### *Initial Data Collection*

El telecom-churn-me dataset contiene diversa información que puede ayudar a crear predicciones, es importante mencionar que no fue necesario realizar una recolección de datos ya que se trabajó con un set ya existente:

**Year Joined** - Esta columna contiene la información del año en el que se unieron los clientes. Esta columna puede ser utilizada para saber la lealtad de los mismos y cuánto tiempo llevan con la empresa.

**Party Nationality** - Brinda información demográfica de los usuarios, del país en el que se encuentran y de la mano con la columna de socio economic segment, brindan información importante para el análisis de estancia de los clientes.

**Bill Amount** - Hace referencia a la cantidad que los clientes pagan por los servicios que brinda la compañía telefónica. Cada determinado tiempo, los clientes reciben una cuenta que les informa la cantidad que deben pagar para poder continuar con el servicio.

**Complaints** - Esta columna hace referencia a la cantidad de quejas que tienen los clientes sobre su servicio. Dicha columna puede ser utilizada para analizar en qué clase de servicio se tienen mayores quejas.

**Status** - El status, muestra si la cuenta está activa o inactiva. Es una columna que cuenta con muchos valores iguales. Es por ello que de primera vista no parece ser una columna muy prometedora.

En total se tienen 1,140,606 filas de datos y 28 columnas que serán analizados más adelante y los datos no cuentan con valores nulos. Por el momento es suficiente para crear un programa. En este punto Naatik no tiene intenciones de comprar bases de datos externas, es por ello que sólomente se utilizarán los set de datos que se tienen hasta el momento. Este set de datos no fue modificado ni unido con otro dataset.

## 2.2 Collect Initial Data

### *Data Description*

Existen diversas maneras de realizar las descripciones de los datos. Pero la mayoría de los datos se enfocan en la cantidad y la calidad de los datos. Esto incluye con cuántos datos se cuenta y la calidad de los datos.

**Cantidad de datos** - Grandes conjuntos de datos pueden producir modelos más precisos, pero también pueden alargar el tiempo de procesamiento. En un futuro, si se desea ampliar un dataset es importante tomar en cuenta y considerar trabajar con un subconjunto de datos.

**Value types** - Los datos que se encuentran en el dataset están en 2 formatos principales, numérico (int/float) y categórico (string).

**Coding schemes.** En este dataset se encuentra un esquema de datos, este puede ser observado en la columna de party\_gender\_cd en el cual se observan valores de M/F/U para representar el género de los clientes.

Este dataset cuenta con 28 columnas, de las cuales 4 son variables categóricas, 23 son variables numéricas y existe una variable de esquema de codificación. De igual manera, se cuenta con un total de 1,140,606 datos con ningún valor nulo.

A continuación se muestra el despliegue de los tipos de datos de las variables que contiene en el dataset:

```

Data columns (total 28 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Unnamed: 0                                1140604 non-null  int64
1   PTY_PROFILE_SUB_TYPE                      1140604 non-null  object
2   SOCIO_ECONOMIC_SEGMENT                   1140604 non-null  object
3   PARTY_NATIONALITY                        1140604 non-null  object
4   PARTY_GENDER_CD                          1140604 non-null  object
5   TARGET                                    1140604 non-null  int64
6   YEAR_JOINED                              1140604 non-null  int64
7   CURRENT_YEAR                             1140604 non-null  int64
8   BILL_AMOUNT                              1140604 non-null  float64
9   PAID_AMOUNT                              1140604 non-null  float64
10  PAYMENT_TRANSACTIONS                     1140604 non-null  int64
11  PARTY_REV                                1140604 non-null  float64
12  PREPAID_LINES                            1140604 non-null  int64
13  POSTPAID_LINES                           1140604 non-null  int64
14  OTHER_LINES                              1140604 non-null  int64
15  LINE_REV                                 1140604 non-null  float64
16  STATUS                                    1140604 non-null  object
17  MOUS_TO_LOCAL_MOBILES                    1140604 non-null  float64
18  MOUS_FROM_LOCAL_MOBILES                  1140604 non-null  float64
19  MOUS_TO_LOCAL_LANDLINES                  1140604 non-null  float64
20  MOUS_FROM_LOCAL_LANDLINES                 1140604 non-null  float64
21  MOUS_TO_INT_NUMBER                       1140604 non-null  float64
22  MOUS_FROM_INT_NUMBER                     1140604 non-null  float64
23  DATA_IN_BNDL                            1140604 non-null  float64
24  DATA_OUT_BNDL                           1140604 non-null  float64
25  DATA_USG_PAYG                           1140604 non-null  float64
26  COMPLAINTS                               1140604 non-null  int64
27  Years_stayed                             1140604 non-null  int64
dtypes: float64(13), int64(10), object(5)
memory usage: 243.7+ MB

```

Por el momento no se tiene planeado crear nuevas variables a partir de los datos obtenidos, el único cambio que se realizó al dataset fue crear un label encoding de los datos categóricos y de esquema de codificación. Esto significa asignar un valor numérico a cada uno de los valores que aparecen en estas columnas.

No existe una gran cantidad de valores únicos en las columnas, solamente se cuenta con dos columnas con una alta cantidad de valores únicos, la primera es “unnamed: 0” que puede considerarse un ID y la segunda el “status”, que en su mayoría cuenta con el valor de active. Estas columnas no serán tomadas en cuenta para el procesamiento de datos y será necesario sacarlas del modelo.

## 2.3 Explore Data

### *Data Exploration*

Realizar la exploración de los datos puede ayudar a formular hipótesis y dar forma a las tareas de transformación de datos que se realizarán en la siguiente etapa de transformación de datos. Es importante tomar en cuenta que el proceso de la exploración de datos puede y debe personalizarse cada vez que se trabaje con un

dataset, ya que cada uno tiene necesidades particulares y diferentes formas de ser tratado.

En la parte de exploración se revelan resúmenes interesantes sobre los datos que pueden ayudar a darse una idea de la estructura del dataset y de las variables que pueden ser utilizadas. A continuación se encuentra el resumen de la información obtenida en esta sección de exploración.

	Unnamed: 0	TARGET	YEAR_JOINED	CURRENT_YEAR	BILL_AMOUNT	PAID_AMOUNT	PAYMENT_TRANSACTIONS
count	1.140604e+06	1.140604e+06	1.140604e+06	1.140604e+06	1.140604e+06	1.140604e+06	1.140604e+06
mean	5.703077e+05	5.278344e-02	2.013381e+03	2.018947e+03	3.811804e+02	3.921391e+02	1.346936e+00
std	3.292671e+05	2.236010e-01	6.082378e+00	2.236010e-01	3.697039e+02	3.725608e+02	7.309284e-01
min	0.000000e+00	0.000000e+00	1.994000e+03	2.018000e+03	-2.810494e+03	0.000000e+00	0.000000e+00
25%	2.851548e+05	0.000000e+00	2.013000e+03	2.019000e+03	1.741378e+02	1.816667e+02	1.000000e+00
50%	5.703065e+05	0.000000e+00	2.016000e+03	2.019000e+03	2.907239e+02	3.007292e+02	1.000000e+00
75%	8.554612e+05	0.000000e+00	2.017000e+03	2.019000e+03	4.609771e+02	4.764233e+02	2.000000e+00
max	1.140614e+06	1.000000e+00	2.018000e+03	2.019000e+03	2.702629e+04	2.274363e+04	3.000000e+01

La exploración también es útil para buscar errores en los datos. Si bien la mayoría de las fuentes de datos se generan automáticamente, también hay datos que son ingresados a mano. Es por ello que la exploración ayuda a detectar las dimensiones de los datos, las variables e incluso errores tipográficos.

## 2.4 Verify data

### *Data Quality*

Rara vez se tiene un perfecto dataset ya que la mayoría de los datos contienen errores de codificación, valores faltantes o algún otro tipo de inconsistencia que algunas veces dificultan el análisis. Es por ello que antes de realizar el modelo es muy importante realizar una normalización de los datos. imputación en caso de ser requerido y excluir las variables que cuenten con valores nulos recurrentes.

Es importante analizar la calidad de los datos basándonos en 3 principales problemas comunes:

**Missing Data** - Hace referencia a datos faltantes, que por alguna razón no fueron registrados en el dataset. Afortunadamente el dataset de entrenamiento con el que se está trabajando no contiene valores nulos, ni valores faltantes.

**Data Errors** - Hace referencia a buscar errores tipográficos en la base de datos y estos son regularmente encontrados y manejados en la etapa de exploración de los datos. A simple vista no parecen haber errores en los datos, sin embargo se tomará en cuenta esta posibilidad al momento de realizar la preparación de los datos.

**Measurement Errors** - Es probable que existan discrepancias si se juntan datasets, ya que algunos datos pueden contener horas y otros en minutos, por ejemplo, este tipo de errores pueden ser encontrados y tratados en el proceso de exploración. Sin embargo, en este dataset no se encuentran este tipo de errores y la información proviene de un solo set de datos, no de múltiples.

## 3 Data Preparation

### 3.1 Selecting Data

#### *Dataset Description*

Ya que uno de los requerimientos de nuestro socio formador era la generalización del código para el procesamiento y análisis de datos, se utilizaron 2 conjuntos de datos referentes a la permanencia o no de un cliente, que se pueden encontrar en la plataforma Kaggle.

### 3.2 Cleaning Data

- **Evaluación de columnas:**

Las columnas que tuvieran un 65% o más de valores nulos fueron eliminadas para evitar afectar el procedimiento de entrenamiento.

De igual manera, se eliminaron las columnas que estuvieran completamente compuestas de valores únicos, esto porque se pretende generalizar el código y una columna sin valores repetidos **puede** ser una sin significado (ejemplo: ID, Nombre, etc.).

También se eliminaron las columnas que estuvieran completamente compuestas de un sólo valor, pues no aportarían al modelo.

- **Evaluación de filas:**

De igual manera como con las columnas, las filas que tuvieran un 65% o más de valores nulos fueron eliminadas para evitar afectar el procedimiento de entrenamiento.

- **Multicolinealidad:**

La multicolinealidad ocurre cuando dos o más variables independientes tienen una alta correlación y pueden causar una estimación poco confiable, por lo que estas variables deben detectarse y descartarse.

Para la detección de multicolinealidad se utilizó la técnica **Variance Inflation Factor (VIF)**. Este método realiza la regresión lineal de cada variable independiente contra todas las demás. El VIF se calcula:

$VIF = \frac{1}{1-R^2}$ , donde  $R^2$  es el coeficiente de determinación en regresión lineal. Un VIF más alto denota una fuerte colinealidad; generalmente, un VIF superior a 5 indica una alta multicolinealidad.

- **Manejo de valores extremos:**

Teniendo el z-score, es fácil identificar los valores atípicos con un umbral. El umbral es el valor que define el valor atípico como desviaciones estándar, por lo general se elige el número 3, ya que el 99,7 % de los puntos de datos se encuentran entre 3 desviaciones estándar utilizando el enfoque del teorema del límite central (distribución gaussiana).

Los valores extremos fueron guardados en un dataframe para después ser utilizados en el testeo del modelo.

### 3.3 Constructing New Data

- **Codificación de variables categóricas:**

Las columnas categóricas fueron identificadas apartir de sus valores únicos. Si se cuenta con un pequeño número de valores únicos, **puede** que se trate de una columna categórica.

Una vez identificadas las columnas categóricas utilizamos un label encoder para hacer la categorización numérica de las columnas. No se hizo la creación de columnas por categoría dado que este número sería demasiado alto y alentaría el modelo.

- **Imputación**

Se utilizó la función de impute (iterative imputer) de la librería de statsmodel para realizar la estimación de valores nulos en el conjunto de datos.

### 3.4 Formatting Data

- **Estandarización:**

El z-score (puntuación estándar) es un método popular para estandarizar datos. La z-score (puntuación estándar) es una métrica de puntuación numérica que expone qué tan lejos está el punto de datos de la media.

La fórmula para estandarizar los datos es:  $z_{score} = \frac{data\_point - mean}{std\_desviación}$

## 4 Modeling

### 4.1 Selecting Modeling Techniques

Tomando en cuenta que se trata de un problema de clasificación se buscaron modelos de Machine Learning utilizados para este tipo de problemática.

De acuerdo con Gong D. (Feb. 2022) existen distintos algoritmos que pueden hacer el proceso de clasificación, entre ellos se encuentran:

1. Logistic Regression
2. Decision Tree
3. Random Forest
4. Support Vector Machine (SVM)
5. K-Nearest Neighbors (KNN)
6. Multilayer Perceptron
7. Naive Bayes

De estos modelos se utilizarán los siguientes ya que el algoritmo de K-Nearest Neighbour no es considerado como un algoritmo óptimo, pues realiza las predicciones basado en la distancia de las variables y Support Vector Machine es un algoritmo que utiliza un hiperplano y basa su resultado igualmente en la distancia de variables:

1. Logistic Regression
2. Decision Tree
3. Random Forest
4. Multilayer Perceptron

Adicionalmente se agregó un algoritmo del arte de Deep Learning, llamado Convolutional Neural Network de una sola dimensión.

### 4.2 Building and Assessing the model



Para poder seleccionar el modelo de clasificación se realizaron los 5 modelos de Machine Learning seleccionados haciendo iteraciones y mejoras con el objetivo de obtener el mejor resultado:

- Regresión Logística (Modelo de Benchmark)
- Multilayer Perceptron
- Decision Tree
- Random Forest
- Convolutional Neural Network

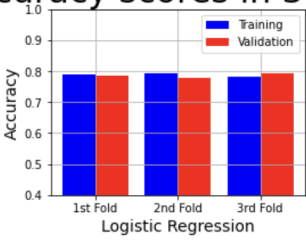
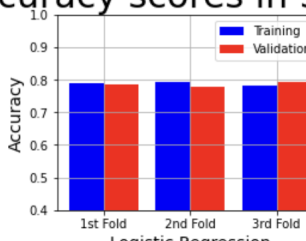
Es importante mencionar que como parte de la solución del reto y por requerimiento de los socios formadores, se necesita tener un modelo “generalizado” que permita la entrada de cualquier archivo churn. Es por ello que la elección del entrenamiento del modelo la tomaremos haciendo uso del dataset ‘WA\_Fn-UseC\_-Telco-Customer-Churn’ esperando que su performance sea similar con los nuevos inputs de los socios formadores.

Para la selección de este modelo se probarán distintas configuraciones, esperando obtener un buen nivel de precisión con la clasificación (si es churn o no es churn). Así mismo, como parte de la selección del modelo se considera el requerimiento de que el modelo debe ser fácil y rápido de procesar, ya que no se estará trabajando con un servidor sino con una computadora perteneciente a los socios formadores. Es importante resaltar que en el dataset probado se tiene un total de 7,043 datos, de los cuales el 73% son clientes sin churn (5,174) y el 27% son clientes con churn (1,869).

### Regresión logística (modelo de benchmark)

Como modelo de benchmarking hemos decidido utilizar una regresión logística, que es una técnica simple de aprendizaje automático para clasificación y se utilizará como punto de comparación para otros modelos a generar:

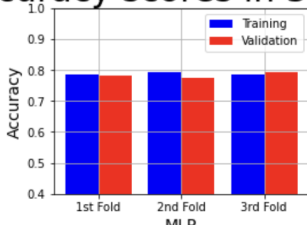
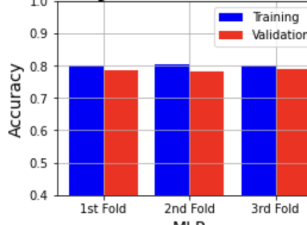
Modelo	Regresión logística	Regresión logística (mejorada)
Configuración	<b>penalty:</b> none <b>tolerance:</b> 1e-10 <b>regularization strength:</b> 0.5 <b>random state:</b> 100	<b>penalty:</b> l2 <b>tolerance:</b> 1e-4 <b>regularization strength:</b> 1.0 <b>random state:</b> 100
MSE Train	0.2123	0.2127

Train Accuracy	0.7877	0.7873																								
MSE Test	0.2112	0.2107																								
Test Accuracy	0.7888	0.7893																								
Folds Validation	<b>Accuracy scores in 3 Folds</b>  <table border="1"> <caption>Accuracy scores in 3 Folds (Logistic Regression)</caption> <thead> <tr> <th>Fold</th> <th>Training</th> <th>Validation</th> </tr> </thead> <tbody> <tr> <td>1st Fold</td> <td>0.78</td> <td>0.78</td> </tr> <tr> <td>2nd Fold</td> <td>0.78</td> <td>0.78</td> </tr> <tr> <td>3rd Fold</td> <td>0.78</td> <td>0.78</td> </tr> </tbody> </table>	Fold	Training	Validation	1st Fold	0.78	0.78	2nd Fold	0.78	0.78	3rd Fold	0.78	0.78	<b>Accuracy scores in 3 Folds</b>  <table border="1"> <caption>Accuracy scores in 3 Folds (Logistic Regression)</caption> <thead> <tr> <th>Fold</th> <th>Training</th> <th>Validation</th> </tr> </thead> <tbody> <tr> <td>1st Fold</td> <td>0.78</td> <td>0.78</td> </tr> <tr> <td>2nd Fold</td> <td>0.78</td> <td>0.78</td> </tr> <tr> <td>3rd Fold</td> <td>0.78</td> <td>0.78</td> </tr> </tbody> </table>	Fold	Training	Validation	1st Fold	0.78	0.78	2nd Fold	0.78	0.78	3rd Fold	0.78	0.78
Fold	Training	Validation																								
1st Fold	0.78	0.78																								
2nd Fold	0.78	0.78																								
3rd Fold	0.78	0.78																								
Fold	Training	Validation																								
1st Fold	0.78	0.78																								
2nd Fold	0.78	0.78																								
3rd Fold	0.78	0.78																								
Matriz de confusión	<table> <tr> <td></td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1419</td><td>314</td></tr> <tr> <td>1</td><td>132</td><td>247</td></tr> </table>		0	1	0	1419	314	1	132	247	<table> <tr> <td></td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1420</td><td>314</td></tr> <tr> <td>1</td><td>131</td><td>247</td></tr> </table>		0	1	0	1420	314	1	131	247						
	0	1																								
0	1419	314																								
1	132	247																								
	0	1																								
0	1420	314																								
1	131	247																								

Sobre los resultados, podemos observar, que las métricas tanto de error como de accuracy, resultan ser mejores en el train para el modelo de regresión logística inicial y en el test para el modelo de regresión logística mejorado. La diferencia realmente no es significativa ya que varían por milésimas, así mismo en el k-folds validation test ambos modelos tienen resultados similares, en cuanto a la matriz de confusión se puede observar que en ambos modelos se detecta la misma cantidad de positivos que si son positivos, sin embargo el primer modelo detecta un falso negativo más que el segundo modelo, lo que hace el segundo modelo uno mejor para detectar y retener clientes con churn. Por lo tanto, se ha decidido tomar como modelo de benchmark el modelo mejorado, por su performance en el set de prueba y sus resultados en la matriz de confusión.

## Multilayer Perceptron

Un MLP puede ser ocupado como modelo de clasificación cuando los inputs tienen asignada una clase o etiqueta. En este caso se considera el churn como etiqueta. Las configuraciones y los resultados se encuentran a continuación.

Modelo	Multilayer Perceptron	Multilayer Perceptron (mejorado)																		
Configuración	<b>Maximum number of iterations:</b> 100 <b>Hidden layer sizes:</b> (50,50) <b>activation:</b> Logistic <b>Solver:</b> Adam <b>Random state:</b> 1	<b>Maximum number of iterations:</b> 200 <b>Hidden layers:</b> (7, 5, 3) neuronas <b>activation:</b> relu <b>Solver:</b> lbfgs <b>Random state:</b> 1																		
MSE Train	0.2170	0.2024																		
Train Accuracy	0.7830	0.79760																		
MSE Test	0.2050	0.2027																		
Test Accuracy	0.7950	0.7973																		
Folds Validation	<b>Accuracy scores in 3 Folds</b> 	<b>Accuracy scores in 3 Folds</b> 																		
Matriz de confusión	<table> <tr> <td></td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1381</td><td>263</td></tr> <tr> <td>1</td><td>170</td><td>298</td></tr> </table>		0	1	0	1381	263	1	170	298	<table> <tr> <td></td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1412</td><td>289</td></tr> <tr> <td>1</td><td>139</td><td>272</td></tr> </table>		0	1	0	1412	289	1	139	272
	0	1																		
0	1381	263																		
1	170	298																		
	0	1																		
0	1412	289																		
1	139	272																		

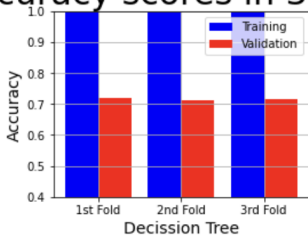
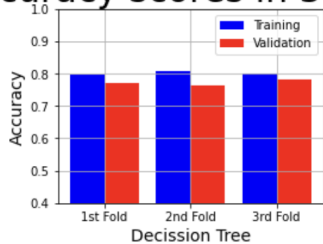
Para este modelo, los resultados del MLP mejorado se observan considerablemente mejor, ya que tanto las métricas de accuracy y error obtienen mejores resultados en test y train de este modelo. Así mismo se obtuvo una mejora de 1.31% en el Train Accuracy y de 1.01% en Test Accuracy en

comparación con el modelo de benchmark. Por otro lado, los resultados del k-folds validation lucen similares, pero se puede observar menos variación en el modelo mejorado.

Sobre la matriz de confusión, podemos observar que el modelo mejorado detecta 25 positivos que son positivos más, lo que equivale a una mejora del 10.12% en detección de churn, sin embargo en la detección de negativos, detectó 8 negativos que si son negativos menos, lo que no es problema, ya que estos entraron en los falsos positivos y no es una cantidad considerable, tomando en cuenta el total de los datos.

## Decision Tree

Los árboles de decisión son un modelo de predicción utilizado en la inteligencia artificial para realizar clasificaciones. La estructura natural de un árbol se recorre secuencialmente evaluando el estatuto lógico de cada nodo hasta llegar a la final de la predicción y evaluar un “si” o un “no”, en este caso de la variable churn.

Modelo	Decision Tree	Decision Tree (mejorado)
Configuración	<b>Criterion:</b> log_loss <b>Splitter:</b> best <b>min_samples_split:</b> 2 <b>Max depth:</b> None <b>Random state:</b> 0	<b>Criterion:</b> gini <b>Splitter:</b> random <b>min_samples_split:</b> 50 <b>Max depth:</b> None <b>Random state:</b> 0
MSE Train	0.0063	0.1892
Train Accuracy	0.9937	0.8108
MSE Test	0.2670	0.2202
Test Accuracy	0.7330	0.7799
Folds Validation	Accuracy scores in 3 Folds 	Accuracy scores in 3 Folds 

Matriz de confusión	0 1		0 1	
	0	1	0	1
0	1301	279	1430	362
1	285	247	121	199

En este caso nos encontramos con resultados parecidos al modelo de benchmark, en donde los resultados tanto del MSE como del Accuracy del Train son muy buenos, en realidad el resultado del set de prueba con 99% de accuracy es muy prometedor, sin embargo con el set de Test sólo obtiene un 73% lo que significa un overfitting en el modelo de entrenamiento. Sin embargo con el algoritmo mejorado, aunque sólo alcanza un 81% con el set de prueba, con el set de entrenamiento se alcanza un accuracy de casi 78% lo que lo hace un modelo más estable pero aún así no mejor que el modelo de benchmark, el cual obtiene un 78.73% en train accuracy y un 78.93% en test accuracy. Así mismo, si se analiza el error el primer modelo tiene un error muy bajo, lo que es ideal, pero con el set de prueba se dispara el valor más alto de error que se haya observado hasta ahora.

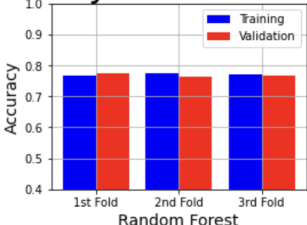
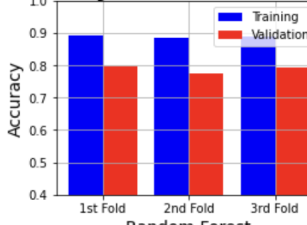
Aplicando k-folds validation observamos el mismo comportamiento, básicamente el primer modelo tiene un performance impresionante con el set de train y un performance muy pobre con el set de validación. Mientras que con el segundo modelo se observa ese equilibrio que se busca entre los resultados de train y validación. La matriz de confusión, por otro lado, nos muestra resultados muy decepcionantes, ya que detecta 48 positivos menos, lo que equivale a un 19.43% menos, pero se tiene una mejora de 10 usuarios en la detección de negativos que si son negativos, sin embargo lo que queremos es detectar clientes con churn, por lo cual es una “mejora” poco considerable para el modelo.

Analizando la matriz de confusión del primer modelo (el no mejorado) no se tiene más que un resultado igual en positivos que si son positivos en comparación con el benchmark y una cantidad mucho menor en detección de negativos que son negativos (119) lo que equivale a un 8.38% más de falsos positivos, poco ideal para los clientes.

## Random Forest

Los bosques aleatorios o los bosques de decisiones aleatorias son un método de aprendizaje comúnmente utilizado para la clasificación. Funciona construyendo un grupo de árboles de decisión al momento del entrenamiento. Para propósitos

de clasificación, la salida resultante de un bosque aleatorio es la clase seleccionada por la mayoría de los árboles.

Modelo	Random Forest	Random Forest (mejorado)																		
Configuración	<b>max_depth:</b> 2 <b>criterion:</b> gini <b>random_state:</b> 0 <b>max_features:</b> sqrt	<b>max_depth:</b> 10 <b>criterion:</b> entropy <b>random_state:</b> 0 <b>max_features:</b> log2																		
MSE Train	0.2294	0.1320																		
Train Accuracy	0.7706	0.8680																		
MSE Test	0.2287	0.2093																		
Test Accuracy	0.7713	0.7907																		
Folds Validation	<p>Accuracy scores in 3 Folds</p> 	<p>Accuracy scores in 3 Folds</p> 																		
Matriz de confusión	<table> <tr> <td></td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1507</td><td>439</td></tr> <tr> <td>1</td><td>44</td><td>122</td></tr> </table>		0	1	0	1507	439	1	44	122	<table> <tr> <td></td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1412</td><td>303</td></tr> <tr> <td>1</td><td>139</td><td>258</td></tr> </table>		0	1	0	1412	303	1	139	258
	0	1																		
0	1507	439																		
1	44	122																		
	0	1																		
0	1412	303																		
1	139	258																		

Este modelo obtiene mejores resultados con la configuración mejorada. En general, es decir tanto en test y train se obtienen mejores resultados de MSE y Accuracy. Comparado con el modelo de benchmark se tiene una mejora del 10.25% en los resultados del train accuracy y de un 0.18% en el test accuracy. En cuanto a los resultados del MSE se obtiene un 0.66% menos de error en el modelo del benchmark del test y un 37.94% menos de error en el entrenamiento

comparado con el benchmark. Sin embargo se observa un desequilibrio importante al momento de realizar el k-folds validation, ya que el training se observa con valores de accuracy elevados, mientras que la validación obtiene un accuracy muy por debajo del valor de entrenamiento.

Por otra parte, analizando la matriz de confusión, se observa un total de 11 positivos que si son positivos más que el modelo de benchmark, lo que equivale a un 4.45% más y un total de 8 negativos que si son negativos menos, lo que equivale a un 0.66% no detectado que entran a la categoría de falsos positivos.

## Convolutional Neural Network (del arte de Deep Learning)

Una red neuronal típica tendrá una capa de entrada, capas ocultas y una capa de salida. Las CNN están inspiradas en la arquitectura del cerebro. Al igual que una neurona en el cerebro procesa y transmite información por todo el cuerpo, las neuronas artificiales o nodos en las CNN toman entradas, las procesan y envían el resultado como salida.

La red neuronal convolucional o CNN es un tipo de red neuronal artificial, que se usa ampliamente para la clasificación de imágenes/objetos.

Modelo	CNN	CNN (mejorado)
Configuración	<pre>-Conv1D(128, 3, activation='relu', input_shape=(x_train.shape[ 1],1)) -MaxPooling1D(2) -LeakyReLU() -Dropout(0.5) -Dense(32, activation='relu') -LeakyReLU() -Dropout(0.5) -Flatten() -Dense(64, activation='relu') -Dense(2)</pre>	<pre>-Conv1D(32, 2, activation='relu', input_shape=(x_train.shape[1 ],1)) -Dense(16, activation='relu') -MaxPooling1D(1) -Dropout(0.3) -Conv1D(16, 2, activation='relu') -Flatten() -Dense(16, activation='relu') -Dense(2)</pre>
MSE Train	0.2198	0.2028
Train Accuracy	0.7802	0.7972

MSE Test	0.2311	0.2216																		
Test Accuracy	0.7689	0.7784																		
Matriz de confusión	<table> <tr> <td></td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1413</td><td>157</td></tr> <tr> <td>1</td><td>325</td><td>217</td></tr> </table>		0	1	0	1413	157	1	325	217	<table> <tr> <td></td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1389</td><td>181</td></tr> <tr> <td>1</td><td>294</td><td>248</td></tr> </table>		0	1	0	1389	181	1	294	248
	0	1																		
0	1413	157																		
1	325	217																		
	0	1																		
0	1389	181																		
1	294	248																		

En primera instancia se optó por una red robusta para el entrenamiento del modelo, esto resultó en una precisión del 76.89% para el test y 78.02% para el entrenamiento. Esto nos indicaba que había un claro overfit, por lo que optamos por hacerle una modificación y disminuir el número de neuronas que se tenían en las capas oculta, como resultado obtuvimos un 79.72% en el entrenamiento y 77.84% en el test de precisión; a pesar de que esto también indica un overfit, podemos evaluar como mejor modelo al modificado. También podemos observar en las matrices de confusión que se tiene una mejor predicción para las instancias sin churn por más del 1% y que se tiene la misma precisión para las instancias con churn para el modelo mejorado.

## Selección del modelo

Ya que es necesario hacer la selección del modelo lo más generalizada posible y comparando los resultados de los 4 modelos con el modelo de benchmark, se ha llegado a la conclusión de utilizar el modelo Multilayer Perceptron (MLP). Ya que es el modelo que detecta la mayor cantidad de churn positivo que si es positivo, lo cual es lo ideal para la resolución de este reto que pretende prolongar la estadía de los clientes haciendo uso de un servicio. De igual manera, cuenta con un accuracy de predicción (con el set de datos de prueba) de un 80% lo cual es bastante bueno, considerando que el dataset cuenta con un 73% de no churn, es el 7% más.

Es importante mencionar que la selección del modelo está pensado en dos factores fundamentales, el primero es que sea lo más generalizado posible, esta es la razón por la cual no se está considerando una modificación de threshold con curvas ROC. Y el segundo es que sea lo mejor posible detectando usuarios con Churn.



Finalmente es importante mencionar que estas configuraciones dan resultados positivos con el dataset mencionado anteriormente, sin embargo, al ingresar un nuevo archivo puede que se tengan resultados similares, lo cual sería lo ideal, pero al ser tan generalizado, también es posible llegar a obtener un resultado poco satisfactorio.

### 4.3 Clustering

Uno de los requerimientos de los socio formadores es realizar una perfilación de clientes, es decir, cuál es la razón por la cual se dejan de contratar los servicios, se dejan de realizar renovaciones de membresías o se cancela una cuenta (hace churn).

Para poder obtener el método más eficaz se realizó una clusterización con el algoritmo de k-means, brich y de aglomeración. Sin embargo, no se pudo obtener un resultado con el algoritmo de aglomeración. La computadora utilizada tardó un aproximado de 600 minutos intentando clusterizar los datos, por el tiempo de espera, este método fue totalmente descartado.

```
agg_sil.bar(k, agg_silhouette_scores)
agg_sil.set_title('Agglomerative: Number of Clusters vs. Silhouette Score', fontsize = 10)
agg_sil.set_xlabel('Number of Clusters', fontsize = 20)
agg_sil.set_ylabel('Silhouette Score', fontsize = 20)

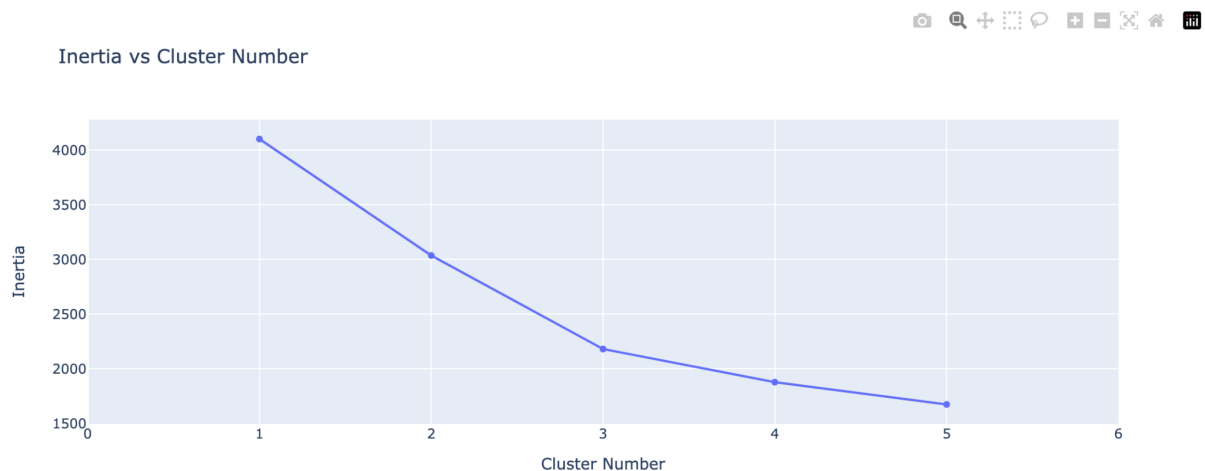
figure.tight_layout()
602m 29.3s
```

La clusterización con k-means tiene menos variación haciendo la comparación con silhouette, de igual manera, se considera el algoritmo mayormente utilizado para hacer clusterización ya que es un algoritmo de clasificación no supervisada que agrupa objetos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o cluster.

Lo necesario para realizar la clusterización es el set de datos, que se obtiene del ETL de datos predecidos por el modelo y el nombre de la columna que contiene el target.

El primer paso para realizar la clusterización es decidir el número de clusters que se desea crear, para ello se puede utilizar la técnica “elbow” la cual realiza un

cálculo de inercia dependiendo del número de clusters. Una vez obtenidos los valores de la inercia tras aplicar el K-means de 1 a N Clusters, en donde N es el número máximo de clusters (el numero total de variables) éste se representa en una gráfica lineal la inercia respecto del número de Clusters. La gráfica muestra un cambio brusco en la evolución de la inercia, teniendo la línea representada una forma similar a la de un brazo y su codo. El punto en el que se observa ese cambio brusco en la inercia nos dirá el número óptimo de Clusters a seleccionar para ese dataset. A continuación se muestra la gráfica resultante de aplicar este método en el archivo 'WA\_Fn-UseC\_-Telco-Customer-Churn' para detectar de manera visual el elbow point, equivalente al número de clusters:



El resultado visual es bastante claro, en este caso deberían considerarse 3 clusters, sin embargo hacerlo de manera visual puede ocasionar falsos resultados ya que algunas veces no es tan claro el punto elbow y se necesitaría de alguien que analice estos resultados, lo cual sería poco óptimo. Tomando en cuenta que necesitamos automatización y generalización para que el clustering funcione con cualquier conjunto de datos, hemos utilizado una función de la librería de Python 'kneed' llamada KneeLocator, la cual ayuda a automatizar este proceso. En el código, esta librería se usa en la función elbow y el resultado es el número de clusters recomendado para realizar la clusterización. Lo que se muestra a continuación es el resultado de la aplicación de esta función en el conjunto de datos, la cual coincide con los resultados de la gráfica.

```
clusters_number = elbow(x)
clusters_number
```

✓ 0.4s

Una vez definido el modelo de clusterización a utilizar y el número de clusters, se implementa el método, en este caso de K-means, haciendo uso de sklearn cluster se importa la función de Kmeans y se hace un fit con el dataframe. El resultado de esta operación agrega una columna al dataframe llamada 'label' la cual contiene el número de cluster al que pertenece.

	tenure	StreamingTV	PaymentMethod	MonthlyCharges	TotalCharges	label
0	0.595650	0.238870	0.595668	0.670930	-1.302066	0
1	1.328564	0.238870	0.595668	0.589507	0.996394	0
2	0.147758	-1.071381	1.466061	-0.801318	-0.315353	0
3	1.450717	-1.071381	0.595668	0.366842	-1.492021	0
4	1.613587	1.549122	0.595668	-1.299822	-0.282111	2
...	...	...	...	...	...	...
7038	-0.748027	0.238870	1.466061	0.521379	-0.327489	0
7039	-0.585157	-1.071381	1.466061	0.862023	0.994811	0
7040	-0.625874	-1.071381	-0.274724	-0.287859	0.892974	1
7041	0.473497	-1.071381	-1.145117	0.984987	1.226451	1
7042	1.409999	0.238870	1.466061	1.393760	0.394869	0

Para hacer una mejor visualización de los resultados se decidió realizar una gráfica polar, la cual realiza una agrupación por labels (clusters) y realiza un promedio del resultado de los valores de las variables. El resultado es una clusterización clara de cómo se comportan los perfiles de cada cluter en promedio con respecto a las variables involucradas.

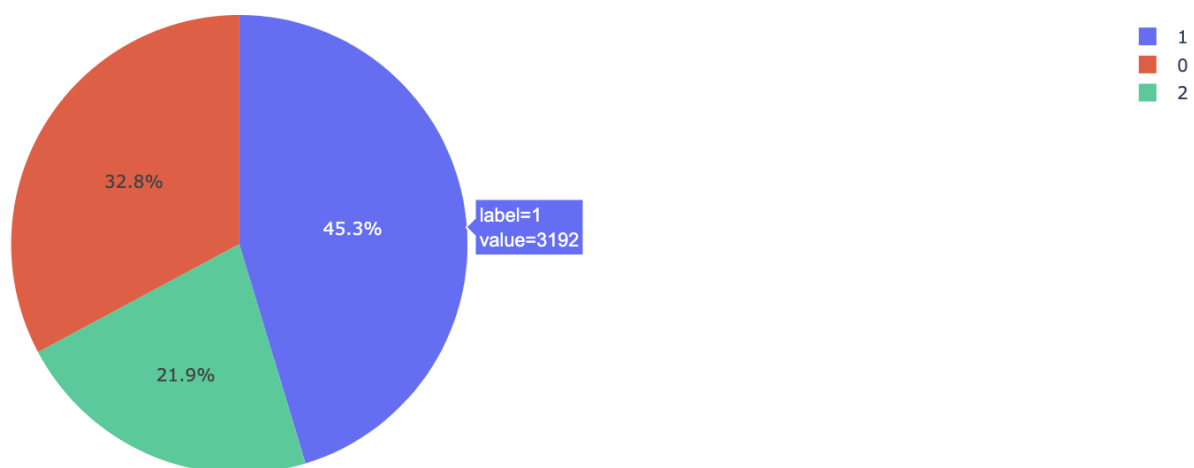


Debido a que es un modelo de aprendizaje no supervisado, a la fecha no hay una forma de asignar un perfilado más claro y se necesita de alguien que realice la

interpretación de los resultados y le asigne un nombre de perfil específico, por ejemplo:

- El perfil 0 tiene un comportamiento de tenure alto, así mismo tiene un total de cargos alto, que corresponde a una cantidad elevada de cargos mensuales y su tipo de pago es mayor, sin embargo tiene un Streaming moderado. Este perfil podría ser considerado como el cliente que más hace uso del servicio.
- El perfil 1 tiene todas las variables con el menor valor. Este cliente podría ser considerado como el cliente que menos hace uso del servicio.
- El perfil 2 tiene un total de cargos bajo, no tiene cargos mensuales, sus métodos de pago son moderados y hace el mayor uso de streaming con un valor moderado de Tenure. Este perfil puede ser considerado como el Streamer.

Así mismo se realiza una gráfica de pastel que ayuda a visualizar la distribución de datos en los de clusters .

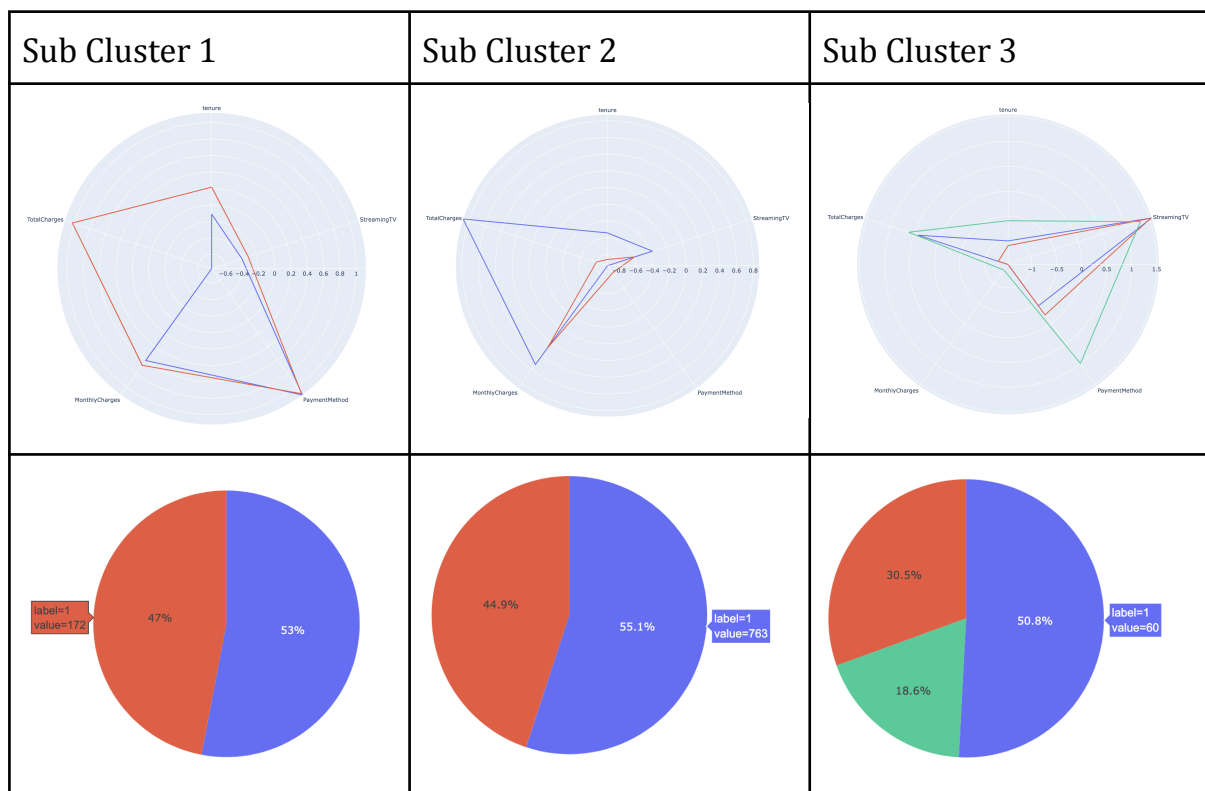


Estos resultados son los resultados de todos los datos, incluyendo clientes que tienen churn y no churn, sin embargo los socio formadores necesitan una perfilación más clara, es por ello que hemos decidido hacer subclusters, es decir, por cada cluster realizar una nueva clusterización solamente de los perfiles que presentan churn.

Para poder perfilar estos clientes que si se van a ir, lo primero que se hace es definir cuántos sub-clusters se tendrán. En este caso el número de sub-clusters que se tienen es el número de clusters del modelo inicial, es decir 3 en este caso.

Posteriormente se realiza un filtro del dataset por label, en total se tendrán 3 nuevos dataset, el primero será de datos con label 0, el segundo de datos con label 1 y el último de datos con label 2. Una vez realizado este filtro, cada nuevo dataset es filtrado por clientes que hacen churn, es decir, aquellos que tienen 1 como target.

Finalmente, haciendo uso de un iterador de 0 al número de sub clusters (3 en este caso) se realiza todo el proceso de clusterización nuevamente, iniciando con leer el dataset, calculando el elbow point de manera automática, realizando la clusterización, creando una gráfica polar de los resultados y finalizando con crear una gráfica de pastel de la distribución. El resultado de esta función aplicada al archivo 'WA\_Fn-UseC\_-Telco-Customer-Churn' es el siguiente:



Esto nos brinda un mejor entendimiento de los perfiles de los clientes que dejarán de utilizar el servicio, pagar suscripciones o cualquier otro caso que involucre Churn. Nuevamente estos datos deben ser interpretados, ya que se trata

de un algoritmo de aprendizaje no supervisado y la forma de hacerlo es parecida a la redactada anteriormente.

## 5 Evaluation

### 5.1 Evaluating the results

La evaluación de los resultados se fue haciendo a la par que se configuraban los modelos. Como métricas de evaluación se utilizó:

- Accuracy para Train
- Mean Squared Error para Train
- Accuracy para Test
- Mean Squared Error para Test
- Matriz de confusión
- K-Folds validation

#### Métrica de Accuracy

Se define como una función que retorna una fracción de las predicciones correctas. Es definida como:

$$accuracy(\hat{y}, y) = \frac{1}{n} \sum_{i=0}^n (\hat{y}_i = y_i)$$

#### Métrica de Mean Squared Error

El promedio del cuadrado de los errores (diferencia entre valores reales contra esperados)

$$MSE(\hat{y}, y) = \frac{1}{n} \sum_{i=0}^n (\hat{y}_i - y_i)^2$$

#### Matriz de confusión

Herramienta que permite la visualización del desempeño de un algoritmo de entrenamiento.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

*Verdadero Positivo*: predicción de positivo y es cierto.

*Verdadero Negativo*: predicción de negativo y es cierto.

*Falso positivo*: predicción de positivo y es falso.

*Falso Negativo*: predicción de negativo y es falso.

## **K-Folds validation**

El K-Cross Folds validation es un método estadístico utilizado para estimar la habilidad de los modelos de aprendizaje automático.

Se usa comúnmente en el aprendizaje automático aplicado para comparar y seleccionar un modelo para un problema de modelado predictivo dado porque es fácil de entender, fácil de implementar y da como resultado estimaciones de habilidades que generalmente tienen un sesgo menor que otros métodos.

## **5.1 Review process**

### **5.1 Determining the next steps**

- Método que ordene u omita las columnas por relevancia.
- Mejor visualización y guardado de los modelos ya existentes.
- Recuperar funcionalidad de dinero ahorrado que fue omitida.
- Agregar la opción visual para la implementación del algoritmo SMOTE (ya implementado en el repositorio principal).

## **6 Deployment**

### **6.1 Planning for deployment**

Para el modelo hicimos una aplicación web desarrollada en Next JS (React) la cuál interactúa directamente con los modelos y muestra gráficas de los resultados que nos arroja el análisis del documento csv ingresado, así como también genera un PDF en el cuál viene un resumen de todo lo que se muestra en dicha página web e información general del modelo de clasificación usado.

Esta página está conectada con un backend programado en Flask que realizará el procesamiento de datos y llamará a los modelos que están localizados en el mismo servidor.

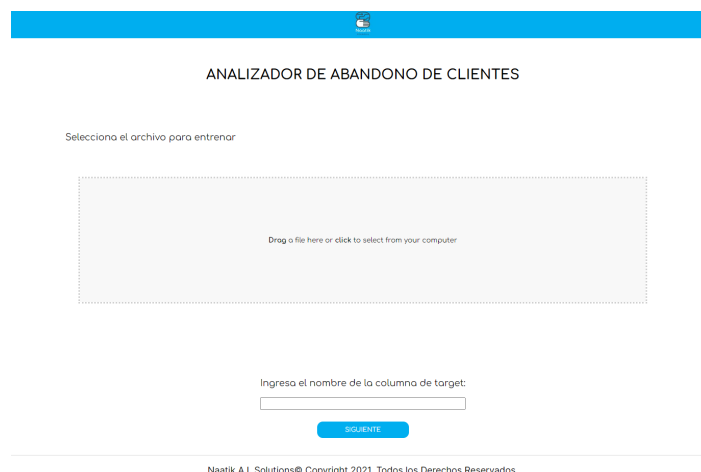


A continuación se muestran los mockups de las diferentes pantallas del sistema web:

- Pantalla inicial para seleccionar la opción de entrenar o realizar predicciones.



- Pantalla para subir archivo para entrenar modelo.





- Pantalla para subir archivo para predecir modelo.

ANALIZADOR DE ABANDONO DE CLIENTES

Selecciona el archivo a analizar

Upload your files here or **BROWSE**

File Name:

**SIGUIENTE**

Naozik AI. Solutions© Copyright 2021. Todos los Derechos Reservados.

- Pantalla para seleccionar los sliders del sistema.

CONFIGURACIÓN DEL MODELO

Porcentaje de Probabilidad de Abandono <sup>①</sup>

5% 50% 90%

**ATRÁS** **SIGUIENTE**

Naozik AI. Solutions© Copyright 2021. Todos los Derechos Reservados.

- Pantalla para guardar el modelo entrenado.

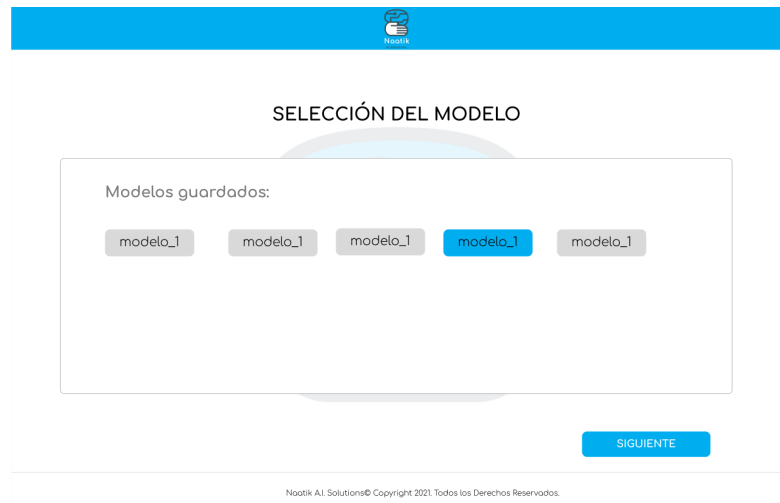
CONFIGURACIÓN DE MODELO

Nombre del modelo a guardar: modelo\_1

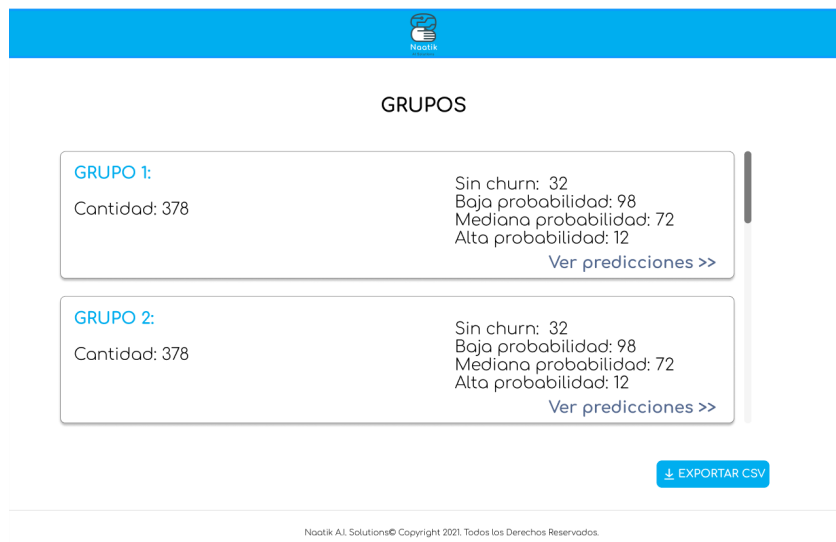
**SIGUIENTE**

Naozik AI. Solutions© Copyright 2021. Todos los Derechos Reservados.

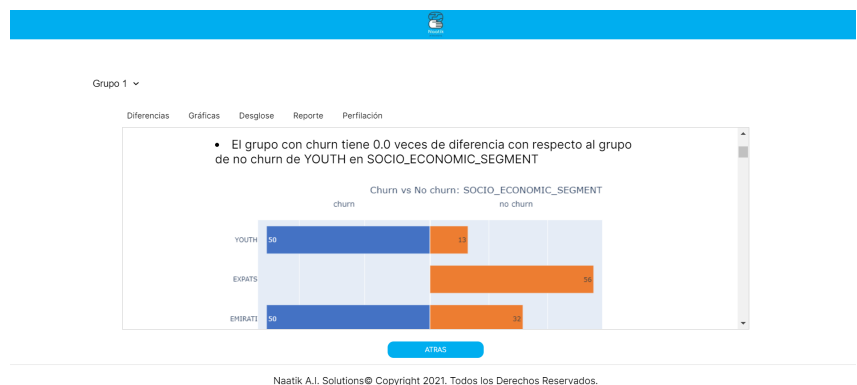
- Pantalla para seleccionar un modelo previamente entrenado.



- Pantalla donde se muestran los grupos determinados por el clustering.



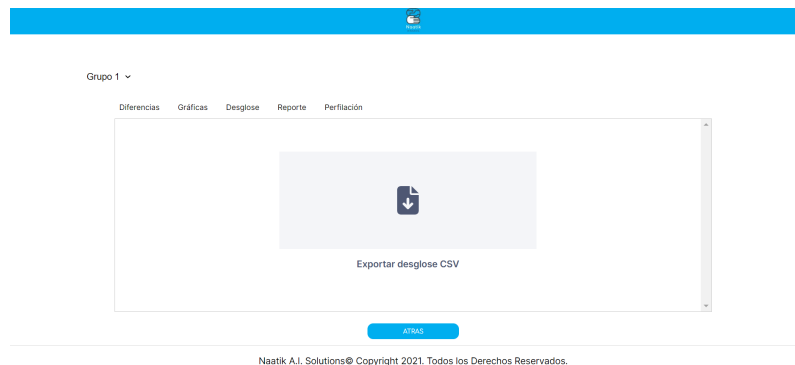
- Pantalla donde se muestra el análisis del dataset ingresado al sistema.



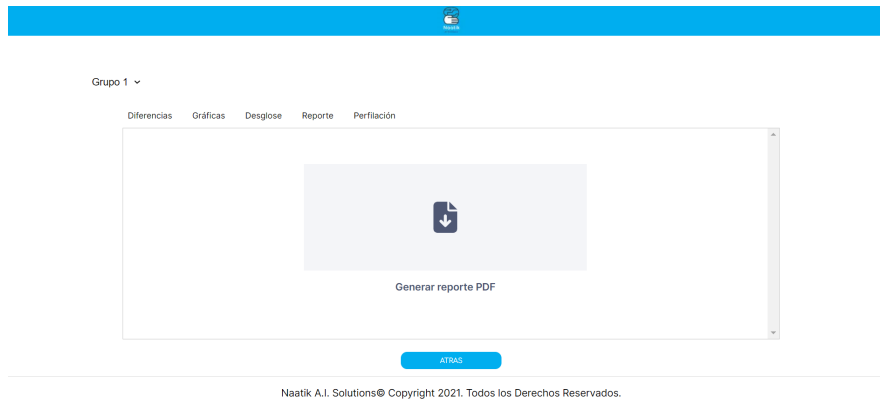
- Pestaña donde se muestran gráficas generales de churn y no churn del dataset ingresado.



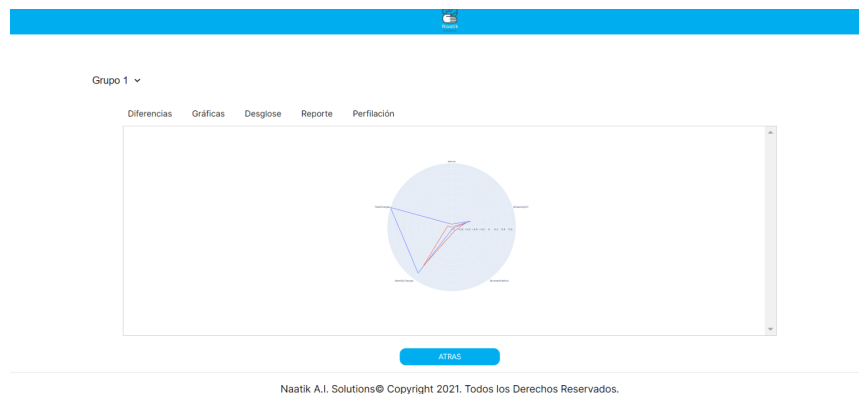
- Pestaña donde se puede descargar un desglose del dataset original ingresado con las probabilidades de churn y a qué grupo de slider pertenece.



- Pestaña donde se muestra un botón para descargar un reporte PDF con todo el análisis que se encuentra en el sitio web.



- Pestaña donde se mostrará la perfilación de los diferentes clientes que hacen churn.



## 6.2 Planning monitoring and maintenance

La eficiencia del modelo puede ser revisada por los usuarios por medio de la matriz de confusión que presenta el reporte generado por la aplicación web.

El monitoreo de que se está usando apropiadamente el modelo y que no está sufriendo de un declive de eficiencia debe ser realizado por la empresa Naatik.

## 6.3 Producing a final report

El sistema web genera un archivo PDF con todo el análisis del dataset ingresado al sistema para que pueda ser tratado como prefiera la empresa Naatik. En este archivo viene información general del archivo ingresado, así como un resumen del análisis realizado.

También se encuentran las gráficas de todos los clusters realizados con información de diferencias de variables significativas y gráficas generales de churn y no churn.

Considerando que los datos de no churn son de 94% (no churn es el dato que está más presente), se puede observar que hay una mejoría al utilizar este modelo comparado con solo apostar a que siempre será el mismo resultado.

## **6.4 Conducting a final project review**

### **6.4.1 Qué funcionó:**

- Se decidió realizar el desarrollo de la sección del frontend usando Next JS (React), lo cuál fue una buena decisión ya que al utilizar esta tecnología, los diferentes componentes desarrollados pudieron dividirse y trabajarse de una forma más organizada, lo cuál también si en un futuro se quiere agregar alguna funcionalidad, ésta se puede desarrollar sin comprometer la estructura del proyecto completo.
- Usar python en backend para tener una mejor interacción con los modelos y los procesos de limpieza

### **6.4.2. Qué no funcionó:**

- Se tuvo que eliminar la sección de “Ahorros” ya que cada dataset es diferente y no se tiene una variable “explícita” sobre ahorros de algún tipo, por lo que no se pueden calcular los ahorros de dicha variable.

## **Referencias**

- Naatik. (s/f). “Naatik AI Solutions”. Naatik. Recuperado el 10 de octubre del 2022 del sitio web: <https://www.naatik.ai/>
- ift. (s/F). “Instituto Federal de Telecomunicaciones”. ift. Recuperado el 10 de octubre del 2022 del sitio web: <https://www.ift.org.mx/usuarios-y-audiencias/telefonía-fija>
- IBM. (s/f). “Integrating Data”. Recuperado el 11 de noviembre del 2022 del sitio web: <https://www.ibm.com/docs/en/spss-modeler/saas?topic=preparation-integrating-data>

