UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMATICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

# DISEÑO DE UN MODELO DE CLIENTES APLICANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO PARA CORREOS DE CHILE

TESIS PARA OPTAR AL GRADO DE MAGISTER EN TECNOLOGÍAS DE LA INFORMACION

RUBEN CARLOS CRUZ OLIVARES

PROFESOR GUIA:

MAURICIO CERDA VILLABLANCA

MIEMBROS DE LA COMISION:

ANDRES ABELIUK KIMELMAN

IVAN SIPIRAN MENDOZA

MARCOS SEPULVEDA FERNANDEZ

SANTIAGO DE CHILE
2023

# Resumen Ejecutivo

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE
MAGISTER EN TECNOLOGIAS DE LA INFORMACION
POR: RUBEN CARLOS CRUZ OLIVARES
FECHA: 2023
PROFESOR GUIA: MAURICIO CERDA VILLABLANCA

## DISEÑO DE UN MODELO DE CLIENTES APLICANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO PARA CORREOS DE CHILE

La empresa Correos de Chile es una empresa estatal y autónoma, con presencia en todo el país, dedicada al servicio de correspondencia y al mercado de paquetería nacional e internacional. En los últimos años ha estado en constante cambio debido al comercio electrónico que se expande y exige dar un salto para fortalecer sus datos de Clientes, en una industria altamente competitiva. Los problemas detectados, se fundan en los servicios ofrecidos, los cuales no tienen una calidad adecuada en la data, debiéndose al poco manejo de la información, ya sea, el Rut, Email, Teléfono, Etc. lo cual repercute en la contactabilidad de ellos.

El objetivo del presente trabajo es la implementación de técnicas de distancia y similitud entre cadenas de textos, además de procesos de Machine Learning, que permitan la clasificación e identificación de Clientes, para poder entregar un mejor servicio en la cadena logística, desde la admisión hasta la entrega final de envíos. En el marco teórico se describe detalladamente la situación actual de la Empresa, destacándose la poca integración entre los sistemas con que opera. Además de una investigación bibliográfica de herramientas relacionadas, las métricas y sus validaciones, que nos ayudarán a analizar mejor los resultados.

En el desarrollo del trabajo, se realizó la definición de las fuentes de datos, en particular la identificación, limpieza e integración de los datos. Luego se procedió con la exploración de los datos para determinar los tipos de problemas más frecuentes en la identificación de Clientes. Se realizaron cálculos basado en métodos de distancia y similitud. Además, el uso de algoritmos de aprendizaje automático supervisado, en particular el modelo *Random Forest* para clasificación.

Con una muestra correspondiente a un año de datos de entrega en una ciudad, se elaboró una línea base de identificación de Clientes basada exclusivamente en la búsqueda exacta del Nombre, llamada “Algoritmo JOIN” que obtuvo un 60% de éxito en la identificación de Clientes. A partir de los resultados y análisis, se pudo observar que el algoritmo propuesto “Algoritmo Clasificador”, basado en *Random Forest*, obtuvo un 84% de éxito de identificación de Clientes, mientras que el máximo teórico de identificación de clientes “Algoritmo Oráculo” obtuvo un 92%. (8% corresponde a clientes nuevos).

Se pudo concluir que la aplicación del “Algoritmo Clasificador” cumplió con el objetivo planteado, mostrando cómo aumentar significativamente el éxito en la identificación de Clientes. A futuro, el rendimiento se podría incluso aumentar incorporando más variables al modelo, tales como, el Email y Teléfono.

*A mi familia por su confianza y apoyo incondicional*

# Agradecimientos

Deseo agradecer a todas las personas que me ayudaron en la confección y revisión de este documento. En especial dar las gracias a mi familia que fue un constante apoyo e inagotable fuente de valiosos comentarios. Asimismo, agradecer a mis amigos que siempre me han apoyado en diferentes etapas en la vida.

A mi profesor Mauricio por su infinita paciencia, ya que sin su ayuda este trabajo no hubiera sido posible, más que todo por su apoyo en conversaciones y por la posibilidad de trabajar con él durante el último tiempo.

A mis amigos y compañeros de los diplomados y del programa Magister, que siempre fueron un constante apoyo y conversaciones de experiencias vividas en el mundo de las ciencias de datos y las tecnologías de la información.

También extender al cuerpo académico del departamento de ciencias de la computación (dcc), por la formación de excelencia que recibí de ellos.

Además, agradezco a los funcionarios de facultad de ciencias físicas y matemáticas (fcfm), pues su labor silenciosa que posibilita un ambiente propicio para la enseñanza.

Finalmente quisiera agradecer a la Empresa Correos de Chile por darme la oportunidad de poder desarrollarme, pues este apoyo es de vital importancia para este trabajo, el cual se enmarca en la presente Tesis.

# Tabla de Contenido

# Índice de Tablas

# Índice de Figuras

# 1. Introducción

## 1.1. Contexto

La Empresa de Correos de Chile, también conocida como Correos de Chile es una de las instituciones más antiguas del país, con cerca de 270 años de funcionamiento. Es una empresa estatal y autónoma chilena, dedicada al servicio de correspondencia, giros postales y al mercado de envíos y encomiendas nacionales e internacionales, cumpliendo con las funciones de Servicio Postal Universal. Fue creada en virtud del DFL Nº 10, del 24 de diciembre de 1981 al disolverse el Servicio de Correos y Telégrafos. Actualmente, es la empresa líder del mercado nacional en el envío de correspondencia y paquetería. Anualmente, se mueven más de 20 millones de envíos internacionales y en el negocio nacional se distribuyen 234 millones de envíos nacionales entre correspondencia, paquetería y operaciones especiales (Correos de Chile, 2018).

Desde el año 2015, Correos de Chile se ha adaptado permanentemente a los cambios de la industria y a un mundo en constante evolución. Sin embargo, hoy estos cambios son más acelerados, lo que exige dar un salto "transformacional" para fortalecer la Empresa frente a sus clientes en el contexto de una industria altamente competitiva.

El comercio electrónico, que se expande de manera vertiginosa a nivel mundial, trae nuevas necesidades y exigencias de los clientes. Esto implica que los modelos de operación deben cambiar para satisfacer las nuevas necesidades. En otras palabras, se debe migrar de un modelo de correspondencia a uno de paquetería.

El valor agregado es que Correos de Chile es una organización que tiene un nombre, una historia, un presente y un futuro. Tiene una cobertura única con 500 puntos de atención y 92 centros de distribución postal a lo largo de Chile, con una logística de cerca de 2.500 carteros y alrededor de 400 móviles y una capacidad instalada de 24 plantas (Correos de Chile, 2019). La estructura de distribución (canales), según los servicios que ofrece, es la siguiente:

1. **Mundo Internacional**. Cuenta con una Planta Aeropuerto, la cual admite, clasifica y distribuye envíos a las siguientes Plantas:
    a. Planta CTP. Centro Tecnológico Postal.
    b. Planta CEP. Courier Express y Paquetería.

2. **Mundo Postal**. Cuenta con una Planta llamada "Centro Tecnológico Postal (CTP)", la cual retira, admite, clasifica y distribuye envíos postales y pequeños paquetes nacionales e internacionales a:
    a. Centros de Distribución Postal (CDP). Estos centros también denominados "Salas de Carteros", los cuales están ubicados en cada comuna del país, en la mayoría de las comunas existe uno, salvo las grandes, en donde hay más de uno. Ejemplo: Valparaíso, Concepción, Providencia, Santiago, etc. Estos CDP están divididos en cuarteles y cada Cuartel es atendido por un Cartero.

b. Sucursales. Son locales de atención al público, los cuales son de propiedad de Correos de Chile.
c. Agencias. Son locales de atención al público, los cuales son locales tercerizados.

3. **Mundo Paquetería**. Cuenta con una Planta llamada "Courier Express y Paquetería (CEP)", la cual retira, admite, clasifica y distribuye envíos de paquetería nacionales e internacionales a:
    a. Plantas Regionales. las cuales están ubicadas en cada región del país, las que a su vez distribuyen a Sectores de Distribución Paquetería (SDP). Cada SDP es atendido por un Móvil.
    b. Sucursales. Son locales de atención al público, los cuales son de propiedad de Correos de Chile.
    c. Agencias. Son locales de atención al público, los cuales son locales tercerizados.

Todos los Canales de Distribución, ya sea, Móviles, Carteros, Sucursales, Agencias, Centros de Entrega, Etc. cuentan con dispositivos móviles PDA (Personal Digital Assistant), para la trazabilidad de los envíos.

En cada admisión existe un proceso denominado "Normalización" el cual se encarga a partir de la dirección destino, poder identificar el código postal con el cual se puede determinar el CDP o SDP que se encargará de la distribución.

En el año 2018, la empresa trabajó en un Plan Estratégico (2018-2022), centrado en una mirada digital con foco en el cliente, por lo que la Gerencia de Tecnología, específicamente el Área de Inteligencia de Negocios, formó parte fundamental de los pilares de este plan. Las razones fundamentales fueron que el manejo de la información de la empresa debía ser administrada y centralizada por una sola unidad. Además, en el área ya está implementada una plataforma de Big Data en donde existe un Lago de Datos, en el cual se está almacenando información de los principales sistemas de la empresa, potenciando la trazabilidad de los envíos, además de disponer de una fuente única, robusta, consistente y centralizada de información.

En la actualidad, la mayoría de los sistemas y proyectos que se han desarrollado no se han preocupado por manejar datos centralizados, al contrario, cada uno maneja los suyos aumentando así la inconsistencia y duplicidad. Lo que impacta a las diferentes áreas de la empresa al requerir información para análisis.

## 1.2. Definición del Problema

Un estudio realizado en el año 2019 arrojó que cada día que pasa, existe una fuga importante de clientes insatisfechos, lo cual está teniendo consecuencias económicas casi irreversibles o de muy difícil recaptura para volver a generar la confianza que algún día existió.

También se detectó, que las diferentes áreas de la Empresa son reactivas a los problemas, o sea, no existen procesos proactivos que se anticipen a los problemas recurrentes.

La calidad en el "nivel de servicios" de la Empresa tiene muchas críticas, frente a una dura competencia, que les ofrecen a sus clientes mucha más información. En la Tabla 1, se muestra un cuadro comparativo de los principales competidores de Correos de Chile y los servicios ofrecidos a sus clientes, lo cual hace evidente el estancamiento que está ocurriendo en la Empresa.

| SERVICIOS a CLIENTES | CORREOSCHILE | chilexpress | starken | Blue express | DHL |
|---|---|---|---|---|---|
| Soluciones de Integración | | Si | Si | | Si |
| Logística Reversa | Sucursal | Sucursal | Sucursal / Domicilio | Domicilio | Domicilio |
| Notificaciones a SMS | Manual | Si | Si | | Previo registro |
| Notificaciones a Mail | | Si | Si | | Si |
| Canal de Entrega | Sucursal | Sucursal | Sucursal / Domicilio | | Sucursal / Domicilio |
| Entregas AM | Si | Si | | | |
| POD – Prueba de Entrega | | Si | | | Si |
| Imagen Firma de Entrega | | Si | Si | | Si |
| Aviso retiro en Sucursal / Agencia | | Si | Si | Si | |
| Avisos Carteros | Físico | Físico / Digital | Físico / Digital | Físico | Digital |

*Tabla 1: Comparativo de los Servicios a Clientes.*

Cada sistema computacional dentro de la empresa, tiene su propio repositorio de datos de clientes, productos, direcciones, etc. La única integración que existe entre ellos es para el manejo de la trazabilidad de cada envío.

Los grandes clientes empresas que operan con Correos de Chile, disponibilizan mes a mes envíos para su distribución, y en su gran mayoría son los mismos destinatarios que se repiten siempre, y los procesos de clasificación los vuelven a procesar como data nueva.

Solo los Clientes que contienen información en el campo Rut tienen la oportunidad de ser identificados, pero aun así no hay un repositorio centralizado que los almacene y menos que les asigne un **ID** único para permitir individualizarlos. Es casi imposible poder identificar y asociar al

Cliente que recibió un envío en el mes de mayo con el Cliente que recibió un envío en el mes de junio.

En definitiva, Correos de Chile "**No conoce a sus Clientes**", no sabe cómo opera y se desconoce el comportamiento de éste, frente a eventos del mercado.

## 1.3. Oportunidad Abordada

El problema a abordar en esta Tesis es poder asociar automáticamente una entrega a otras ya realizadas, como paso necesario para identificar a los Clientes. Al contar con la identificación de Clientes automática, se genera la contactabilidad en varios sistemas y por varios canales. Es decir, el problema abordado tiene un impacto potencial en:

- En el área Comercial. Conocer mejor al Cliente para poder entregarle facilidades en sus futuros despachos y nuevos productos.

- En el área Distribución. Repartos con entregas más efectivas al contar con más información para su ubicación.

- En el área Servicio Atención a Clientes. Mayor eficiencia en la gestión de búsqueda para resolver incidencias y/o pérdidas.

## 1.4. Objetivos

A continuación, se detallan los objetivos perseguidos en este trabajo de Tesis, todos en el contexto del Diseño de un Modelo de Clientes Aplicando Técnicas de Aprendizaje Automático.

### 1.4.1. Objetivo General

Desarrollar procesos o algoritmos que permitan la identificación de los Clientes que no cuentan con información única para poder diferenciarlos unos de otros, que considere todos los sistemas con que opera la Empresa. Logrando mejorar la contactabilidad de Clientes, desde la admisión hasta la entrega de envíos.

### 1.4.2. Objetivos Específicos

Los objetivos específicos que conforman este proyecto son los siguientes:

- Generar una Base de Clientes que incluya procesos de extracción y limpieza de la data relacionada con los remitentes y destinatarios a partir de los diferentes sistemas que operan en la Empresa.

- Procesar, clasificar e identificar los Clientes que no contienen información en el campo Rut en el modelo de datos creado, usando técnicas tradicionales de procesamiento de lenguaje natural.

- Diseñar procesos de identificación y clasificación de los Clientes que no contienen información en el campo Rut, aplicando técnicas de Machine Learning para el procesamiento de lenguaje natural.

## 1.5. Estructura de la Tesis

Para facilitar su lectura y comprensión de esta Tesis, se resume la estructura de ella según la siguiente lógica de capítulos:

- Capítulo 1: **Introducción**. Es el actual capítulo en el que se plantea el contexto de la Empresa, se define el problema, la oportunidad abordada y los objetivos de la Tesis.

- Capítulo 2: **Marco Teórico**. Se abordan los tópicos utilizados para llevar a cabo el trabajo de la Tesis, en donde se explica la situación actual de la Empresa, las técnicas de medición, la definición de Machine Learning y sus modelos, las métricas y su validación.

- Capítulo 3: **Desarrollo del Trabajo**. Se abordan los principales trabajos realizados y los pasos seguidos en cada etapa. Se dividirá el proyecto en tres etapas: definición de los datos a analizar, exploración de los datos, identificación de clientes basado en métodos de distancias y algoritmos propuestos para identificar clientes.

- Capítulo 4: **Resultados y Análisis**. Se abordan los procesos de evaluación, resultados obtenidos basados en métodos de distancias, resultados obtenidos basados en algoritmos propuestos y las limitaciones de la evaluación.

- Capítulo 5: **Propuesta de Implementación.** Se abordan los principales procesos que debiesen aplicarse para poder poner en producción el algoritmo de clasificación.

- Capítulo 6: **Conclusiones y Trabajos Futuros**. Se aprecian las principales conclusiones de este proyecto y todo lo que quedaría por realizar en los trabajos futuros, además de proponer posibles mejoras.

# 2. Marco Teórico

## 2.1. Descripción Situación Actual en la Empresa

Cada sistema que maneja la Empresa, tiene información de la trazabilidad de los envíos e información de los remitentes y destinatarios. Cuando existe alguna integración entre sistemas, solo se transmite la información del envío y parte de la información de los Clientes, ya sea, los siguientes datos:

- Número de Envío
- Fecha y Hora de Creación
- Código del Producto
- Nombre del Destinatario
- Dirección del Destinatario
- Comuna del Destinatario
- Código Postal

La información que no se transmite, queda almacenada dentro de cada sistema que la originó, estos datos son:

- Rut del Remitente
- Email del Remitente
- Teléfono del Remitente
- Dirección del Remitente
- Comuna del Remitente
- Rut del Destinatario
- Email del Destinatario
- Teléfono del Destinatario

Cada cierto tiempo por trabajos específicos, se realizan relaciones entre sistema en donde el objetivo es poder consolidar la información del Cliente en diferentes sistemas, pero esta labor se realiza en forma manual.

Para el proceso llamado “última milla”, que es cuando un móvil o cartero realiza la operación de distribución del envío, actualmente utilizan dispositivos digitales que cuentan con GPS en donde se registran los datos del Cliente (Nombre y Rut) y las coordenadas georreferenciales (Latitud y Longitud) de cada transacción realizada. Esta información tampoco es utilizada con posterioridad.

En los sistemas que manejan envíos internacionales, en su gran mayoría no vienen todos los datos, ya sea, el Rut, Teléfono, Email, Código Postal, etc.

Además, en los sistemas que manejan en las sucursales de venta al público, la mayoría de los campos son opcionales, ya sea, el Rut, Email, Teléfono, Apellidos, Etc. Tampoco tiene una

validación de consistencia, lo que hace imposible determinar que la información sea verdadera. Un ejemplo de estos datos se puede observar en la Tabla 2, la cual grafica de mejor forma lo expresado anteriormente.

| Envío | Fecha y Hora | RUT | Nombre | Dirección | Comuna | Teléfono | Email | Código Postal | Latitud | Longitud |
|---|---|---|---|---|---|---|---|---|---|---|
| 0668 | 19-02-2020 16:45:00 | 0 | Juan Alejandro Zaragoza B. | Alameda 123 | La Florida | 569 9123 45XX | juanato@yimail.com | 25340 | -33,654 | -71,3456 |
| 3577 | 15-03-2020 12:30:00 | 0 | Juan Zaragoza B. | Meiggs 321 | Estación Central | 9123 45XX | juan.zaragozinxx@llimail.cl | 91602 | -33,655 | -71,2534 |
| 1050 | 10-05-2020 21:30:00 | 0 | Juan Zaragoza Bahamondes | 321 Meiggs | Estación Central | 9123 45XX | juan.zaragozinxx@llimail.cl | 0 | 0 | 0 |
| 4507 | 16-06-2020 09:30:00 | 0 | Juan Zaragoza B. | Alameda 123 | La Florida | | | 25340 | -33,654 | -71,3456 |

*Tabla 2: Registro de Clientes de Correos de Chile.*

Dentro del Área de Inteligencia de Negocios, se está realizando un trabajo de consolidación de la data, trabajo que ha resultado en su gran mayoría con un esfuerzo muy costoso. Todo debido a que las herramientas utilizadas, no son las más idóneas para el manejo de millones de registros.

Actualmente el historial de la información que se maneja es de aproximadamente 18 a 24 meses, dependiendo de cada sistema. Recién el año 2018-2019 se implementó un Lago de Datos en un Clúster Hadoop, el cual está almacenando la información de las transacciones de los envíos, para propósitos de gestión de los Cubos Analíticos, y se pretende el año 2021-2022 incluir la información de los Clientes.

Al contar los Clientes con la información del Rut, su identificación es más exacta, pero no así, cuando no cuentan con este dato. Actualmente con las herramientas tradicionales se hace casi imposible poder identificarlo.

## 2.2. Técnicas de Medición de Distancia Entre Dos Puntos

A continuación, se explican maneras de medir distancias entre dos puntos, ya sea en un plano o en una esfera, distancias que se utilizarán en el desarrollo de este trabajo.

### 2.2.1. Distancia Euclidiana

La distancia Euclidiana es la más común de las medidas de distancia, se define como la distancia entre dos puntos en un espacio multidimensional y se calcula a partir de las coordenadas cartesianas de los puntos utilizando el teorema de Pitágoras (Grootendorst, 2021). En la Figura 1, se muestra la fórmula.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}.$$

*Figura 1: Fórmula de Distancia Euclidiana.*

Por lo general se necesita normalizar los datos antes de usar esta medida. Geométricamente esta distancia representa el camino más corto entre dos puntos. En la Figura 2, se aprecia un ejemplo de cómo se interpreta en el plano cartesiano.

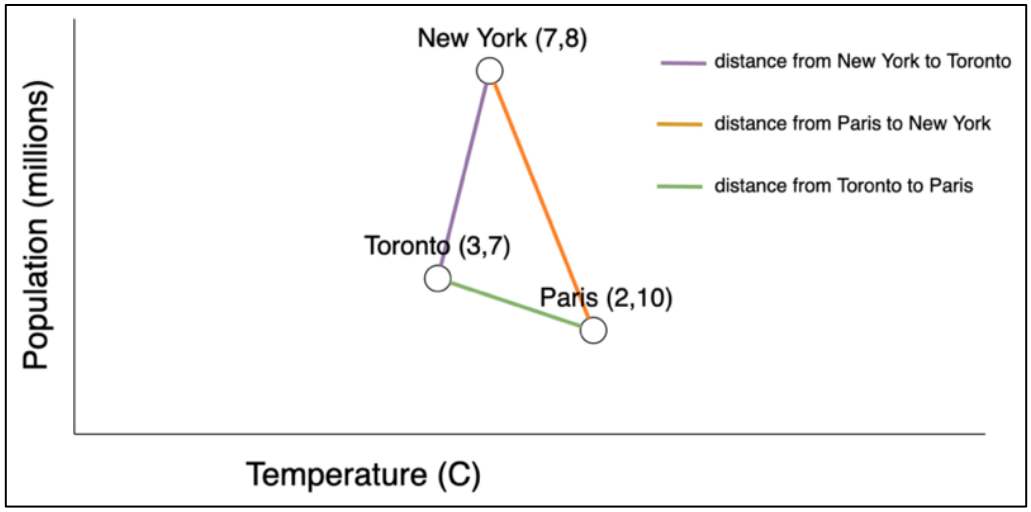


*Figura 2: Ejemplo de Uso de Distancia Euclidiana.*

## 2.2.2. Distancia Haversine

La distancia de Haversine es la distancia entre dos puntos sobre una esfera y está dada por sus longitudes y latitudes. Es muy similar a la distancia Euclidiana (Grootendorst, 2021). En la Figura 3, se muestra la fórmula.

$$d = 2r\ arcsin\left(\sqrt{\sin^2\left(\frac{\Phi_2 - \Phi_1}{2}\right) + \cos(\Phi_1)\cos(\Phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right).$$

*Figura 3: Formula de Distancia Haversine.*

Donde $\Phi_1$, $\Phi_2$ y $\lambda_1$, $\lambda_2$ se refieren a la latitud y la longitud, expresadas ambas en radianes, de los puntos 1 y 2 respectivamente y r corresponde al radio terrestre (Ecuatorial 6,378,1 Km, Polar 6.356,8 Km, Medio 6.371,0 Km).

La distancia Haversine la podemos expresar en Kilómetros o Metros. En la Figura 4, vemos como se representan los 2 puntos en una esfera.

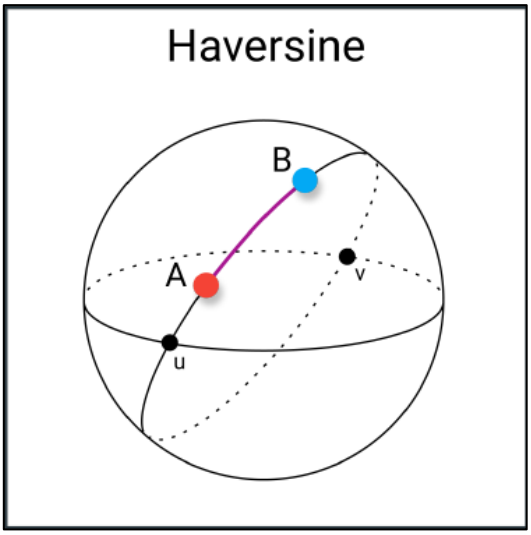


*Figura 4: Representación de 2 Puntos Geográficos.*

En la Figura 5, se ve cómo se aplica la distancia Haversine:

```
// Obtenga la distancia en metros usando la fórmula de Haversine
var distanceInMeters = getDistanceBetweenPoints(
    // LatA
    7.099473939079819,
    // LongA
    -73.10677064354888,
    // LatB
    4.710993389138328,
    // LongB
    -74.07209873199463
);

// Imprime: Distance in Meters:  286476.96153465303
console.log("Distancia en metros:", distanceInMeters);

// Imprime: Distance in Kilometers:  286.476961534653
console.log("Distancia en kilómetros:", (distanceInMeters * 0.001));
```

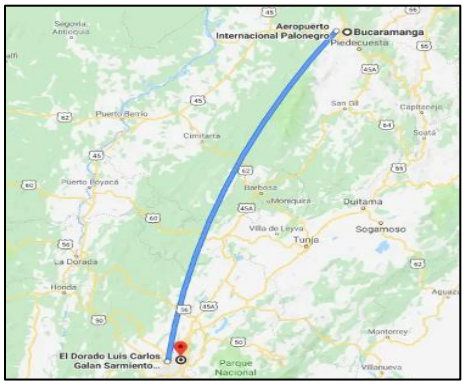


*Figura 5: Ejemplos de Uso de la Distancia Haversine.*

## 2.3. Técnicas de Medición de Distancia Entre Cadenas de Textos

En esta sección se explora el concepto de distancia y similitud entre cadenas de texto, describiéndose algunas de las más comunes usadas al comparar cadenas de texto en el contexto del Procesamiento de Lenguaje Natural (NLP), la Extracción de información y otras aplicaciones (D. Campos, 2019).

Es frecuente encontrar la necesidad de comparar diferentes palabras o frases entre sí. En muchos casos no solamente las coincidencias exactas, sino tener una medida de aproximación o similitud (D. Campos, 2019).

En el contexto de comparación de cadenas de caracteres se considera también el concepto distancia de edición, el cual corresponde al número mínimo de operaciones de edición necesarias para convertir una cadena de texto en la otra.

### 2.3.1. Distancia de Hamming

Esta distancia de uso común en teoría de la información criptografía y telecomunicaciones, es una de las métricas de distancia más simples (Invarato, 2016).

Al comparar dos cadenas de texto se hace un conteo del número de posiciones en las que los símbolos de las cadenas son diferentes. Una restricción de esta distancia es que las cadenas para ser comparadas deben ser del mismo tamaño. Otra forma de ver está distancia, es como una distancia de edición que cuenta el número mínimo de sustituciones de caracteres requeridas para convertir una cadena en la otra. En la Figura 6, se muestra la fórmula.

$$DH = \left( \sum_{i=1}^{k} |x_i - y_i| \right).$$

*Figura 6: Formula de Distancia Hamming.*

En la Figura 7, se muestra un ejemplo del uso de la distancia de Hamming, en donde se aprecia como destaca y cuenta las diferencias encontradas en la comparación con la variable A y B (Grootendorst, 2021).

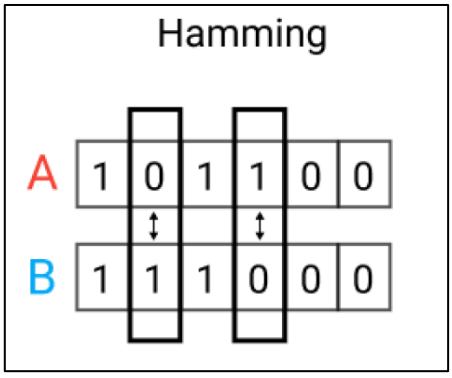


*Figura 7: Ejemplo de Uso de Distancia Hamming.*

## 2.3.2. Distancia de Levenshtein

Fue creada en 1965 por el científico ruso Vladimir Levenshtein. Es la distancia de edición más usada, la cual representa un número mínimo de operaciones (inserción, eliminación o sustitución de un carácter) requeridas para transformar una cadena de caracteres en otra (Levenshtein, 1966). Si proporciona un valor igual a 0, indica que ambas cadenas son idénticas. Mientras más lejos del 0 la cadena es menos similar a la otra. En la Figura 8, se muestra la fórmula.

$$\operatorname{lev}(a,b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ \operatorname{lev}(\operatorname{tail}(a), \operatorname{tail}(b)) & \text{if } a[0] = b[0] \\ 1 + \min \begin{cases} \operatorname{lev}(\operatorname{tail}(a), b) \\ \operatorname{lev}(a, \operatorname{tail}(b)) \\ \operatorname{lev}(\operatorname{tail}(a), \operatorname{tail}(b)) \end{cases} & \text{otherwise.} \end{cases}$$

*Figura 8: Formula de Distancia Levenshtein.*

Para su cálculo se utiliza una matriz en la que se van almacenando los resultados intermedios, la cadena origen va a la izquierda y la cadena destino en la parte superior. En la casilla inferior derecha queda almacenado el resultado final (D. Campos, 2019). En la Figura 9, se aprecian algunos ejemplos.

| | | E | L | E | P | H | A | N | T |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| R | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| E | 2 | 1 | 2 | 2 | 3 | 4 | 5 | 6 | 7 |
| L | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| E | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 |
| V | 5 | 4 | 3 | 2 | 2 | 3 | 4 | 5 | 6 |
| A | 6 | 5 | 4 | 3 | 3 | 3 | 3 | 4 | 5 |
| N | 7 | 6 | 5 | 4 | 4 | 4 | 4 | 3 | 4 |
| T | 8 | 7 | 6 | 5 | 5 | 5 | 5 | 4 | 3 |

| | | a | m | i | g | o |
|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 |
| a | 1 | 0 | 1 | 2 | 3 | 4 |
| m | 2 | 1 | 0 | 1 | 2 | 3 |
| i | 3 | 2 | 1 | 0 | 1 | 2 |
| g | 4 | 3 | 2 | 1 | 0 | 1 |
| o | 5 | 4 | 3 | 2 | 1 | 0 |

*Figura 9: Ejemplos de Uso de Distancia Levenshtein.*

### 2.3.3. Similitud de Coseno

Es una medida de similitud entre dos vectores en un espacio que posee un producto interior. Si proporciona un valor igual a 1, el ángulo comprendido es cero, es decir ambos vectores apuntan a un mismo lugar (Tan et al., 2006). Cualquier ángulo existente entre los vectores, el coseno arrojaría un valor inferior a 1 y si apuntasen en sentido contrario su valor sería -1. De esta forma, el valor de esta métrica se encuentra en el intervalo cerrado [-1, 1].

Esta distancia se emplea frecuentemente en la búsqueda y recuperación de información representando las palabras. Dados dos vectores de atributos, A y B, la similitud del coseno, se representa mediante un producto escalar y la magnitud como Esta es una representación gráfica de la similitud de coseno. Donde $A_i$ y $B_i$ son componentes del vector A y B respectivamente. En la Figura 10, se muestra la fórmula.

$$\boldsymbol{similarity} = \cos(\theta) = \frac{A*B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}.$$

*Figura 10: Formula de Similitud Coseno.*

La principal ventaja de la similitud de coseno es que incluso si los dos objetos de datos similares está muy separados por la distancia euclidiana debido al tamaño, aún podrían tener un ángulo más pequeño entre ellos. Cuanto menor sea el ángulo, mayor será la similitud. Cuando se traza en un espacio multidimensional, la similitud del coseno captura la orientación (el ángulo) de los objetos de datos y no la magnitud. En la Figura 11, se muestran unos ejemplos (Qian, 2020).

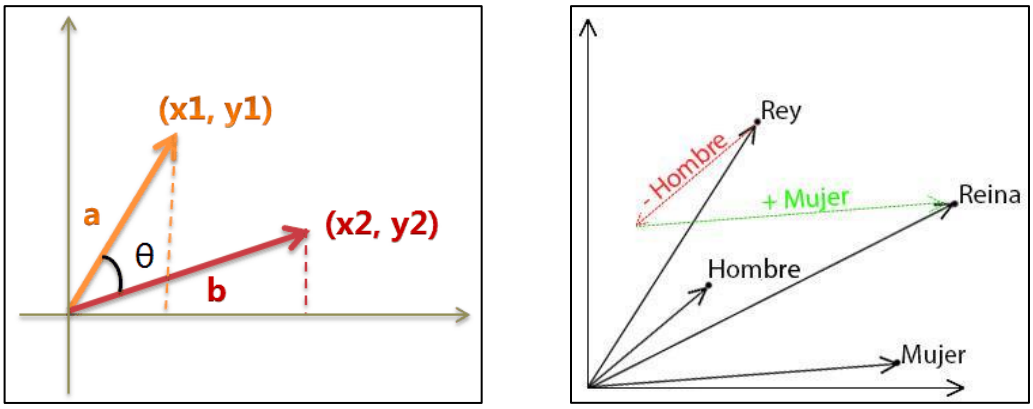


*Figura 11: Representación Gráfica de Similitud Coseno.*

### 2.3.4. Distancia de Jaro-Winkler

Otra distancia que se puede utilizar para medir la similitud entre dos cadenas de texto es la de Jaro-Winkler que es una modificación propuesta por Winkler en 1990 de la distancia que Jaro había propuesto un año antes. Si proporciona un valor igual a 1, indica que ambas cadenas son idénticas. Cualquier número bajo el 1 es menos similar entre las dos secuencias (Rodríguez, 2020).

La similitud de Jaro, la única operación de edición que utiliza es la transposición de caracteres. Por lo que solamente es necesario contar el número de caracteres iguales entre dos cadenas y el número de transposiciones que son necesaria para llegar de una cadena a otra (Elmagarmid et al., 2006).

Se define la distancia Jaro entre las cadenas de texto *a* y *b*. En donde |*a*| es la longitud de la cadena *a*, |*b*| es la longitud de la cadena *b*, *m* es el número de caracteres coincidentes de ambas cadenas y *t* es la mitad de número de transposiciones necesarios para convertir una cadena en otra. En la Figura 12, se muestra la fórmula.

$$j_{a,b} = \begin{cases} 0 & si\ m = 0, \\ \frac{1}{2}\left(\frac{m}{|a|} + \frac{m}{|b|} + \frac{m-t}{m}\right) & si\ m \neq 0 \end{cases}$$

*Figura 12: Fórmula de Distancia Jaro.*

Se introduce una escala que otorga calificaciones más favorables a las cadenas que coinciden desde el principio. Siendo la longitud de prefijo establecida un hiperparámetro que es necesario fijar. Lo cual se modifica mediante la expresión. En donde *l* es la longitud del prefijo común al comienzo de las cadenas y *p* es un factor de escala constante con el que se ajusta la puntuación hacia arriba por tener prefijos comunes. En la Figura 13, se muestra la fórmula.

$$\boldsymbol{jk_{a,b}} = j_{a,b} + lp\left(1 - j_{a,b}\right).$$

*Figura 13: Fórmula de Distancia Jaro-Winkler.*

En la Figura 14, se muestra cómo funciona la distancia Jaro-Winkler, en donde puede verse que la "a" de cargo y la "a" de grado hacen match al estar en las posiciones 2 y 3 respectivamente. Algo análogo ocurre con las letras "r" de ambas cadenas y también las letras "o" del final, ya que coinciden en su posición, sin embargo, las letras "g" no hacen match al estar alejadas por más de una posición. Así el número de match es igual a 3 (a,r,o) (D. Campos, 2019).

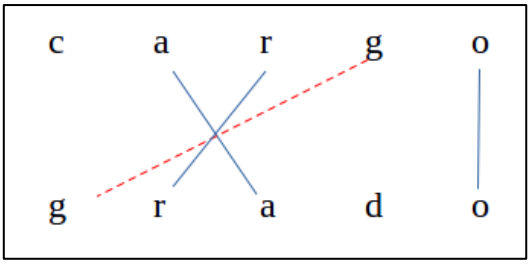


*Figura 14: Ejemplo de Uso de Distancia Jaro-Winkler.*

### 2.3.5. Otras Distancias

Existen otras medidas de distancia para variables binarias, que son derivaciones de las anteriores descritas y que cumplen con otras características, sus ventajas y desventajas (Grootendorst, 2021).

Estas solo son nombradas aquí:

- **Distancia Manhattan**. Es la distancia entre 2 puntos calculada como la longitud de cualquier camino que los una mediante segmentos verticales y horizontales, todos miden lo mismo. También conocida como la Distancia del Taxista.

- **Distancia Minkowski**. Esta distancia puede considerarse una generalización de las distancias Euclidianas y Manhattan.

- **Distancia Jaccard**. Métrica que opera sobre conjuntos, por lo que se utiliza para comparar párrafos completos como un conjunto de palabras.

- **Distancia Chebyshev**. Esta distancia es la mayor diferencia entre sus dimensiones. También conocida como la distancia del tablero de ajedrez.

- **Distancia Sorensen-Dice**. Es una métrica relacionada con el índice Jaccard. Debido a esta relación y al hecho de que no satisface la desigualdad de triangulo.

## 2.4. Aprendizaje de Máquinas

Se conoce como aprendizaje de máquinas o *Machine Learning* a una rama de la inteligencia artificial. Proceso en el cual se utiliza un sistema que puede aprender de la información (Mitchell, 1997) para presentar luego predicciones. Para poder implementar aprendizaje automático debe existir un patrón subyacente en los datos, ya que es necesario que la pertenencia de los datos a una clase u otra (en el caso de un problema clasificación) esté dada por un patrón que explique el por qué un objeto pertenece a una clase particular. Además, se debe contar con información, ya que, el aprendizaje automático aprende típicamente de los datos históricos.

En Figura 15, se aprecia cómo se estructuran los dos tipos métodos clásicos del aprendizaje automático (Slim, 2018).

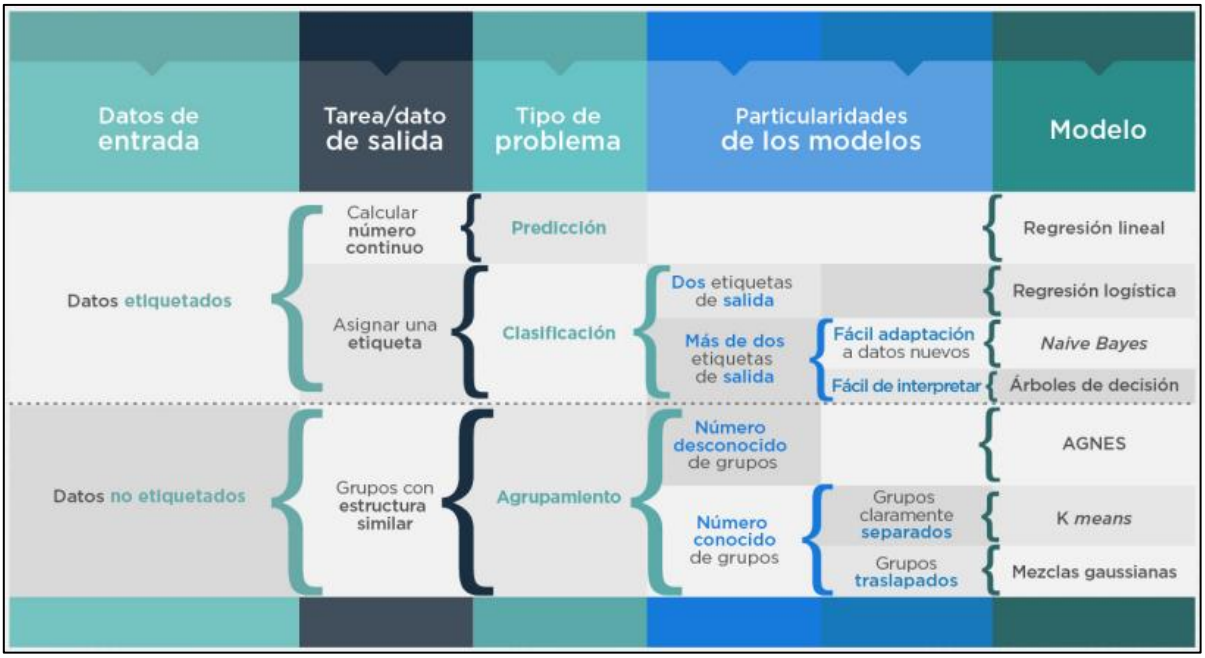


*Figura 15: Estructura de Machine Learning (Fuente: Fundación Carlos Slim, A.C.).*

En el mundo de aprendizaje automático, se distinguen dos grupos de métodos:

- **Aprendizaje No Supervisado.** Difiere del aprendizaje supervisado, puesto que no hay un conocimiento a priori, y su objetivo es describir ciertas características del conjunto de datos de entrada, y entender como estos se encuentran organizados. Uno de sus principales ejemplos son los métodos de *clustering*.

  No se cuenta con datos etiquetados y sirve para resolver, *Problemas de Agrupamiento* en donde el programa divide el conjunto de datos en grupos con características similares. En la Figura 16, se muestra un ejemplo.

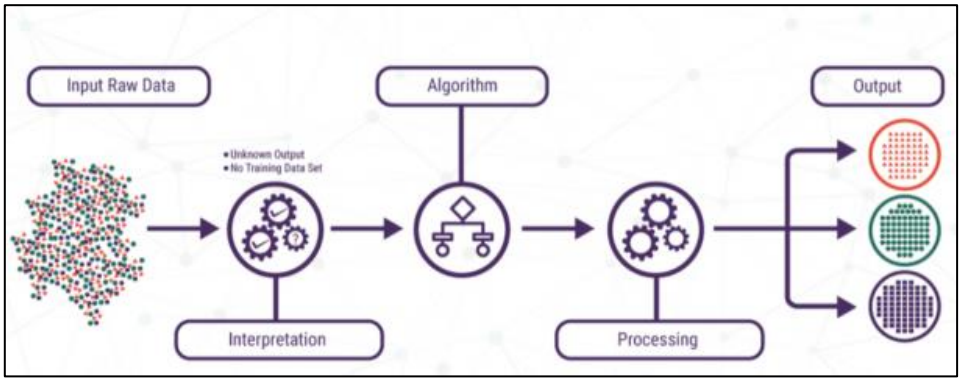


*Figura 16: Diagrama de Aprendizaje No Supervisado.*

- **Aprendizaje Supervisado**. Construye una función a partir de los datos de entrenamiento. Los datos de entrenamiento consisten en pares de objetos con los datos de entrada y sus etiquetas correspondientes. El resultado de la evaluación de la función en una observación puede ser tanto un valor numérico o una etiqueta de clase. El objetivo principal es crear una función capaz de predecir el valor correspondiente a cualquier objeto de entrada después de haber visto una serie de ejemplos (Mitchell, 1997).

  Se cuentan con datos etiquetados y sirven para resolver, *Problemas de Regresión* (valores continuos) en donde el programa obtiene datos de entrada y predice datos numéricos de salida para cada uno de ellos y *Problemas de Clasificación* (valores discretos) en donde el programa obtiene datos de entrada y especifica la categoría a la cual pertenece cada uno de ellos. En la Figura 17, se muestra un ejemplo.

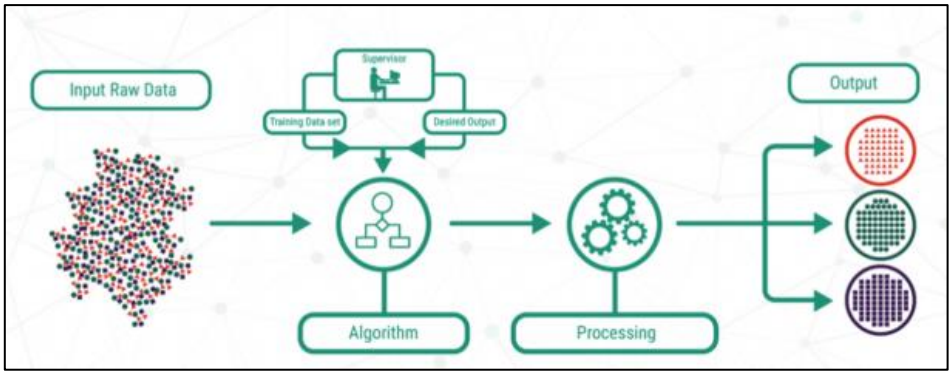


*Figura 17: Diagrama de Aprendizaje Supervisado - Clasificación.*

La clasificación de Clientes realizada en este trabajo utiliza el método de aprendizaje supervisado, ya que al algoritmo se le proporcionan las muestras de entrenamiento previamente etiquetadas.

### 2.4.1. Support Vector Machine

El método de clasificación Support Vector Machine (Maquinas de Soporte Vectorial) o SVM, originalmente fue desarrollado como un método de clasificación binaria. Sin embargo, su aplicación se ha extendido a problemas de clasificación múltiple y regresión. Ha resultado ser uno de los clasificadores con buenos resultados en múltiples tareas (Vapnik, 1999).

El modelo SVM se consideran los datos como puntos en un espacio de alta dimensionalidad y se busca generar el mejor hiperplano separador entre los elementos de distintas clases. Se distinguen tres elementos de este algoritmo: el hiperplano, los vectores de soporte y el margen. El hiperplano es el límite de decisión entre clases, por ende, separa datos en función de la clase a la que corresponda. Los vectores de soporte son aquellos datos que están cerca del hiperplano o que está mal clasificados. Por último, el margen es un espacio formado por dos hiperplanos paralelos al del límite de decisión y el punto más cercano (Vapnik, 1999).

Para lograr el objetivo el algoritmo genera los hiperplanos que logran segregar clases de mejor forma. Luego se maximiza el margen del plano usando vectores de soporte donde, finalmente, el hiperplano definitivo será aquel que logre mayor margen (Vapnik, 1999).

Usualmente se hace necesario realizar transformaciones de los datos para poder realizar la separación espacial, por lo que se utilizan funciones *Kernel* que permiten trabajar en un espacio de características de mayor dimensionalidad. Los más comunes son el *Polinomial*, *Sigmoide* y *RBF* (Radial Base Functions) Gaussiano.

En la Figura 18, se aprecia la aplicación de SVM Lineal, ya que el hiperplano es simplemente una recta que logra separar los datos.

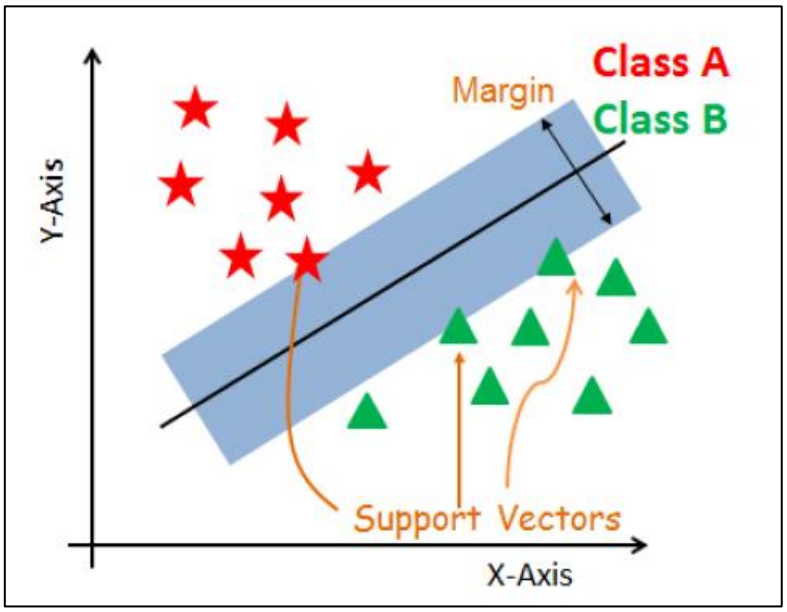


*Figura 18: Gráfico de Clasificación SVM Lineal (Fuente: https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python).*

## 2.4.2. K Nearest Neighbors

El método de clasificación K Nearest Neighbors (K Vecinos Más Cercanos) o KNN, es un Clasificador de aprendizaje supervisado no paramétrico que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual. Si bien se puede usar para problemas de regresión o clasificación, generalmente se usa como un algoritmo de clasificación, partiendo de la suposición que se pueden encontrar puntos similares cerca uno del otro aprendidos en la etapa de entrenamiento (Mucherino et al., 2009).

Este Clasificador está basado en instancias, lo que quiere decir que nuestro algoritmo no aprende explícitamente. En cambio, memoriza las instancias de entrenamiento que son usadas como "base de conocimiento" para la fase de predicción (IBM Analytics, 2021).

Aquí se aprecian dos clases de familias A y B. La K es la cantidad de "puntos vecinos" que se seleccionen en el entrenamiento, y que cada vez que se modifique, cambiará la mayoría de los votos. En la Figura 19, se muestra un ejemplo.

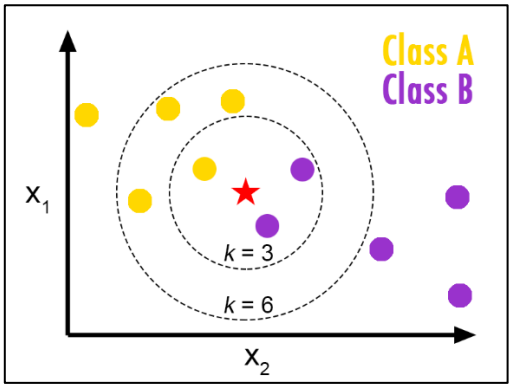


*Figura 19: Ejemplo del Método K Nearest Neighbors (Fuente: https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d).*

Un ejemplo práctico, del uso de este método de clasificación se aprecia en la Figura 20.

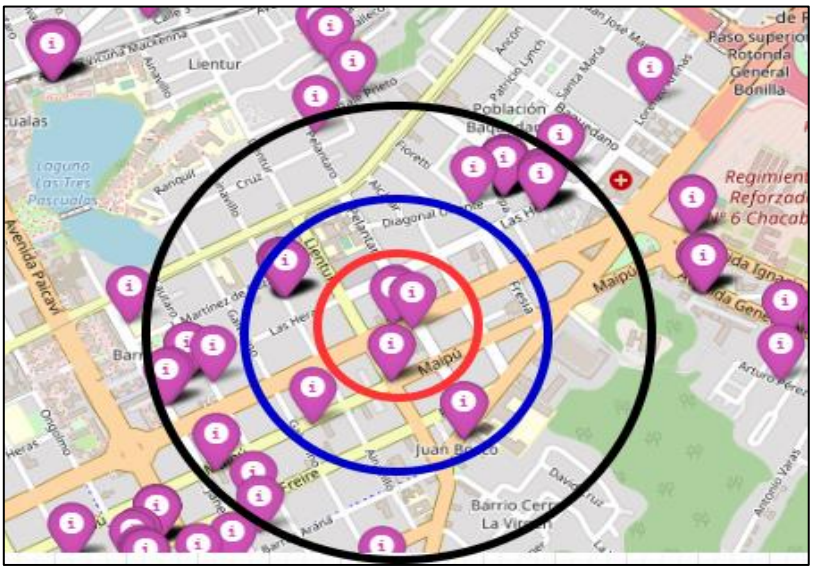


*Figura 20: Ejemplo Práctico del Método K Nearest Neighbors.*

## 2.4.3. Decision Trees

El método de clasificación Decision Trees (Árboles de Decisión) o DT, son representaciones gráficas en forma de árbol de posibles soluciones a una decisión basadas en ciertas condiciones, es una de los algoritmos de aprendizaje supervisado más utilizado en Machine Learning y pueden realizar tareas de clasificación o regresión (Kotu & Deshpande, 2018).

Los árboles de decisión tienen un primer nodo llamado raíz (root) y luego se descomponen el resto de atributos de entrada en dos o más ramas planteando una condición que puede ser cierta o falsa. Se bifurca cada nodo en 2 y vuelven a subdividirse hasta llegar a las hojas que son nodos finales y que equivalen a respuestas a la solución.

En la Figura 21, se ve un ejemplo de la estructura de un árbol de decisión.

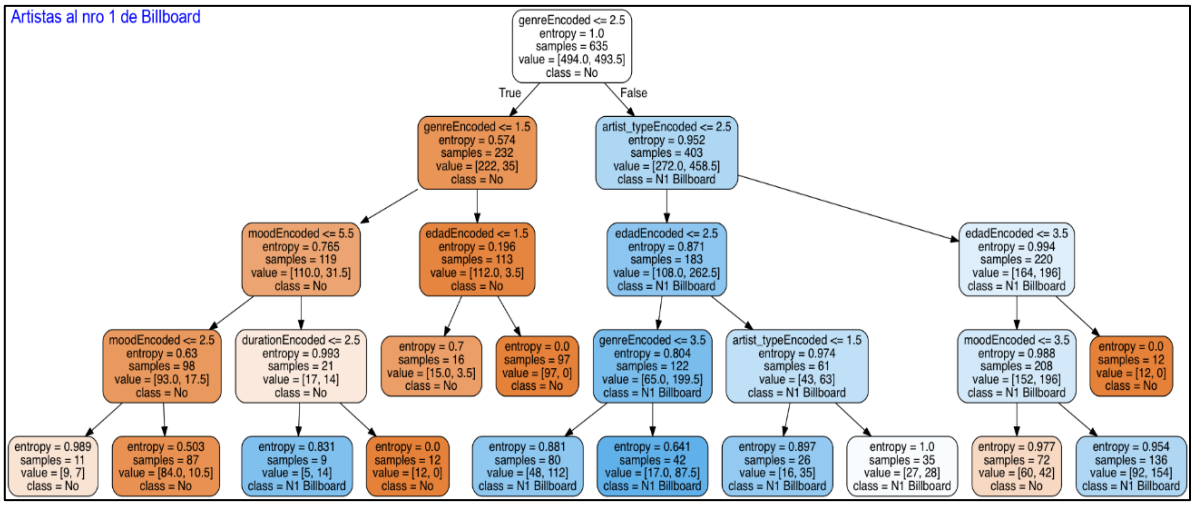


*Figura 21: Árbol de Decisión (Fuente: https://www.aprendemachinelearning.com).*

Dentro de los tipos de árboles de decisión se encuentre el tipo Classification And Regression Tree (Árbol de Clasificación y Regresión) o CART, el cual, es un tipo de árbol de decisión binario, esto es, en cada nodo sólo se pueden dividir los datos en dos grupos. Este tipo, es capaz de manejar tanto datos numéricos como categóricos, e incluso posee sofisticados métodos para tratar valores perdidos, como por ejemplo el uso de variables sustitutas. Otra característica, es que se puede usar en problemas de clasificación como de regresión.

Hay varios otros tipos de árboles de decisión, una descripción más detallada de ellos se encuentra en (Singh & Gupta, 2014).

- ID3 (Iteractive Dichotomiser 3)
- C4.5 (Sucessor of ID3)
- CHAID (CHI-squared Automatic Interaction Detector).

### 2.4.4. Random Forest

El método de clasificación Random Forest (Bosques Aleatorios) o RF son un conjunto de árboles predictores que a través de su voto mayoritario clasifican un elemento. Así como todos los modelos descritos anteriormente, un árbol binario también tiene problemas de sesgo y varianza (Breiman, 2001).

Para reducir el sobreajuste, se han propuesto múltiples estrategias, pero una de las más usadas es generar múltiples muestras de largo k provenientes de la base de datos completa, así como también de las variables de estas, tras lo cual se entrenan distintos modelos para cada muestra, agregando los resultados finales. De esta forma se obtiene un estimador promedio con menos varianza, evitando también el sobreajuste respecto a los modelos de aprendizaje automático clásicos (Kroese et al., 2019).

A esta estrategia de múltiples modelos se les llama *bosques aleatorios* en la cual se utilizan varios árboles de decisión tratados con la técnica de *bagging* para reducir la varianza de las predicciones. Esta técnica lo que hace es generar subconjuntos dentro del set de entrenamiento para que la correlación de las variables, si es que existe, no afecte a los resultados, reduciendo la varianza. En la Figura 22, se muestra un ejemplo.

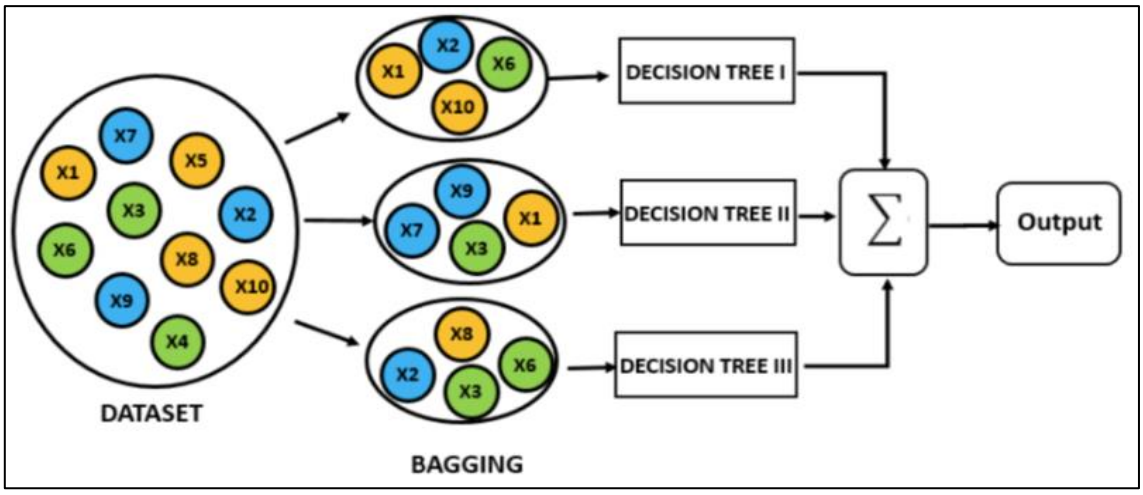


*Figura 22: Ilustración del Proceso Bagging - Fuente:*(Orellana, 2018).

Los bosques aleatorios generan dos medidas de importancia:

1. **MDA (Mean Decrease Accuracy).** Esta medición de importancia se basa en la contribución de la variable al error de predicción, es decir, al error de mal clasificados. Para determinar la importancia de cada una de las variables se permutan aleatoriamente los valores de esa variable en particular, dejando intacto el resto de las variables, y se vuelven a clasificar los mismos individuos según el mismo árbol, pero ahora con la variable permutada. Este valor se obtiene como la medida de los incrementos en todos los árboles donde actúa la variable (Liaw & Wiener, 2002).

2. **MDG (Mean Decrease Gini).** Esta medición se obtiene del índice *Gini*, el cual mide la impureza de cada nodo una vez que se haya seleccionado la variable de división de éste.

## 2.5. Definición de Métricas y Evaluación de Rendimiento

Para medir el comportamiento de un Clasificador se utilizan métricas que en conjunto generan información útil para tomar decisiones respecto al desarrollo de modelamiento y uso del Clasificador.

Para medir el rendimiento de un modelo de clasificación se pueden utilizar diversas métricas, pero una de las más utilizadas es la matriz de confusión, en donde se puede detectar fácilmente los casos en que el modelo acierta a las clases correctas y cuando se equivoca en la clasificación. En una matriz de confusión de una clasificación binaria como se observa en la Figura 23, se puede identificar la información sobre los aciertos (diagonal) y errores (fuera de la diagonal) (Dinov, 2018):

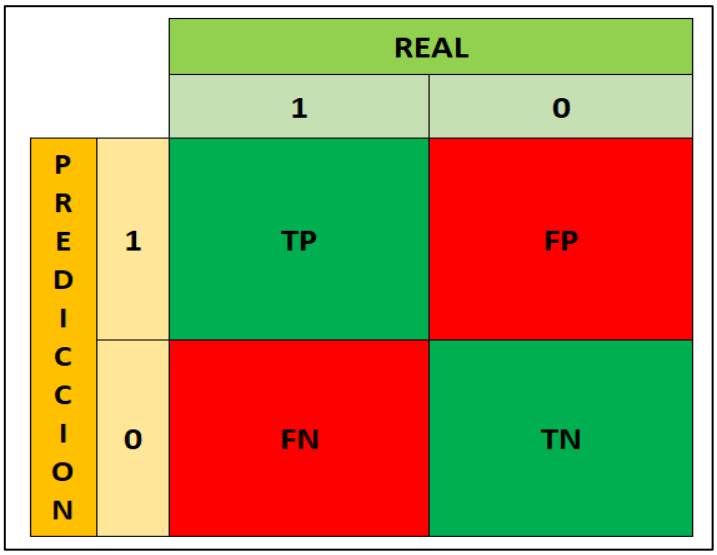


*Figura 23: Matriz de Confusión Binaria.*

- **TP – True Positives**. Verdadero positivo. Cuando la clase real es 1 y la resultante también lo es. Una persona está enferma y la prueba así lo demuestra.

- **TN – True Negatives**. Verdadero negativo. Cuando la clase real es 0 y la resultante también es 0. Una persona no está enferma y la prueba así lo demuestra.

- **FP – False Positives**. Falso positivo. Cuando la clase real es 0 y la pronosticada es 1. Una persona no está enferma, pero la prueba nos dice de manera incorrecta que si lo está.

- **FN – False Negatives**. Falso negativo. Cuando la clase real es 1 y la pronosticada es 0. Una persona está enferma, pero la prueba nos dice de manera incorrecta que no lo está.

Las matrices de confusión también pueden efectuarse en clasificación multiclases, y los indicadores anteriores se ampliarían para la detección de las demás etiquetas, pero su significado vendría a ser lo mismo, el escenario real es cuando se tiene un valor de 0 en FP y FN. Además, a partir de estas cifras, se puede calcular indicadores que dan más claridad al desempeño del modelo (Liu, 2020).

- **Accuracy (Exactitud).** Se define como la cantidad de detecciones correctas dentro del total de predicciones. Es considerada la medida más directa de calidad de los clasificadores. En caso de clases con gran desbalance, no se usará sola, puesto que no es capaz de representar el rendimiento de clases minoritarias. En la Figura 24 se muestra la fórmula.

$$\boldsymbol{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP}.$$

*Figura 24: Formula de Accuracy.*

- **Recall (Sensibilidad)**. Se define como la cantidad de detecciones correctas positivas dentro del total de positivos verdaderos. Responde a "¿Qué cantidad de datos de una clase clasifica correctamente?". En la Figura 25, se muestra la fórmula.

$$\boldsymbol{Recall} = \frac{TP}{TP + FN}.$$

*Figura 25: Formula de Recall.*

- **Precision (Precisión)**. Se define como la cantidad de detecciones correctas positivas dentro del total de elementos identificados como positivos. Con esta métrica se puede medir la calidad del modelo de Machine Learning. Responde a "¿Cuántos datos pertenecen a dicha clase?". En la Figura 26, se muestra la fórmula.

$$\boldsymbol{Precision} = \frac{TP}{TP+FP}.$$

*Figura 26: Formula de Precision.*

- **Specificity (Especificidad)**. Se define como la cantidad de detecciones correctas identificadas como negativas fuera del total de negativas. En la Figura 27, se muestra la fórmula.

$$\boldsymbol{Specificity} = \frac{TN}{TN+FP}.$$

*Figura 27: Formula de Specificity.*

- **F1**. El valor F1 se utiliza para combinar las medidas de Precision y Recall en un solo valor. Esto es práctico porque hace más fácil el poder comparar el rendimiento combinado de la precisión y la exhaustividad entre varias soluciones. En la Figura 28, se muestra la fórmula.

$$\boldsymbol{F1} = \frac{2*Precision*Recall}{Precision+Recall}.$$

*Figura 28: Fórmula de la Métrica F1.*

En la Figura 29, se muestra un resumen con todas las métricas asociadas a la matriz de confusión (Barrios, 2019).

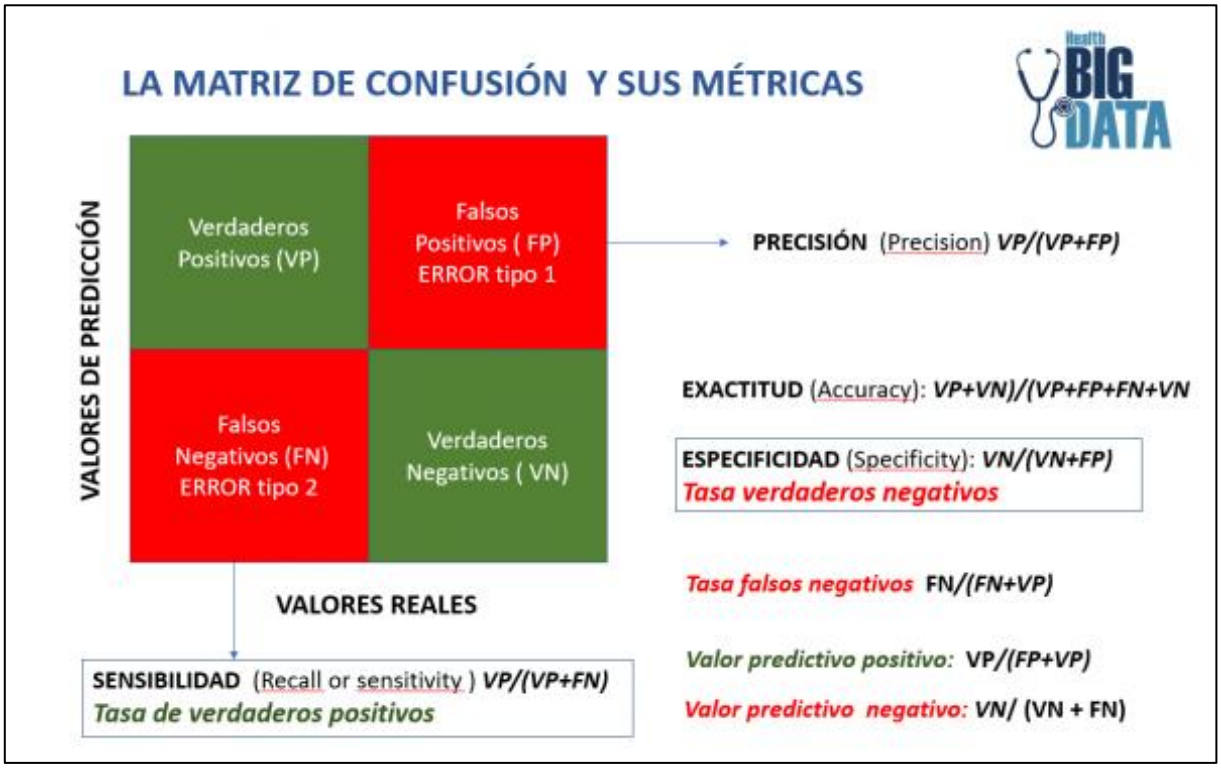


*Figura 29: Matriz de Confusión y sus Métricas (Fuente: https://www.juanbarrios.com).*

En la Figura 30, se observa gráficamente el comportamiento de las métricas de Exactitud (Accuracy) y Precisión (Precision), dependiendo del resultado obtenido y cómo están los datos distribuidos.

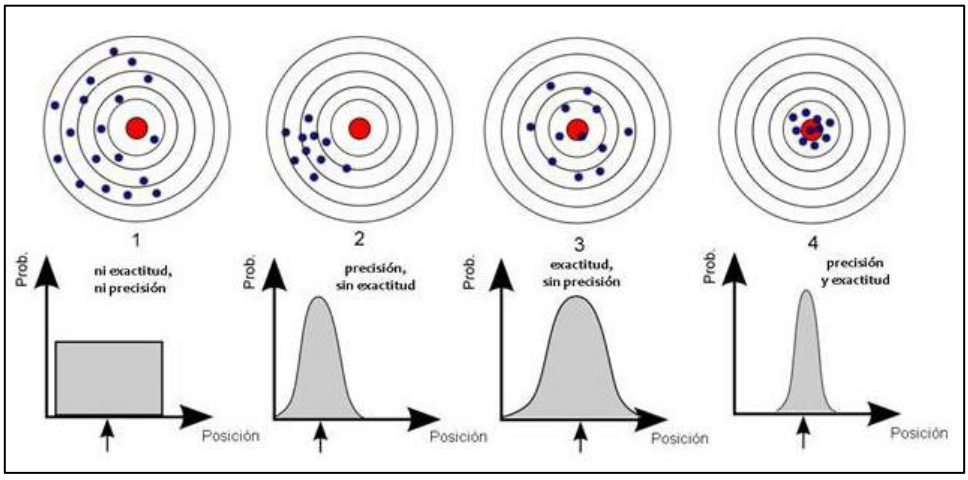


*Figura 30: Comportamiento de Métricas (Fuente: https://www.juanbarrios.com).*

- **Curva ROC**. Se conoce como Curva de Características Operativas de Receptor (ROC, por sus siglas en inglés), la cual permite establecer la capacidad de discriminación de una prueba de clasificación dicotómica (que admite solo dos respuestas posibles) presencia o ausencia de la variable de interés (Del Valle, 2017). Permite elegir el mejor valor umbral para una prueba y minimizar los errores cometidos por una mala (incorrecta) clasificación (Markam & DataSchool, 2014).

  El eje vertical se define como sensibilidad de la prueba, la proporción de objetos que presentan la característica de interés (la prueba indica que pertenecen al grupo con esa característica). El eje horizontal se define como el complemento de la especificidad de la prueba (viene dado por la fórmula *1 – especificidad*), la proporción de objetos que no presentan la característica de interés (son clasificados en el grupo que no posee la característica) (Del Valle, 2017). En otras palabras, la sensibilidad mide los verdaderos positivos; y la especificidad, los verdaderos negativos. En la Figura 31, se muestra la Curva ROC.

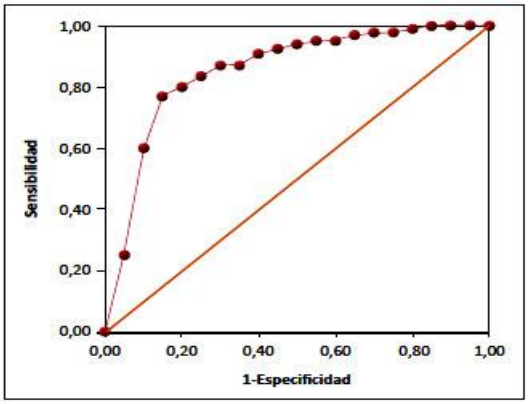


*Figura 31: Curva ROC (Fuente: https://www.scielo.cl/).*

Las curvas ROC pueden distinguirse por su grado de convexidad, A) poca convexidad significa un área pequeña entre la diagonal y la curva, lo cual indica que el modelo es deficiente; B) un área mayor entre la diagonal y la curva indica un modelo de decisión mejor; y C) la mejor curva presenta un área mayor, su crecimiento es muy rápido al inicio y comienza a ser casi horizontal en valores altos de sensibilidad, lo que permite elegir un mejor valor umbral (J. Campos, 2021). En la Figura 32, se muestran Curvas ROC con distintos grados de convexidad.

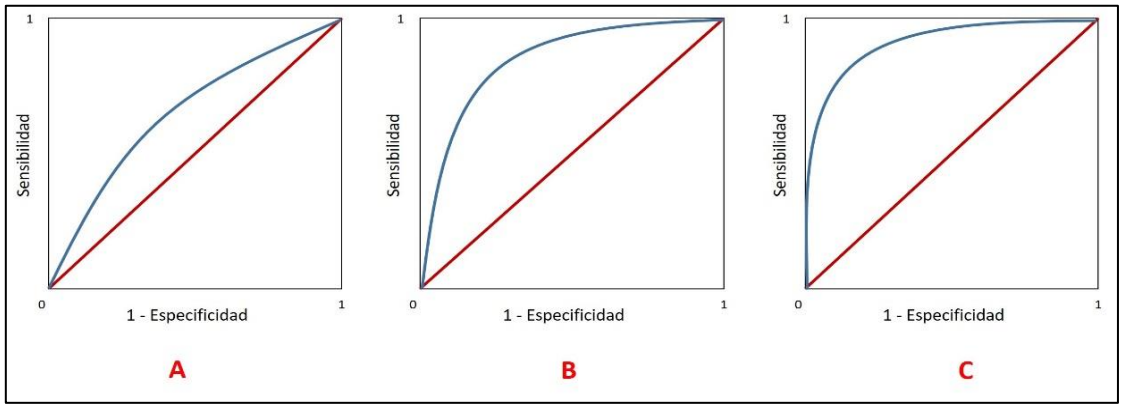


*Figura 32: Curvas ROC Distintos Grados Convexidad (Fuente: https://www.scielo.cl/).*

- **AUC.** Se conoce como Área Bajo la Curva (AUC, por sus singlas en inglés) y se utiliza como un resumen del rendimiento del modelo. Cuanto más esté hacia la izquierda la curva, más área habrá contenida bajo ella y, por ende, mejor será el Clasificador. El Clasificador Aleatorio tendría una AUC de 0.5 mientras que el Clasificador Perfecto tendría un AUC de 1. En la Figura 33, se muestran los distintos valores.

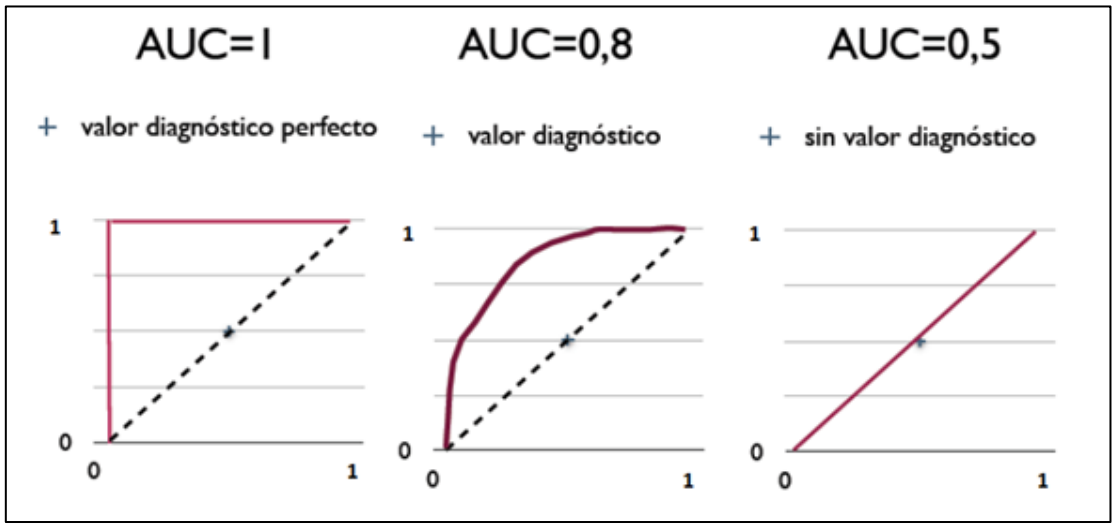


*Figura 33: Diferentes Tipos de AUC (Fuente: https://www.scielo.cl/).*

# 3. Desarrollo del Trabajo

## 3.1. Definición de los Datos a Analizar

La definición del Modelo de Datos se comenzó por etapas, desde la selección de las fuentes de datos hasta la creación de la base de datos.

### 3.1.1. Selección de la Fuente de Datos

En esta sección se identificaron y clasificaron todas las fuentes de datos donde se almacena y procesa la información de los Clientes, dando como resultado diez fuentes, las cuales representan casi el **90%** de la información que maneja la Empresa.

Cada fuente de datos fue clasificada según el rol que cumple, ya sea, paquetería internacional y nacional, giros nacionales e internacionales, postal nacional e internacional, servicio al cliente, sucursal virtual, etc. En la Tabla 3, se aprecia la distribución de las fuentes de datos.

| Paquetería Internacional | | Paquetería Nacional | Postal Internacional | | Postal Nacional | Giro Internacion | Giro Nacional | Avisos | | | Sucursal Virtual | Servicio Clientes | Padrón Electoral |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ALI | WISH | Alertran | CDS | GDA | Sisve | Moneygra | Giros | Avisos | Sitrac | TGR | SV | REC | BPE |

*Tabla 3: Distribución de Fuentes de Datos.*

### 3.1.2. Identificación de los Campos

En esta sección se hizo un análisis a las fuentes de datos y se procedió a realizar la selección y clasificación de los campos que tienen relación con información de los Clientes y que ayudarán al proceso de identificación de estos.

Estos campos fueron seleccionados de tal forma que permita poder discriminar un Cliente de otro, como se aprecia en la Figura 34.

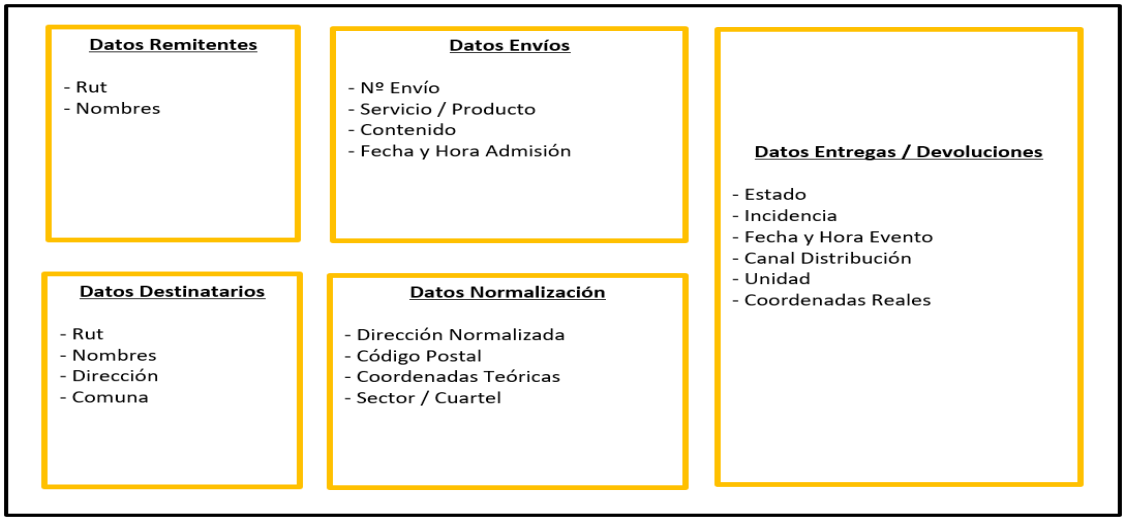


*Figura 34: Identificación de los Campos.*

## 3.1.3. Consolidación de los Campos

En la consolidación de los campos se analizó la calidad de ellos, clasificándolos en "**Estables**" y "**No Estables**". En donde los campos clasificados como "**Estables**", son los campos que siempre contienen información para poder ser trabajada. Los campos clasificados como "**No Estables**", son los campos que no siempre contienen información o no es válida para poder ser trabajada. En la Tabla 4, se aprecia la consolidación de estos campos.

| Fuentes de Datos | Paquetería Internacional | | Paquetería Nacional | Postal Internacional | | Postal Nacional | Giro Internacion | Giro Nacional | Avisos | | | Sucursal Virtual | Servicio Clientes | Padrón Electoral |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ALI | WISH | Alertran | CDS | GDA | Sisve | Moneygra | Giros | Avisos | Sitrac | TGR | SV | REC | BPE |

| # | Campos | Datos al Ingresar o Admitir la Transacción | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Rut | Estable | Estable | No Estable | No Existe | Estable | No Estable | No Estable | No Estable | No Estable | No Estable | Estable | Estable | Estable | Estable |
| 2 | Nombre Cliente | Estable | Estable | No Estable | No Estable | No Estable | No Estable | Estable | Estable | No Estable | No Estable | No Estable | Estable | No Estable | Estable |
| 3 | Nombres | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | Estable | Estable | No Estable | No Estable | No Estable | Estable | No Estable | Estable |
| 4 | Apellidos | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | Estable | Estable | No Estable | No Estable | No Estable | Estable | No Estable | Estable |
| 5 | Direccion | No Estable | No Estable | No Estable | No Estable | Estable | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | Estable | No Estable | Estable |
| 6 | Comuna | Estable | Estable | No Estable | No Estable | Estable | Estable | Estable | Estable | Estable | No Estable | Estable | Estable | No Estable | Estable |
| 7 | Código Postal | Estable | Estable | No Estable | No Estable | Estable | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | Estable | No Estable | Estable |
| 8 | Telefono | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | Estable | Estable | No Estable |
| 9 | Email | Estable | Estable | No Estable | No Estable | Estable | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | Estable | Estable | No Estable |
| 10 | X, Y Teóricas | No Estable | No Estable | Estable | No Estable | No Estable | Estable | No Estable | No Estable | No Estable | No Estable | No Estable | Estable | No Estable | Estable |
| 11 | Nº Envío | Estable | Estable | Estable | Estable | Estable | Estable | No Existe | No Existe | Estable | Estable | Estable | Estable | Estable | No Existe |

| # | Campos | Datos al Entregar la Transacción | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 12 | X, Y Reales | Estable | Estable | Estable | No Estable | No Estable | Estable | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable | No Estable |

*Tabla 4: Consolidación de los Campos.*

## 3.1.4. Diseño del Modelo de Datos

Después de realizar la consolidación de los campos relevantes, se procedió a realizar el diseño del Modelo de Datos de Clientes y la creación de la base de datos, que permita el almacenamiento de toda la data requerida.

Teniendo como identificador el Rut del Cliente y la generación de un Id de cliente interno que permita la relación entre las diferentes tablas que componen este modelo. En la Figura 35, se muestra el diagrama.

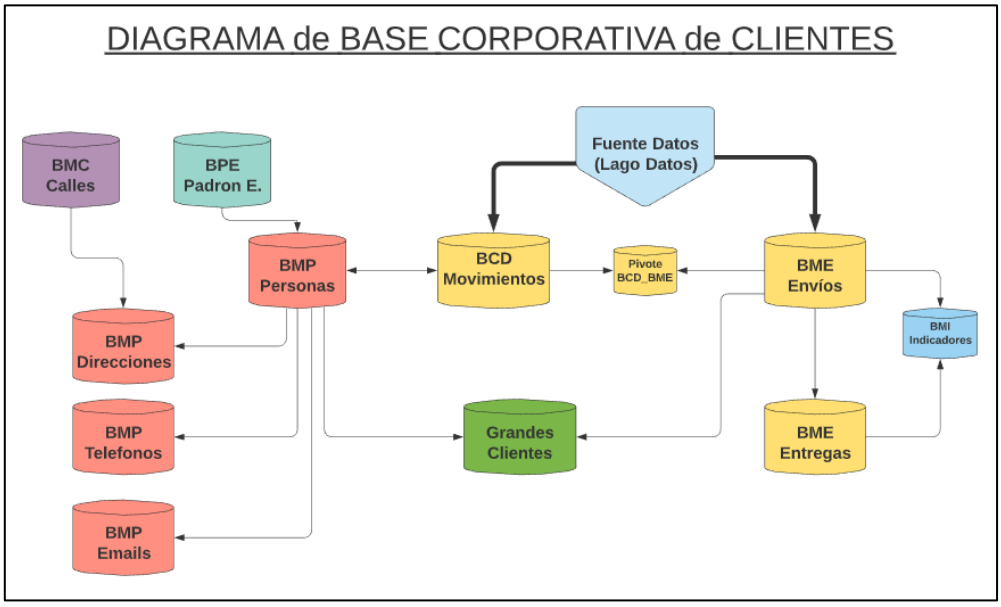


*Figura 35: Diagrama de Base Corporativa de Clientes.*

### 3.1.5. Corrección de los Datos

En este proceso se realizó la corrección de los datos para que técnicamente puedan ser leídos y procesados por las herramientas de Extracción, Transformación y Carga, ETL (Extract, transform and load).

Luego se procedió a:

- Cambiar el formato de los archivos que vienen desde el origen (csv, xls, xml, json, etc.), ya sea, en formatos más livianos como 'TXT'.
- Estandarizar el separador de columnas de los campos en los archivos TXT, dejando como primera instancia el punto y coma ";" y en segunda instancia el doble pipe "||".
- Corregir codificación de texto (ASCII, UTF-8, ISO-8859-1, etc.) quedando con el formato que soporta los caracteres acentuados y la letra 'ñ'.
- El formato de fecha se estandarizó (dd-mm-yyyy, yyyymmdd, mm-dd-yy, etc.), quedando con el siguiente formato 'yyyy-mm-dd'.

Por temas de sesgos con los datos, se tuvo precaución de no eliminar información valiosa que puede ser relevantes para los procesos de exploración de la data.

### 3.1.6. Limpieza de los Datos

En esta etapa se realizaron la limpieza de los datos relevantes, para así poder almacenarlos con una buena calidad, estos datos fueron los siguientes:

- **Rut**. Se realizó la limpieza de este dato, solo permitiendo el ingreso de Rut que fueron validados por la rutina de validación de Rut (Algoritmo para obtener y validar el dígito verificador). Además de eliminar Rut de pruebas, tales como '1-9', '2-7', '11111111-1', Etc.
- **Nombres y Apellidos**. Se realiza la limpieza de estos datos, aplicando la rutina de expresión regular que acepte solo letras y el largo del dato sea válido.
- **Email**. Se realiza la limpieza de este dato, aplicando la rutina de expresión regular que valida que estructuralmente el Email sea válido, ya sea, 'nombre@dominio'.
- **Teléfono**. Se realiza la limpieza de este dato, aplicando la rutina de expresión regular que solo acepte números y el largo del dato sea válido.
- **Dirección y Comuna**. Se realiza la limpieza del dato, extrayendo todos los caracteres especiales, solo aceptando números y letras. Para el caso de la comuna se valida con una tabla de Comunas.
- **Coordenadas Georreferenciales**. Se realiza la limpieza del dato, solo aceptando números decimales negativos.
- **Fecha**. Que solo contenga fechas válidas y que ellas no tengan fechas mayores al día de hoy o muy antiguas, definiendo un rango válido.

### 3.1.7. Integración de los Datos

Con respecto a la integración de datos, se realizaron cruces con fuentes de datos oficiales, en las cuales su codificación y descripción está validada por organismos del Estado de Chile.

Para las Comunas, Provincias, Regiones, etc. en la cual se usó el código CUT (Código Único Territorial) que es una clasificación oficial del territorio, su fuente es la Subsecretaría de Desarrollo Regional y Administrativo dependiente del Ministerio de Bienes Nacionales.

Para los Rut Naturales, estos se validaron con el Padrón Electoral Año 2020, en donde su fuente oficial es el Servicio Electoral de Chile, que es un órgano autónomo y no tiene dependencias de los Ministerios.

Para los Rut Jurídicos, estos se validaron con el Registro de Personas Jurídicas y Empresas, en donde su fuente oficial es el Servicio de Impuestos Internos dependiente del Ministerio de Hacienda.

### 3.1.8. Definición del Universo de los Datos

Según la situación detectada en los pasos anteriores, en donde los Clientes al no contar con el Rut, el proceso de identificación se hacía muy complicado y la duplicidad de la información se estaba haciendo muy recurrente.

Primero se procedió a seleccionar el DataSet (Conjunto de Datos) que servirá para realizar los procesos de entrenamiento, clasificación e identificación de Clientes. Y para esto se acotó el universo de datos a lo siguiente:

- Sistema = **Paquetería Internacional**
- Año = **2020**
- Comuna = **Concepción**
- Estado = **Entregado**
- Total = **250 mil (aproximado)**

Obteniendo un **Universo Real** de aproximadamente "**239 mil envíos**", los cuales representan el **3%** de las entregas realizadas en paquetería internacional en todo el país, para el año 2020. En la Tabla 5, se muestra el detalle de las entregas por mes.

| CONCEPCION - Año = 2020 | | | | |
|---|---|---|---|---|
| Mes | Universo Total | Universo Real | Acumulado | Muestra |
| Ene | 17.099 | 16.418 | 16.418 | 300 |
| Feb | 8.966 | 8.362 | 24.780 | 300 |
| Mar | 11.805 | 11.651 | 36.431 | 300 |
| Abr | 9.696 | 9.061 | 45.492 | 300 |
| May | 17.743 | 17.743 | 63.235 | 300 |
| Jun | 17.867 | 16.734 | 79.969 | 300 |
| Jul | 28.720 | 26.553 | 106.522 | 300 |
| Ago | 21.712 | 20.334 | 126.856 | 300 |
| Sep | 23.397 | 20.931 | 147.787 | 300 |
| Oct | 32.680 | 32.345 | 180.132 | 300 |
| Nov | 36.574 | 34.914 | 215.046 | 300 |
| Dic | 24.393 | 23.922 | 238.968 | 300 |
| | 250.652 | 238.968 | | |

*Tabla 5: Universo de Datos Clasificados por Mes.*

En la Figura 36, se muestra la estructura del DataSet que servirá como universo para el trabajo de la Tesis.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 250652 entries, 0 to 250651
Data columns (total 14 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   Id_Bcd                  250652 non-null  int64
 1   Id_Marca                250652 non-null  float64
 2   Id_Persona              250652 non-null  int64
 3   Rut_Destinatario        250652 non-null  float64
 4   Nombre_Destinatario     250652 non-null  object
 5   Email_Destinatario      250652 non-null  object
 6   Telefono_Destinatario   250652 non-null  float64
 7   Direccion_Destinatario  250652 non-null  object
 8   Iata_Comuna             250652 non-null  object
 9   Comuna_Destinatario     250652 non-null  object
 10  Cp_Destinatario         250652 non-null  int64
 11  Fecha_Entrega           250652 non-null  object
 12  Latitud_Y_Real          250652 non-null  float64
 13  Longitud_X_Real         250652 non-null  float64
dtypes: float64(5), int64(3), object(6)
memory usage: 26.8+ MB
```

*Figura 36: Estructura del DataSet de Movimientos.*

En la Tabla 6, se muestra un ejemplo de los datos extraídos.

| Envío | Fecha y Hora | RUT | Nombre | Dirección | Comuna | Teléfono | Email |
|---|---|---|---|---|---|---|---|
| 0668 | 19-02-2020 16:45:00 | 0 | Juan Alejandro Zaragoza B. | Alameda 123 | Concepción | 569 9123 45XX | juanatoxx@yimail.com |
| 3577 | 15-03-2020 12:30:00 | 0 | Juan Zaragoza B. | Meiggs 321 | Concepción | 9123 45XX | juan.zaragozinnxx@llimail.cl |
| 1050 | 10-05-2020 21:30:00 | 0 | Juan Zaragoza Bahamondes | 321 Meiggs | Concepción | 9123 45XX | juan.zaragozinxx@llimail.cl |
| 4507 | 16-06-2020 09:30:00 | 0 | Juan Zaragoza B. | Alameda 123 | Concepción | | |

*Tabla 6: Muestra de Datos del DataSet.*

## 3.2. Exploración de los Datos

Después de realizada la carga histórica e incremental de los datos se procedió a realizar una exploración de la información cargada, detectándose casi el **70%** de los datos no tenía asociado el campo **Rut**, por lo cual se duplicaban muchos registros para un mismo Cliente, pero con diferentes **Id** (Códigos Únicos de Clientes).

Se procedió a aislar los datos para poder analizarla en detalle y tratar de encontrar algún proceso que ayudara a la identificación de Clientes únicos. Se analizaron los campos que componen el registro, encontrándose que la mayoría tenía una calidad aceptable en su información y que permitiría poder trabajar sobre ellos. Pero se llegó a la conclusión que el campo "**Nombre**" era el que tenía mayores ventajas con respecto a los otros, tales como, Correo Electrónico, Teléfono, Domicilio, Comuna, etc.

Se procedió al análisis del universo de data seleccionado y se identificaron diferentes variaciones textuales frecuentes (Amón & Jiménez, 2010), que son presentadas en la Tabla 7.

| Tipos | Problemas | Ejemplo Campo "Nombre" | % |
|---|---|---|---|
| 0 | Sin Problemas | Juan Alejandro Zaragoza Bahamondes<br>o<br>Juan Zaragoza Bahamondes | 40% |
| 1 | Tokens en desorden | Zaragoza Bahamondes Juan Alejandro | 1% |
| 2 | Tokens Faltantes: Eliminación de uno o más tokens | Juan Zaragoza | 32% |
| 3 | Abreviaturas: Truncamineto de uno o más tokens | Juan A. Zaragoza Baham | 4% |
| 4 | Errores ortográficos y tipográficos | Juan Alejandro Zaragoza Vahamondes | 9% |
| 5 | Espacios en blanco: Eliminación o adición de espacios y/o caracteres especiales | Juan Alejandro Zaragoza#Bahamondes | 12% |
| 6 | Préfijos/Sufíjos sin valor semántico: Presencia de caracteres al inicio y/o al final | Dr. Juan Alejandro Zaragoza Bahamondes , UCH | 3% |

*Tabla 7: Definición de los Tipos de Problemas.*

### 3.2.1. Principales Variaciones en Nombres de Clientes

Las principales variaciones en los nombres de los Clientes se muestran en los siguientes ejemplos y que corresponden a los Tipos de Problemas definidos en la sección anterior.

- **Tipo 0**. Este tipo de problema en realidad no es problema, es decir, estos registros fueron clasificado con este tipo, por ser registros que tienen el campo **Nombre** con todas sus características y atributos, ya sea, contienen Primer Nombre, Segundo Nombre (Opcional), Apellido Paterno y Apellido Materno. Solo que no cuentan con información en el campo **Rut**, lo que hace difícil poder identificarlo. Estos representan el **40%** del total de la data.

- **<u>Tipo 1</u>**. Este tipo de problema se da por el desorden en los ***tokens***[1]. En la Tabla 8, se muestran ejemplos. Estos representan solo el **1%** del total de la data.

| Id_Bcd | Nombre | Id_Marca | Clasificador | Tipo Problema |
|---|---|---|---|---|
| 6115075 | BONAPARTE ACEITUNO JUAN | 36 | x | 1 |
| 54147 | DIAZ JUAN | 184 | x | 1 |
| 4284384 | GONZALEZ JUAN | 192 | x | 1 |
| 2544200 | PEREZ MARIA | 573 | x | 1 |

*Tabla 8: Ejemplos del Tipo de Problema 1.*

- **<u>Tipo 2</u>**. Este tipo de problema se da por la falta de tokens. En la Tabla 9, se muestran ejemplos. Estos representan el **32%** del total de la data.

| Id_Bcd | Nombre | Id_Marca | Clasificador | Tipo Problema |
|---|---|---|---|---|
| 3917515 | PEDRO GODOY | 64 | x | 2 |
| 8287545 | JUAN GONZALEZ | 76 | x | 2 |
| 3422356 | MARIA DIAZ | 77 | x | 2 |
| 12705486 | JUANA RIVAS | 82 | x | 2 |
| 5795725 | ALEXIS SANCHEZ | 83 | x | 2 |
| 665100 | LUIS SOTO | 84 | x | 2 |

*Tabla 9: Ejemplos del Tipo de Problema 2.*

- **<u>Tipo 3</u>**. Este tipo de problemas se da por el abreviaturas y truncamientos de tokens. En la Tabla 10, se muestran ejemplos. Estos representan el **4%** del total de la data.

| Id_Bcd | Nombre | Id_Marca | Clasificador | Tipo Problema |
|---|---|---|---|---|
| 318358 | CATA JARA | 301 | x | 3 |
| 4657822 | A CAROLINA FUENTES | 302 | x | 3 |
| 2539594 | MA TERESA ROJAS | 304 | x | 3 |
| 573702 | MAFE DIAZ | 317 | x | 3 |
| 9230714 | MARIA DE LAS N ROJAS | 317 | x | 3 |
| 9231524 | ALI SOTO PEREZ | 317 | x | 3 |
| 3747013 | CARLOS ZU IGA | 319 | x | 3 |

*Tabla 10: Ejemplos del Tipo de Problema 3.*

- **<u>Tipo 4</u>**. Este tipo de problemas se da por errores ortográficos y tipográficos. En la Tabla 11, se muestran ejemplos. Estos representan el **9%** del total de la data.

| Id_Bcd | Nombre | Id_Marca | Clasificador | Tipo Problema |
|---|---|---|---|---|
| 1727065 | JUAN NUNEZ | 537 | x | 4 |
| 20017734 | ANTONIO MUNOZ | 538 | x | 4 |
| 12182183 | ANY PERES | 542 | x | 4 |
| 283389 | PEDRO GONSALES | 547 | x | 4 |
| 20074135 | ROSE MARI CARTAGENA | 551 | x | 4 |
| 8791497 | ANA BEL OROSCO | 553 | x | 4 |
| 2046347 | ELIZABE CARRENO | 558 | x | 4 |

*Tabla 11: Ejemplos del Tipo de Problema 4.*

[1] Se les denominan ***tokens*** a las entidades (Apellido Paterno, Apellido Materno y Nombres), que son las palabras que componen el campo Nombre.

- **Tipo 5**. Este tipo de problemas se da por espacios en blanco y caracteres especiales. en la Tabla 12, se muestran ejemplos. Estos representan el **12%** del total de la data.

| Id_Bcd | Nombre | Id_Marca | Clasificador | Tipo Problema |
|---|---|---|---|---|
| 11423090 | ELIZABE MU#OZ@@ | 551 | x | 5 |
| 12079895 | ERIC AVENDA##O | 558 | x | 5 |
| 10726170 | #123CARLOS O@ATE | 568 | x | 5 |
| 901829 | @@CARMEN #ROLD`AN | 571 | x | 5 |
| 6498436 | ANDR#ËS SIMçON | 572 | x | 5 |
| 7160825 | MARIA PORTI¨Çel | 573 | x | 5 |

*Tabla 12: Ejemplos del Tipo de Problema 5.*

- **Tipo 6**. Este tipo de problemas se da por prefijos y sufijos que contiene el campo Nombre. En la Tabla 13, se muestran ejemplos. Estos representan el **3%** del total de la data.

| Id_Bcd | Nombre | Id_Marca | Clasificador | Tipo Problema |
|---|---|---|---|---|
| 2679382 | MR CHRISTIAN PEREZ COTAPOS | 409 | x | 6 |
| 12963023 | JUAN DIAZ283723872398-RED_SALUD | 441 | x | 6 |
| 199355697 | DANIEL SOTO-RUT12345678-9 | 443 | x | 6 |
| 16250114 | ERICA_PARRA_ML20047265384 | 443 | x | 6 |
| 164332103 | DRA_JUANA_DIAZ | 544 | x | 6 |
| 20115352 | PHD-IVAN-SOTO-PUC | 567 | x | 6 |
| 6888163 | PASTOR DIAZ_IGLESIA LOS 10 MANDAMIENTOS | 579 | x | 6 |
| 26779639 | JUEZ LUIS DIAZ - 8AVO JUZGADO | 590 | x | 6 |

*Tabla 13: Ejemplos del Tipo de Problema 6.*

Posterior a la exploración de la data e identificación de las diferentes variaciones del campo **Nombre**. Se analizaron distintas opciones, ya sea, uso de técnicas avanzadas de reconocimiento, clasificación e identificación de Clientes, concluyendo que se debía testear varios métodos apoyados por información adicional que ya se tenía del Cliente, tal como, Coordenadas de Entregas. Y se concluyó lo siguiente.

Para los problemas clasificados como Tipo 1, 2 y 3, se procedió a aplicar métodos basado en "**Métodos de Clasificación"**. Dando énfasis al **Tipo 2 = Tokens Faltantes**. Esto representa un **32%** de los casos.

Para los problemas clasificados como Tipo 4, 5 y 6, se procedió a aplicar métodos basado en "**Métodos de Cálculos de Distancia"**. Dando énfasis al **Tipo 5 = Espacios en Blanco y Caracteres Especiales.** Esto representa un **12%** de los casos.

## 3.3. Identificación de Clientes Basado en Métodos de Distancia

En esta sección se realiza la identificación de Clientes basado en métodos de **Cálculos de Distancia** para poder solucionar los tipos de problemas analizados en la sección anterior y que tiene relación con errores ortográficos, espacios en blanco, caracteres especiales, prefijos y sufijos en el campo "**Nombre**".

### 3.3.1. Cálculo de Distancia de Levenshtein

El uso de este cálculo de distancia nos entregó el número mínimo de operaciones que se debían realizar para que un Cliente se identifique con otro. Al no estar escrito de la misma forma.

Se creó una rutina en lenguaje Python usando las librerías de Python-Levenshtein que ya contiene la función para la medición de la Distancia Levenshtein. Se definió como rango válido la distancia que corresponde a: "**Mayor o Igual que 0 y Menor o Igual a 2**". En la Tabla 14, se muestra un ejemplo de los cálculos.

| Id_Bcd A | Nombre A | Id_Bcd B | Nombre B | Medida |
|---|---|---|---|---|
| 6115075 | BONAPARTE ACEITUNO JUAN | 23701179 | BONAPARTE ACEITUNO JUAN | Levenshtein = 0 |
| 54147 | DIAZ JUAN | 7779734 | DIAZ JUAN | Levenshtein = 0 |
| 4284384 | GONZALEZ JUAN | 23913428 | GONZALEZ JUAN | Levenshtein = 0 |
| 2544200 | PEREZ MARIA | 7696817 | PEREZ MARIA | Levenshtein = 0 |
| 3917515 | PEDRO GODOY | 7853548 | PEDRO GODOY | Levenshtein = 0 |

*Tabla 14: Método de Cálculo de la Distancia Levenshtein.*

### 3.3.2. Cálculo de Similitud de Coseno

El uso de este cálculo de similitud nos entregó el valor entre dos cadenas para que un Cliente se identifique con otro. Al no estar escrito de la misma forma.

Se creó una rutina en lenguaje Python usando librerías para diseñando una función para la medición de la Similitud de Coseno (Ver Anexo 1). Se definió como rango válido la distancia que corresponde a: "**Mayor o Igual que 0,5 y Menor o Igual a 1**". En la Tabla 15, se muestra un ejemplo de los cálculos.

| Id_Bcd A | Nombre A | Id_Bcd B | Nombre B | Medida |
|---|---|---|---|---|
| 3917515 | PEDRO GODOY | 362998 | PEDRO GODOY | Coseno = 0.9999999999999998 |
| 8287545 | JUAN GONZALEZ | 12398672 | JUAN GONZALEZ | Coseno = 0.9999999999999998 |
| 3422356 | MARIA DIAZ | 35369823 | MARIA DIAZ | Coseno = 1.0000000000000002 |
| 12705486 | JUANA RIVAS | 23654 | JUANA RIVAS | Coseno = 1.0000000000000002 |
| 5795725 | ALEXIS SANCHEZ | 79992134 | ALEXIS SANCHEZ | Coseno = 0.9999999999999998 |

*Tabla 15: Método de Cálculo de la Similitud Coseno.*

## 3.4. Algoritmos Propuestos Para Identificar Clientes

En esta sección se desarrollan los algoritmos propuestos. Definiendo la **Cota Inferior** (Algoritmo Join) y la **Cota Superior** (Algoritmo Oráculo) y basándose en el universo de datos que se dispuso.

Para todos los algoritmos propuestos, se utilizó el método del "**Cálculo de Distancia de Haversine**", (Ver Anexo 2), para así poder determinar la distancia entre los puntos de entrega de los Clientes.

Además, se trabajó con el método de "**Clasificación K Nearest Neighbors**" para realizar un análisis de los resultados de las entregas y poder determinar para este trabajo de Tesis que el rango válido entre dos puntos sea "**Menor o Igual a 10 Metros**", y así poder ser considerado y clasificado.

En ocasiones solo existe un punto dentro del círculo, lo que significa que es "**Una Entrega, Un Cliente**", en cambio en otras ocasiones, se ven más puntos de entrega dentro del círculo, lo que significa que en esa dirección hay más de una entrega y posiblemente más de un Cliente o el mismo Cliente. Al graficarlo en un mapa en la Figura 37, se aprecian claramente los puntos de entrega y un círculo que representa el rango de los 10 metros a la redonda.

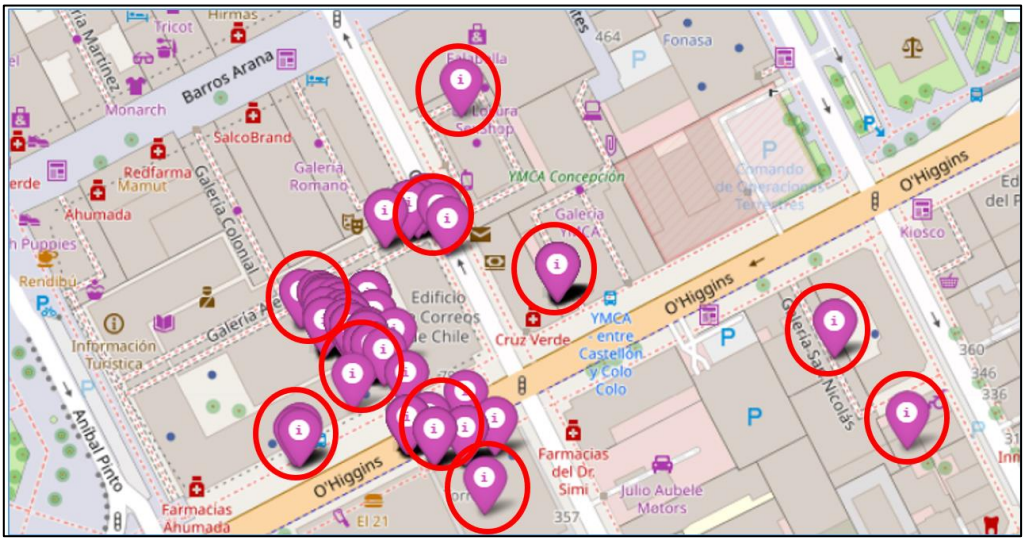


*Figura 37: Clasificación Usando el Método del Cálculo de Distancia Haversine.*

En la Tabla 16, se visualiza una muestra de los Clientes con sus coordenadas georreferenciales, los campos de Latitud, Longitud y Punto Geométrico.

| Id_BCd | Nombre Cliente | Dirección | Comuna | Latitud | Longitud | Geometry |
|---|---|---|---|---|---|---|
| 7325392 | Cliente_1 | Diagonal Los Tome 1345 | CONCEPCION | -36.615493 | -72.958180 | POINT(-72.95818 -36.61549) |
| 324567 | Clinete_2 | Cocrane 137 Block Interiores | CONCEPCION | -36.832046 | -73.055018 | POINT(-73.05501 -36.83204) |
| 5577345 | Cliente_3 | 2284 Anibal Pinto - Casa Azul con Rojo | CONCEPCION | -36.805556 | -73.064507 | POINT(-73.06450 -36.80555) |
| 875732 | Clinete_4 | Las Garzas 407 San pedro de la Paz | CONCEPCION | -36.829968 | -73.116857 | POINT(-73.11685 -36.82996) |
| 92312355 | Cliente_5 | Victorino Chacabuco 953 | CONCEPCION | -36.828956 | -73.044183 | POINT(-73.04418 -36.82895) |

*Tabla 16: Muestra de Clientes y las Coordenadas Georreferenciales.*

### 3.4.1. Algoritmo Oráculo – (Cota Superior)

En este algoritmo se definió la **Cota Superior** (Medida máxima a llegar), que se basa en suponer que se conoce el futuro (posición geográfica donde se entregará el paquete) y que esta posición real se compara con datos históricos de entregas. Este método no es un algoritmo que se pueda implementar en producción, pero permite definir un rendimiento máximo independiente del método para identificar Clientes. Para aplicarlo se realizó lo siguiente.

Primero se procedió a cargar en el DataSet llamado "**df_Acumulado**", un mes de datos, comenzando con enero del año 2020. En este DataSet cada nuevo mes que se procesa, se va acumulando al mes anterior. Luego se procedió a cargar en el DataSet llamado "**df_Mensual**", una muestra aleatoria de 300 registros, extraídos del mes a procesar. Con una rutina desarrollada en lenguaje Python usando librerías Pandas, se comparó registro a registro, en donde se fueron almacenando las coincidencias que encontraba al aplicar el **Cálculo de Distancia Haversine,** (Ver Anexo 2), entre los puntos geográficos de las entregas de los Clientes. Para dicho cálculo se utilizaron los campos de "**Latitud**" y "**Longitud**". Se definió como rango valido la distancia "**Menor o Igual a 10 Metros**".

En la Figura 38, se aprecia un diagrama de flujo con el detalle del paso a paso del método del "**Algoritmo Oráculo"**, (Ver Anexo 3), y todas las condiciones definidas.

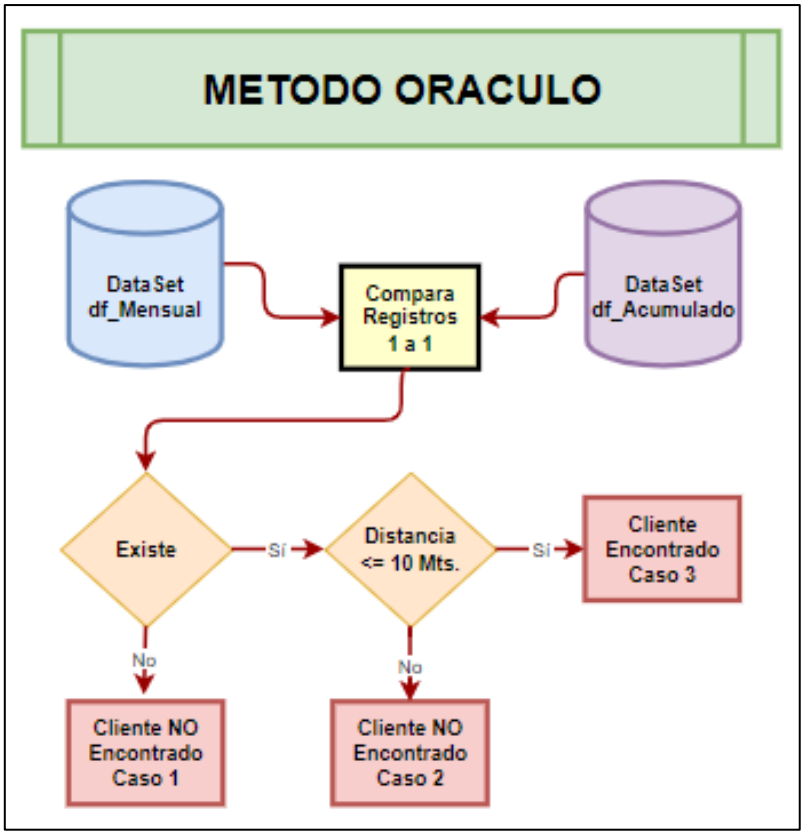


*Figura 38: Diagrama de Flujo Usando el Método del Algoritmo Oráculo.*

En este proceso existen 3 alternativas de término, las cuales son

- **Caso 1 (negro)**. Según las coordenadas del Cliente, al buscar en la historia, no hay coincidencias, ya que nunca se ha llegado a esa dirección.

- **Caso 2 (azul)**. Según las coordenadas del Cliente, al buscar en la historia, hay coincidencias, pero está fuera del rango de los 10 Metros.

- **Caso 3 (rojo)**. Según las coordenadas del Cliente, al buscar en la historia, hay coincidencias y está dentro del rango de los 10 Metros.

En la Figura 39, se aprecian en un mapa los casos definidos anteriormente para el método del "**Algoritmo Oráculo"**.


*Figura 39: Clasificación Usando el Método del Algoritmo Oráculo.*

En la Tabla 17, se aprecia una muestra de los registros procesados con el método del "**Algoritmo Oráculo"**.

```
Registros: 475  - Coinciden: 267 7500916 7663319  - Haversine: 1.6964882766406273
Registros: 476  - Coinciden: 268 20754374 7664391  - Haversine: 3.9857821381073557
Registros: 477  - Coinciden: 269 7678641 7665332  - Haversine: 3.0789036075383533
Registros: 478  - Coinciden: 270 7483661 7665591  - Haversine: 6.305045508042532
Registros: 481  - Coinciden: 271 23701646 7667965  - Haversine: 6.883801078972756
Registros: 483  - Coinciden: 272 7653760 7668399  - Haversine: 0.0
Registros: 484  - Coinciden: 273 7449126 7668969  - Haversine: 0.4213728729743275
Registros: 487  - Coinciden: 274 7461808 7445736  - Haversine: 3.5507017000262264
Registros: 488  - Coinciden: 275 7446263 7446204  - Haversine: 4.662368372768691
```

*Tabla 17: Muestra de Datos Usando el Método del Algoritmo Oráculo*

### 3.4.2. Algoritmo JOIN – (Cota Inferior)

En este método se definió la **Cota Inferior** (Forma actual de cómo se trabaja en la Empresa) basado en la comparación exacta de cadenas de texto que se comparan con datos históricos. Para aplicarlo se realizó lo siguiente:

Primero se procedió a cargar en el DataSet llamado "**df_Acumulado**", un mes de datos, comenzando con enero del año 2020. En este DataSet cada nuevo mes que se procesa, se va acumulando al mes anterior. Luego se procedió a cargar en el DataSet llamado "**df_Mensual**", una muestra aleatoria de **300** registros, extraídos del mes a procesar. Con una rutina desarrollada en lenguaje Python usando librerías Pandas, se comparó registro a registro, en donde se fueron almacenando las coincidencias que encontraba al aplicar lo siguiente:

- **Condición de Igualdad**. Entre los campos "**Nombre**" de los Clientes.

- **Cálculo de Distancia Haversine.** Entre los puntos geográficos de las entregas de los Clientes, para lo cual se utilizaron los campos de "**Latitud**" y "**Longitud**". Se definió como rango valido la distancia "**Menor o Igual a 10 Metros**". (Ver Anexo 2).

En la Figura 40, se aprecia un diagrama de flujo con el detalle del paso a paso del método del "**Algoritmo Join"**, (Ver Anexo 4), y todas las condiciones definidas.

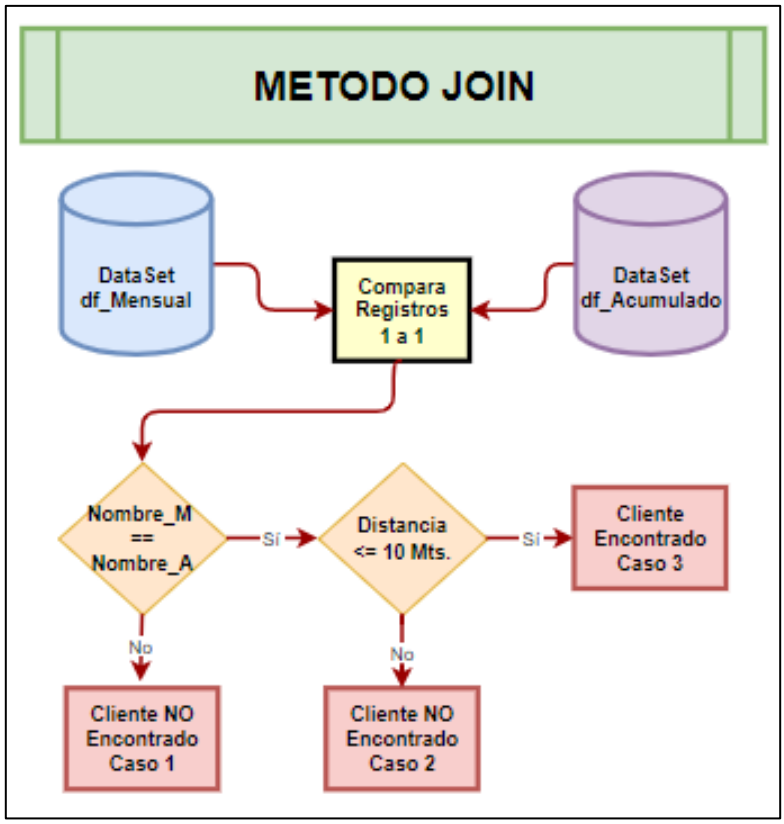


*Figura 40: Diagrama de Flujo Usando el Método del Algoritmo Join.*

En este proceso existen 3 alternativas de término, las cuales son

- **Caso 1 (negro)**. Según el **Nombre** del Cliente, al buscar en la historia, no hay coincidencias, ya que nunca se ha llegado a esa dirección.

- **Caso 2 (azul)**. Según el **Nombre** y las coordenadas del Cliente, al buscar en la historia, hay coincidencias, pero está fuera del rango de los 10 Metros.

- **Caso 3 (rojo)**. Según el **Nombre** y las coordenadas del Cliente, al buscar en la historia, hay coincidencias y está dentro del rango de los 10 Metros.

En la Figura 41, se aprecian en un mapa los casos definidos anteriormente para el método del "**Algoritmo Join"**.


*Figura 41: Clasificación Usando el Método del Algoritmo Join.*

En la Tabla 18, se aprecia una muestra de los registros procesados con el método del "**Algoritmo Join"**.

| | | | | | | |
|---|---|---|---|---|---|---|
| - | 2180 | 2539594 | JUAN PEREZ COTAPOS | jp.cotaposxxx@yimeil.cl | -36.615493 | -72.958180 | |
| | | 573702 | JUAN PEREZ COTAPOS | jp.cotaposxxx@yimeil.cl | -36.832046 | -73.055018 | Haversine = 1.142 |
| | | 9230714 | JUAN PEREZ COTAPOS | jp.cotaposxxx@yimeil.cl | -36.805556 | -73.064507 | Haversine = 2.237 |
| | | 9231524 | JUAN PEREZ COTAPOS | jp.cotaposxxx@yimeil.cl | -36.829968 | -73.116857 | Haversine = 5.556 |
| | | 3747013 | JUAN PEREZ COTAPOS | jp.cotaposxxx@yimeil.cl | -36.828956 | -73.044183 | Haversine = 6.319 |

*Tabla 18: Muestra de Datos Usando el Método del Algoritmo Join.*

### 3.4.3. Algoritmo Propuesto Basado en Aprendizaje de Maquinas

En esta sección se realizó la identificación y clasificación de Clientes basado en métodos de clasificación usando aprendizaje de máquinas, para poder solucionar los tipos de problemas analizados en la sección anterior y que tiene relación con tokens en desorden, tokens faltantes, abreviación y/o truncamiento de tokens en el campo "**Nombre**".

Primeramente, se realizó un análisis exhaustivo para determinar el Clasificador que nos entregara los resultados más óptimos para nuestro caso. Con el objetivo de explorar diferentes opciones y considerando la simplicidad de la implementación, tomamos la decisión de emplear el algoritmo de ***Random Forest*** (Bosques Aleatorios), el cual se caracteriza por su alta precisión en la clasificación y su capacidad de mantener esta precisión incluso ante volúmenes extensos de datos. Esta elección resulta apropiada dado que, en función de la distribución de nuestros datos, es improbable que se establezca una relación lineal directa Figura 43.

Como segundo paso, posterior al análisis de los datos y las características que se ocuparían, se realizó el borrado de las columnas no numéricas y las que no servían. Después, se generaron múltiples muestras de largo **k** provenientes del universo de datos, así como también las variables de estas y se dividió el DataSet en un conjunto de entrenamiento y un conjunto de prueba

- **Train set**. Es el set de datos para el aprendizaje, el cual se entrena mediante el método de aprendizaje supervisado y es el **80%** de la data del DataSet.

- **Test set**. Se emplea para evaluar el modelo final, ya entrenado y ajustado y corresponde al **20%** de la data del DataSet.

En la Figura 42, se muestra distribución del DataSet, en donde se grafican los porcentajes de Train y Test.

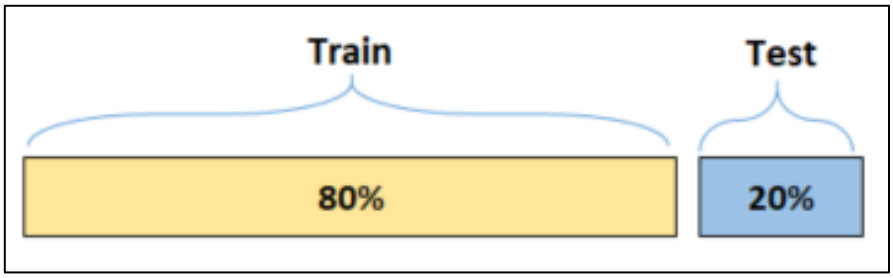


*Figura 42: Esquema de Separación del DataSet.*

En la Tabla 19, se muestra un ejemplo de la data que está contenida en el DataSet.

| Id_Bcd | Nombre 1 | Nombre 2 | Levenshtein | Coseno | Fuzzy | Jaro | Score |
|---|---|---|---|---|---|---|---|
| 3917515 | PEDRO GODOY | JUAN GONZALEZ | 14.0 | 0.0 | 34.0 | 0.51 | 0 |
| 8287545 | LUIS SOTO | JUAN GONZALEZ | 13.0 | 0.0 | 28.0 | 0.51 | 0 |
| 3422356 | MARIA DIAZ | JUAN GONZALEZ | 10.0 | 0.0 | 38.0 | 0.62 | 0 |
| 12705486 | JUANA RIVAS | JUAN GONZALEZ | 8.0 | 0.0 | 53.0 | 0.61 | 0 |
| 5795725 | ALEXIS SANCHEZ | JUAN GONZALEZ | 12.0 | 0.0 | 21.0 | 0.57 | 0 |

*Tabla 19: Muestra de Datos del DataSet.*

Luego se procedió a cargar en el DataSet llamado "df_Mensual", extrayendo una muestra aleatoria de **300** registros, por cada mes del universo de datos. Y se realizaron las predicciones y los resultados fueron almacenados para su posterior análisis.

A graficar un diagrama de dispersión o *scatter plots*, de las características de los registros. Se aprecia mejor la distribución de la data e identifica visualmente la posible correlación entre las dos variables. Además de poder determinar si existe un desbalanceo entre ellas. Esto se aprecia en la Figura 43.

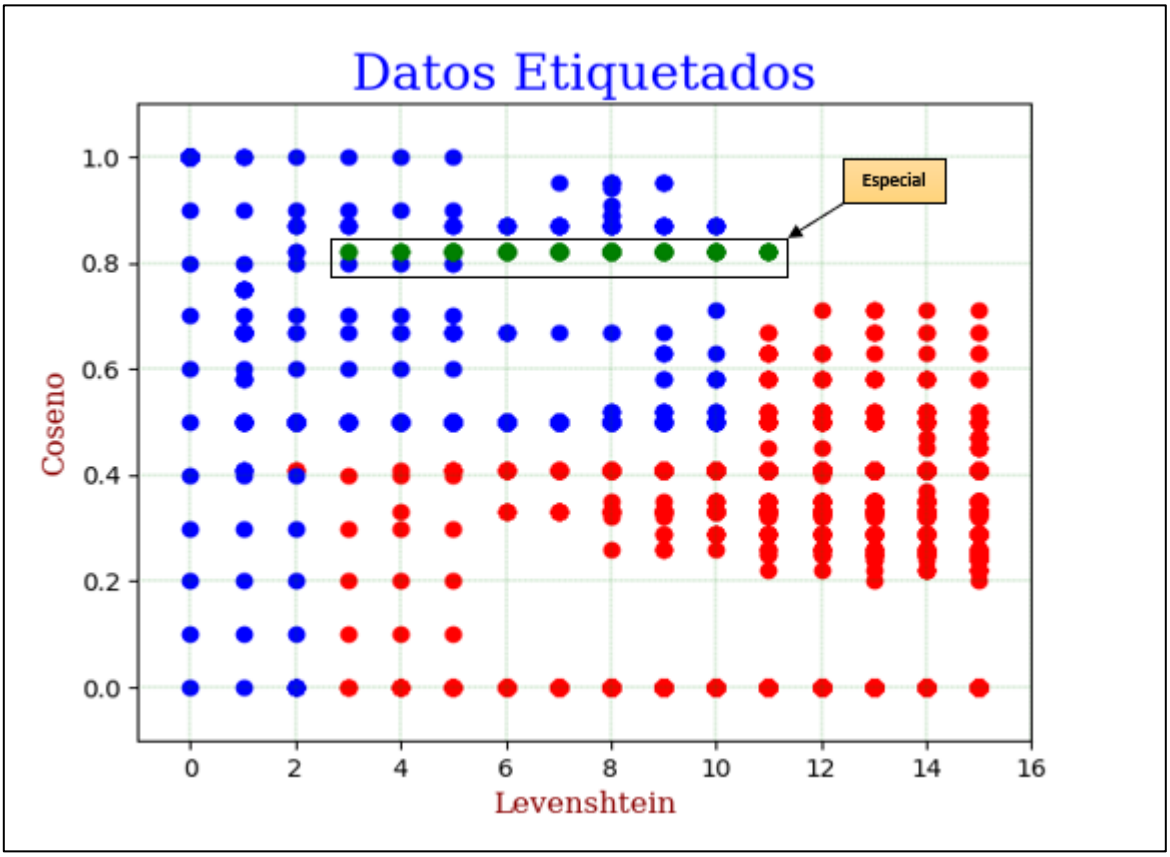


*Figura 43: Distribución del DataSet con las Características Levenshtein y Coseno.*
*Entre los nombres de clientes. Azul indica 2 nombres que son efectivamente el mismo cliente. Rojo indica casos en que dos nombres no son el mismo cliente. Verde indica casos en que en algunas ocasiones son el mismo cliente.*

Los puntos destacados de color verde, que llamaremos "**Especial**", es una clasificación que resultó ambigua, los cuales no fueron considerados en el proceso de entrenamiento del modelo del Clasificador.

En los gráficos de correlación entre las variables de la Figura 44, se aprecia que no es un problema simple que se pudiera manejar con una variable, ya que se solapan entre ellas.

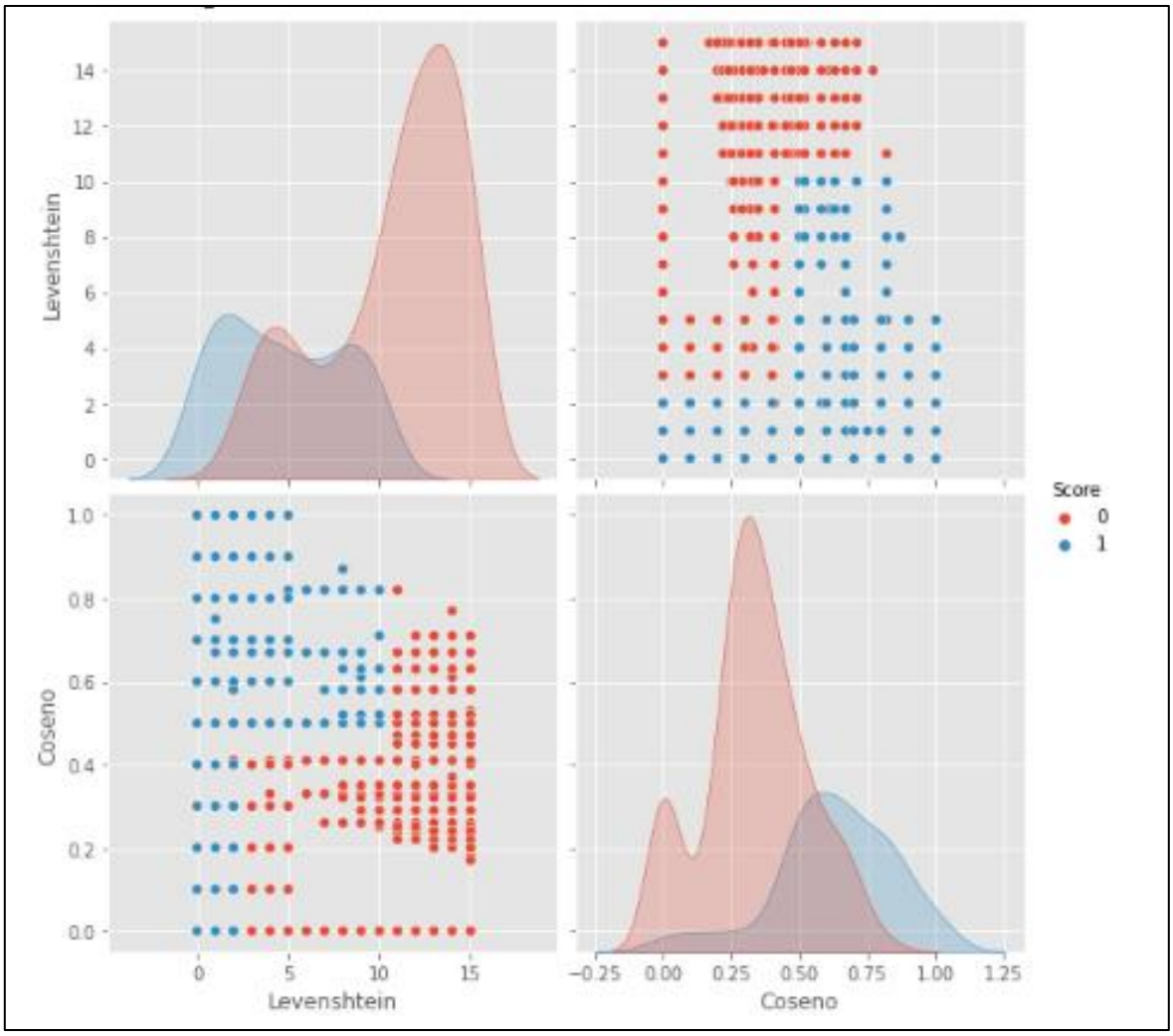


*Figura 44: Correlación de Pearson de las Características Levenshtein y Coseno.*

# 4. Resultados y Análisis

## 4.1. Proceso de Evaluación

En esta sección los procesos de evaluación se realizaron validando los diferentes métodos descritos en el capítulo anterior, los cuales se basaron en el siguiente universo de datos:

- Sistema = **Paquetería Internacional**
- Año = **2020**
- Comuna = **Concepción**
- Estado = **Entregado**
- Total = **250 mil (aproximado)**

Obteniendo un **Universo Real** de aproximadamente "**239 mil envíos**". En la Tabla 20, se muestra el detalle de las entregas por mes.

| CONCEPCION - Año = 2020 | | | | |
|---|---|---|---|---|
| Mes | Universo Total | Universo Real | Acumulado | Muestra |
| Ene | 17.099 | 16.418 | 16.418 | 300 |
| Feb | 8.966 | 8.362 | 24.780 | 300 |
| Mar | 11.805 | 11.651 | 36.431 | 300 |
| Abr | 9.696 | 9.061 | 45.492 | 300 |
| May | 17.743 | 17.743 | 63.235 | 300 |
| Jun | 17.867 | 16.734 | 79.969 | 300 |
| Jul | 28.720 | 26.553 | 106.522 | 300 |
| Ago | 21.712 | 20.334 | 126.856 | 300 |
| Sep | 23.397 | 20.931 | 147.787 | 300 |
| Oct | 32.680 | 32.345 | 180.132 | 300 |
| Nov | 36.574 | 34.914 | 215.046 | 300 |
| Dic | 24.393 | 23.922 | 238.968 | 300 |
| | 250.652 | 238.968 | | |

*Tabla 20: Universo de Datos Clasificados por Mes.*

## 4.2. Resultados Obtenidos Basado en Métodos de Distancia

En esta sección se detallan los resultados obtenidos basado en métodos de distancia para los Tipos de Problemas descritos en la Tabla 21.

| Tipos | Problemas | Ejemplo Campo "Nombre" | % |
|---|---|---|---|
| 4 | Errores ortográficos y tipográficos | Juan Alejandro Zaragoza Vahamondes | 9% |
| 5 | Espacios en blanco: Eliminación o adición de espacios y/o caracteres especiales | Juan Alejandro Zaragoza#Bahamondes | 12% |
| 6 | Préfijos/Sufíjos sin valor semántico: Presencia de caracteres al inicio y/o al final | Dr. Juan Alejandro Zaragoza Bahamondes , UCH | 3% |

*Tabla 21: Tipos de Problemas.*

### 4.2.1. Cálculo de Distancia de Levenshtein

Al aplicar el Cálculo de Distancia de Levenshtein para los tipos de problemas indicados anteriormente, esta medida no cumplió con lo esperado.

El rango válido que se aplicó para los resultados del cálculo fue: "**Mayor o Igual que 0 y Menor o Igual a 2**".

Para los tres tipos de problemas, ya sea, "Errores Ortográficos y Tipográficos", "Espacios en Blanco o Caracteres Especiales" y "Prefijos y Sufijos", se realizaron pruebas con data extraída aleatoriamente y su resultado fue dispar en cada tipo de problemas. Según se aprecia en la Tabla 22.

| Tipo | Nombre1 | Nombre2 | Levenshtein |
|---|---|---|---|
| 4 | FRANCISCA MUÑOZ GONZALEZ | FRANCESCA MUNOZ GONZALEZ | 3 |
| | VANESSA CUITIÑO | VANESSA CUITI#O | 3 |
| 5 | CATALINA ALEJANDRA HERNANDEZ FERNANDEZ | CATALINA ALEJANDRA HERNÄ´NDEZ FERNÄ´NDEZ | 4 |
| | BETTY ARRIAGADA TOLEDO | BETTY ARRIAG ADA T OLEDO | 8 |
| 6 | CATALINA ILLANES HERRERA | CATALINA ILLANES HERRERA 167-BLACKWHITE | 15 |
| | CRISTIAN BELLO CRUCES | MR CRISTIAN BELLO CRUCES | 3 |

*Tabla 22: Resultados Prueba 1 del Cálculo de la Distancia Levenshtein.*

Con los resultados obtenidos en las primeras pruebas, se decidió aplicar técnicas de limpieza avanzada al campo Nombre, usando **Expresiones Regulares**, para corregir problemas de ortografía, tipografía, caracteres especiales, espacios en blanco, prefijos y sufijos. Y que esto pudiera ayudar a las siguientes pruebas.

Nuevamente se realizaron pruebas y se pudieron obtener resultados un poco mejor, pero no lo suficiente para darlos como resultados finales. Como se aprecia en la Tabla 23.

| Tipo | Nombre1 | Nombre2 | Levenshtein |
|---|---|---|---|
| 4 | YERKO BASTIAN MIRANDA MARQUEZ | YERKO BASTIAN MITANDA MARQUEZ | 1 |
| 5 | JUAN PEREZ COTAPO | JUAN#PEREZ COTAPO | 2 |
| 6 | ALBERTO GOMEZ OLIVARES | S ALBERTO GOMEZ OLIVARES | 2 |

*Tabla 23: Resultados Prueba 2 del Cálculo de la Distancia Levenshtein.*

En un análisis aleatorio de los resultados obtenidos, se detectó que muchos registros no estaban siendo considerados dentro de las validaciones, por lo que se amplió el rango de 2 a 5, quedando como: "**Mayor o Igual que 0 y Menor o Igual a 5**".

Al ampliar el rango, si bien es cierto sirvió para el propósito de considerar Clientes que eran excluidos en pruebas anteriores. Se sumó el problema que empezaron a ser considerados Clientes totalmente distintos unos con otros. Como se aprecia en la Tabla 24.

| Nombre1 | Nombre2 | Levenshtein |
|---|---|---|
| CAMILA MUÑOZ OVIEDO | CAMILA MUNOZ VIVERO | 4 |
| SYLVIA GONZALES PINTO | MARIA GONZALEZ PINTO | 5 |
| ADRIANA MORENO ROJO | ADRIANA ROMERO ROJAS | 5 |

*Tabla 24: Resultados Prueba 3 del Cálculo de la Distancia Levenshtein.*

Para concluir, este método de Cálculo de Distancia de Levenshtein, es muy débil y poco confiable al actuar por sí solo, se necesita contar como complemento con otro método de Cálculo de Distancia, y así poder brindar mejores resultados.

### 4.2.2. Cálculo de Similitud de Coseno

Al aplicar el Cálculo de Similitud de Coseno para los Tipos de Problemas indicados anteriormente, resulto mejor que el Cálculo de Distancia de Levenshtein.

El rango válido que se aplicó para los resultados del cálculo fue: "**Mayor o Igual que 0,5 y Menor o Igual a 1**".

Se realizaron las mismas pruebas para los tres tipos de problemas descritos anteriormente. Según se aprecia en la Tabla 25.

| Tipo | Nombre1 | Nombre2 | Levenshtein | Coseno |
|---|---|---|---|---|
| 4 | FRANCISCA MUÑOZ GONZALEZ | FRANCESCA MUNOZ GONZALEZ | 3 | 0,00 |
| | VANESSA CUITIÑO | VANESSA CUITI#O | 3 | 0,00 |
| 5 | CATALINA ALEJANDRA HERNANDEZ FERNANDEZ | CATALINA ALEJANDRA HERNÄ´NDEZ FERNÄ´NDEZ | 4 | 0,35 |
| | BETTY ARRIAGADA TOLEDO | BETTY ARRIAG ADA T OLEDO | 8 | 0,25 |
| 6 | CATALINA ILLANES HERRERA | CATALINA ILLANES HERRERA 167-BLACKWHITE | 15 | 0,77 |
| | CRISTIAN BELLO CRUCES | MR CRISTIAN BELLO CRUCES | 3 | 0,86 |

*Tabla 25: Resultados Prueba 1 del Cálculo de la Similitud Coseno.*

Al comparar los resultados entre los Cálculos de Distancia de Levenshtein y de Similitud de Coseno, se dan los siguientes resultados:

En los tipos 4 y 5 ambos arrojan resultados indicando que no coinciden los Cálculos de Distancia y Similitud. En cambio, en el tipo 6, la Similitud de Coseno si arroja resultados positivos, a pesar de estar "sucia" la cadena de texto del campo Nombre.

Se aplican las técnicas de limpieza avanzada al campo Nombre, usando **Expresiones Regulares**, para corregir problemas de ortografía, tipografía, caracteres especiales, espacios en blanco, prefijos y sufijos. Nuevamente se realizaron pruebas y los resultados en ambos métodos dan resultados positivos, como se aprecia en la Tabla 26.

| Tipo | Nombre1 | Nombre2 | Levenshtein | Coseno |
|---|---|---|---|---|
| 4 | YERKO BASTIAN MIRANDA MARQUEZ | YERKO BASTIAN MITANDA MARQUEZ | 1 | 0,75 |
| 5 | JUAN PEREZ COTAPO | JUAN#PEREZ COTAPO | 2 | 0,66 |
| 6 | ALBERTO GOMEZ OLIVARES | S ALBERTO GOMEZ OLIVARES | 2 | 0,86 |

*Tabla 26: Resultados Prueba 2 del Cálculo de la Similitud Coseno.*

Se realizan ajustes a los rangos y se vuelven a realizar pruebas y en esta ocasión el Cálculo de Similitud de Coseno entrega valores de No coincidencia, ya que aquí los Cliente son diferentes. Como se aprecia en la Tabla 27.

| Nombre1 | Nombre2 | Levenshtein | Coseno |
|---|---|---|---|
| CAMILA MUÑOZ OVIEDO | CAMILA MUNOZ VIVERO | 4 | 0,33 |
| SYLVIA GONZALES PINTO | MARIA GONZALEZ PINTO | 5 | 0,33 |
| ADRIANA MORENO ROJO | ADRIANA ROMERO ROJAS | 5 | 0,33 |

*Tabla 27: Resultados Prueba 3 del Cálculo de la Similitud Coseno.*

### 4.2.3. Cálculos Combinados de Distancia y Similitud

Después de realizar pruebas por separado con las distancias anteriores, se realizaron pruebas con ambos métodos en forma conjunta.

Se aplicaron los mismos rangos válidos que se utilizaron anteriormente, es decir:

- Distancia de Levenshtein = "**Mayor o Igual que 0 y Menor o Igual a 5**".
- Similitud de Coseno = "**Mayor o Igual que 0,5 y Menor o Igual a 1**".

Estas pruebas consistieron en procesar mes a mes todo el universo de datos ya preparado anteriormente, tomando una muestra aleatoria de **300** registros, extraídos del mes a procesar.

Estas pruebas entregaron resultados que fluctuaron entre **28%** y **61**%. Según lo observado en la Tabla 28.

| CONCEPCION - Año = 2020 | | | | | | |
|---|---|---|---|---|---|---|
| Mes | Universo Total | Universo Real | Acumulado | Muestra | Cálculo Distancia | |
| Ene | 17.099 | 16.418 | 16.418 | 300 | 83 | 28% |
| Feb | 8.966 | 8.362 | 24.780 | 300 | 93 | 31% |
| Mar | 11.805 | 11.651 | 36.431 | 300 | 107 | 36% |
| Abr | 9.696 | 9.061 | 45.492 | 300 | 123 | 41% |
| May | 17.743 | 17.743 | 63.235 | 300 | 135 | 45% |
| Jun | 17.867 | 16.734 | 79.969 | 300 | 142 | 47% |
| Jul | 28.720 | 26.553 | 106.522 | 300 | 143 | 48% |
| Ago | 21.712 | 20.334 | 126.856 | 300 | 161 | 54% |
| Sep | 23.397 | 20.931 | 147.787 | 300 | 176 | 59% |
| Oct | 32.680 | 32.345 | 180.132 | 300 | 185 | 62% |
| Nov | 36.574 | 34.914 | 215.046 | 300 | 179 | 60% |
| Dic | 24.393 | 23.922 | 238.968 | 300 | 183 | 61% |
| | 250.652 | 238.968 | | | | |

*Tabla 28: Resultados por Mes Usando Cálculos de Distancia*

En la Figura 45, se aprecia el gráfico del "**Cálculo de Distancia**".

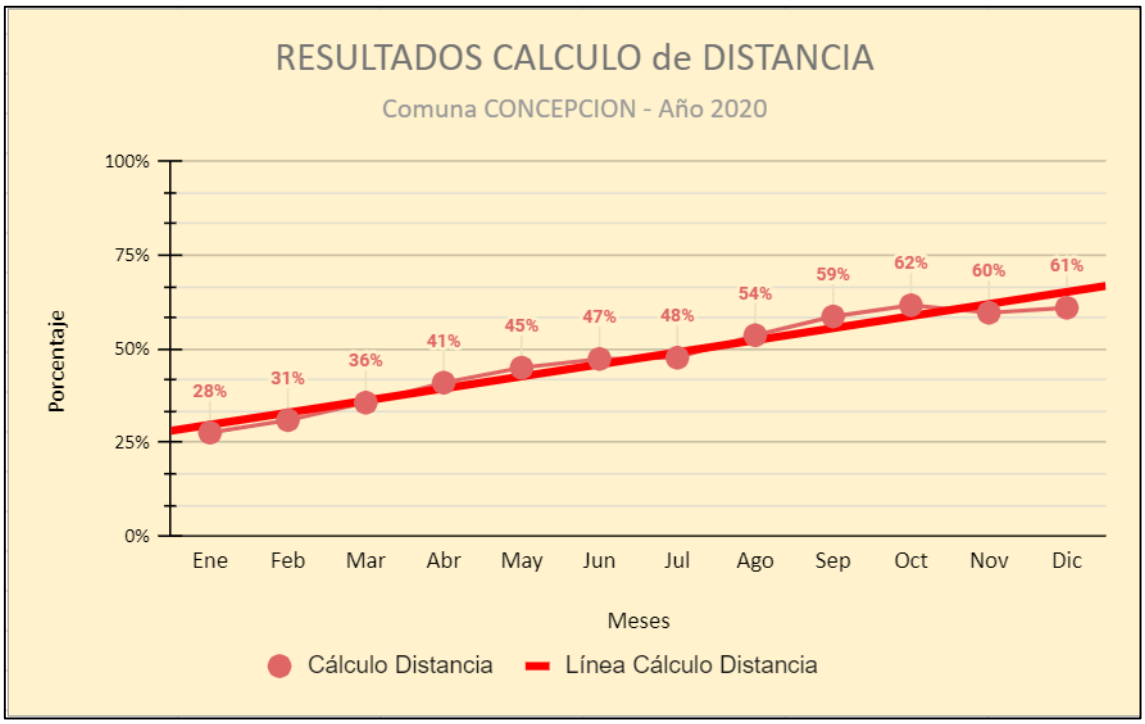


*Figura 45: Gráfico Usando Cálculo de Distancia.*

## 4.3. Resultados Obtenidos Basado en Algoritmos Propuestos

En esta sección se detallan los resultados obtenidos basado en algoritmos propuestos para los Tipos de Problemas mostrados en la Tabla 29.

| Tipos | Problemas | Ejemplo Campo "Nombre" | % |
|---|---|---|---|
| 1 | Tokens en desorden | Zaragoza Bahamondes Juan Alejandro | 1% |
| 2 | Tokens Faltantes: Eliminación de uno o más tokens | Juan Zaragoza | 32% |
| 3 | Abreviaturas: Truncamineto de uno o más tokens | Juan A. Zaragoza Baham | 4% |

*Tabla 29: Tipos de Problemas.*

### 4.3.1. Algoritmo Oráculo – (Cota Superior)

Para el método del "**Algoritmo Oráculo"**, las coincidencias fueron creciendo a medida que se procesaban más meses y los resultados fluctuaron entre **71%** y **92%**. Según lo observado en la Tabla 30.

| CONCEPCION - Año = 2020 | | | | | | |
|---|---|---|---|---|---|---|
| Mes | Universo Total | Universo Real | Acumulado | Muestra | ORACULO | |
| Ene | 17.099 | 16.418 | 16.418 | 300 | 212 | 71% |
| Feb | 8.966 | 8.362 | 24.780 | 300 | 215 | 72% |
| Mar | 11.805 | 11.651 | 36.431 | 300 | 232 | 77% |
| Abr | 9.696 | 9.061 | 45.492 | 300 | 241 | 80% |
| May | 17.743 | 17.743 | 63.235 | 300 | 241 | 80% |
| Jun | 17.867 | 16.734 | 79.969 | 300 | 239 | 80% |
| Jul | 28.720 | 26.553 | 106.522 | 300 | 260 | 87% |
| Ago | 21.712 | 20.334 | 126.856 | 300 | 265 | 88% |
| Sep | 23.397 | 20.931 | 147.787 | 300 | 273 | 91% |
| Oct | 32.680 | 32.345 | 180.132 | 300 | 279 | 93% |
| Nov | 36.574 | 34.914 | 215.046 | 300 | 276 | 92% |
| Dic | 24.393 | 23.922 | 238.968 | 300 | 277 | 92% |
| | 250.652 | 238.968 | | | | |

*Tabla 30: Resultados por Mes Usando el Algoritmo Oráculo.*

Al aplicar este proceso para todo el año, se definió que todo proceso que esté por encima de esta línea se considerará erróneo o no válido. En la Figura 46, se aprecia el gráfico de la **Cota Superior** usando el método del "**Algoritmo Oráculo**".

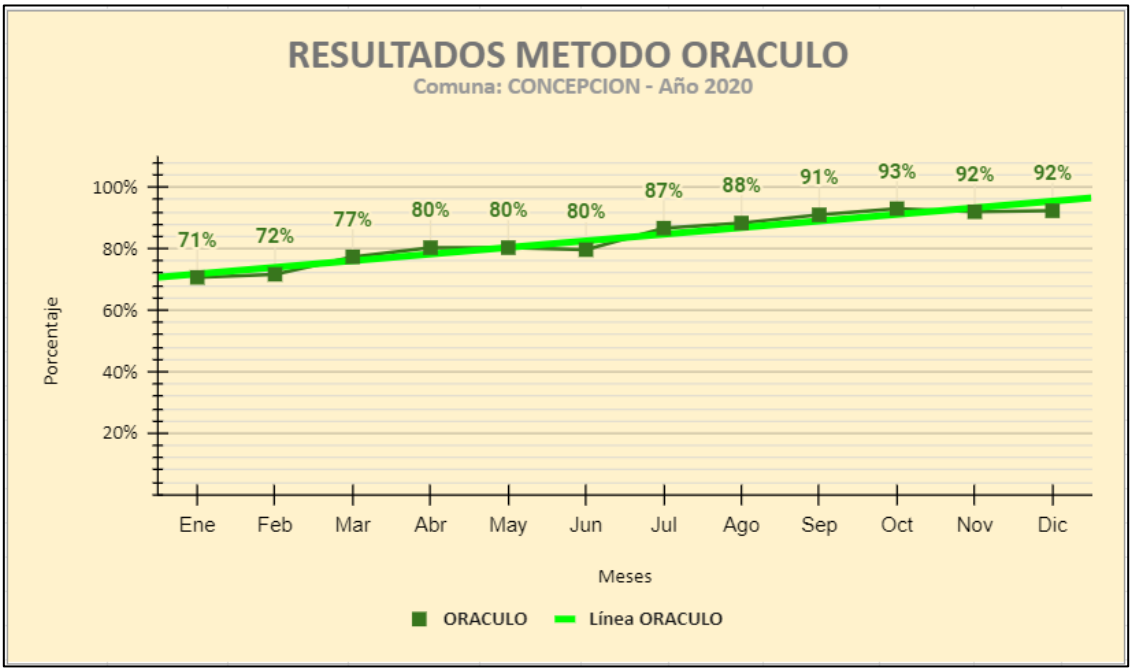


*Figura 46: Gráfico de Cota Superior Usando el Algoritmo Oráculo.*

## 4.3.2. Algoritmo JOIN – (Cota Inferior)

Para el método del "**Algoritmo Join**", las coincidencias también fueron creciendo a medida que se procesaban más meses y los resultados fluctuaron entre **27%** y **60%**. Según lo observado en la Tabla 31.

| CONCEPCION - Año = 2020 | | | | | | |
|---|---|---|---|---|---|---|
| **Mes** | **Universo Total** | **Universo Real** | **Acumulado** | **Muestra** | **JOIN** | |
| **Ene** | **17.099** | **16.418** | **16.418** | 300 | 82 | **27%** |
| **Feb** | **8.966** | **8.362** | **24.780** | 300 | 89 | **30%** |
| **Mar** | **11.805** | **11.651** | **36.431** | 300 | 102 | **34%** |
| **Abr** | **9.696** | **9.061** | **45.492** | 300 | 121 | **40%** |
| **May** | **17.743** | **17.743** | **63.235** | 300 | 135 | **45%** |
| **Jun** | **17.867** | **16.734** | **79.969** | 300 | 137 | **46%** |
| **Jul** | **28.720** | **26.553** | **106.522** | 300 | 136 | **45%** |
| **Ago** | **21.712** | **20.334** | **126.856** | 300 | 160 | **53%** |
| **Sep** | **23.397** | **20.931** | **147.787** | 300 | 173 | **58%** |
| **Oct** | **32.680** | **32.345** | **180.132** | 300 | 181 | **60%** |
| **Nov** | **36.574** | **34.914** | **215.046** | 300 | 177 | **59%** |
| **Dic** | **24.393** | **23.922** | **238.968** | 300 | 181 | **60%** |
| | **250.652** | **238.968** | | | | |

*Tabla 31: Resultados por Mes Usando el Algoritmo Join.*

Al aplicar este proceso para todo el año, se definió que todo proceso que esté por debajo de esta línea se considerará erróneo o no válido. En la Figura 47, se aprecia el gráfico de la **Cota Inferior** usando el método del "**Algoritmo Join".**

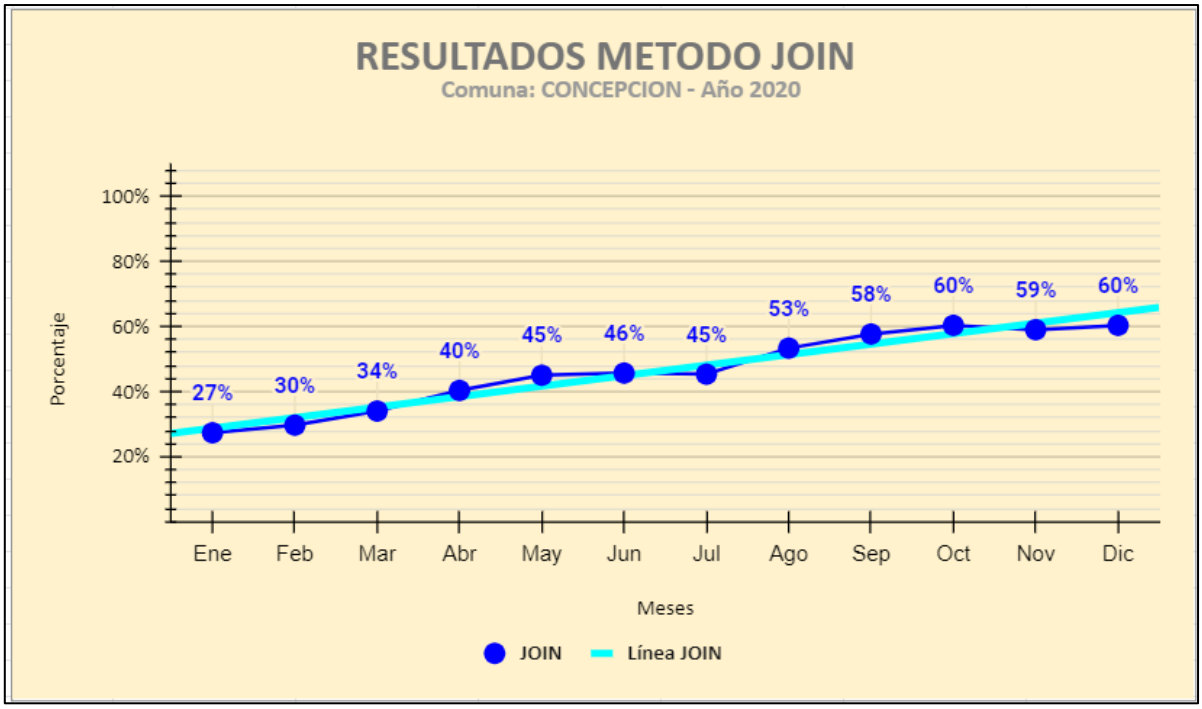


*Figura 47: Gráfico de Cota Inferior Usando el Algoritmo Join.*

### 4.3.3. Algoritmo Basado en Aprendizaje de Maquinas

Para el método del "**Algoritmo Clasificador**" usando Aprendizaje de Maquinas, las coincidencias también fueron creciendo a medida que se procesaban más meses y los resultados fluctuaron entre **39%** y **84%**. Según lo observado en la Tabla 32.

| CONCEPCION - Año = 2020 | | | | | | |
|---|---|---|---|---|---|---|
| Mes | Universo Total | Universo Real | Acumulado | Muestra | RFC | |
| Ene | 17.099 | 16.418 | 16.418 | 300 | 116 | 39% |
| Feb | 8.966 | 8.362 | 24.780 | 300 | 141 | 47% |
| Mar | 11.805 | 11.651 | 36.431 | 300 | 159 | 53% |
| Abr | 9.696 | 9.061 | 45.492 | 300 | 191 | 64% |
| May | 17.743 | 17.743 | 63.235 | 300 | 202 | 67% |
| Jun | 17.867 | 16.734 | 79.969 | 300 | 206 | 69% |
| Jul | 28.720 | 26.553 | 106.522 | 300 | 221 | 74% |
| Ago | 21.712 | 20.334 | 126.856 | 300 | 231 | 77% |
| Sep | 23.397 | 20.931 | 147.787 | 300 | 245 | 82% |
| Oct | 32.680 | 32.345 | 180.132 | 300 | 232 | 77% |
| Nov | 36.574 | 34.914 | 215.046 | 300 | 231 | 77% |
| Dic | 24.393 | 23.922 | 238.968 | 300 | 252 | 84% |
| | 250.652 | 238.968 | | | | |

*Tabla 32: Resultados por Mes Usando el Algoritmo Clasificador.*

Al aplicar este proceso para todo el año, se obtuvo un resultado bastante bueno en bases a los otros métodos aplicados. En la Figura 48, se aprecia el gráfico de los resultados usando el método del "**Algoritmo Clasificador**".

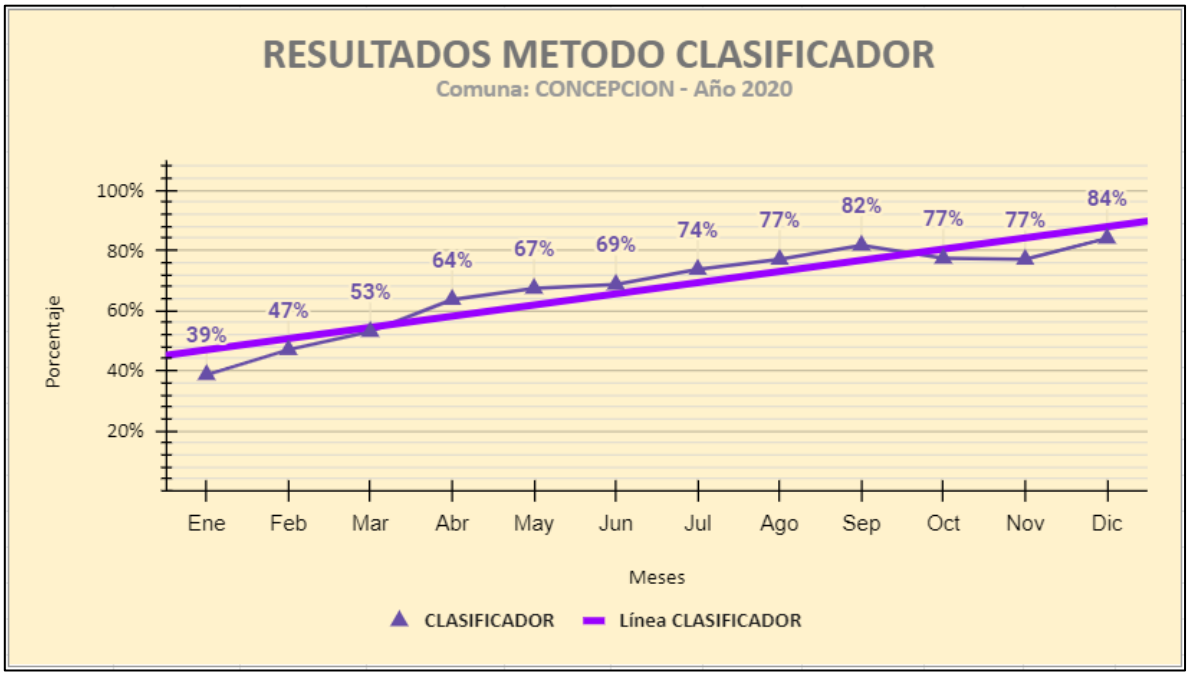


*Figura 48: Gráfico del Resultados Usando el Algoritmo Clasificador.*

Entrenando el modelo de clasificación como **Random Forest** para el mes de diciembre del año 2020, se testeó su desempeño para poder predecir la clasificación de las etiquetas.

En la Figura 49, se presenta la **Curva ROC**, en la cual se evidencia que el **AUC** entrega un valor de **0,948**.

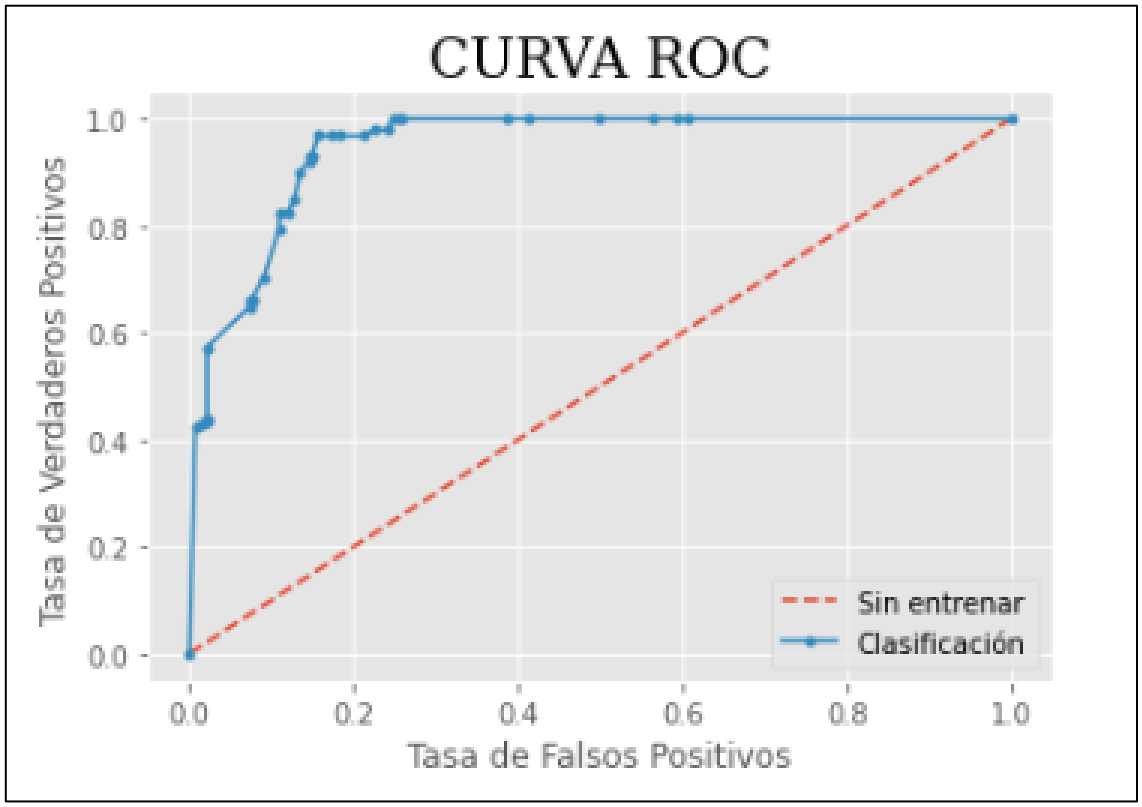


*Figura 49: Curva ROC.*

En la Figura 50, se presenta en forma gráfica la Matriz de Confusión, la cual obtiene las diferentes métricas propias.

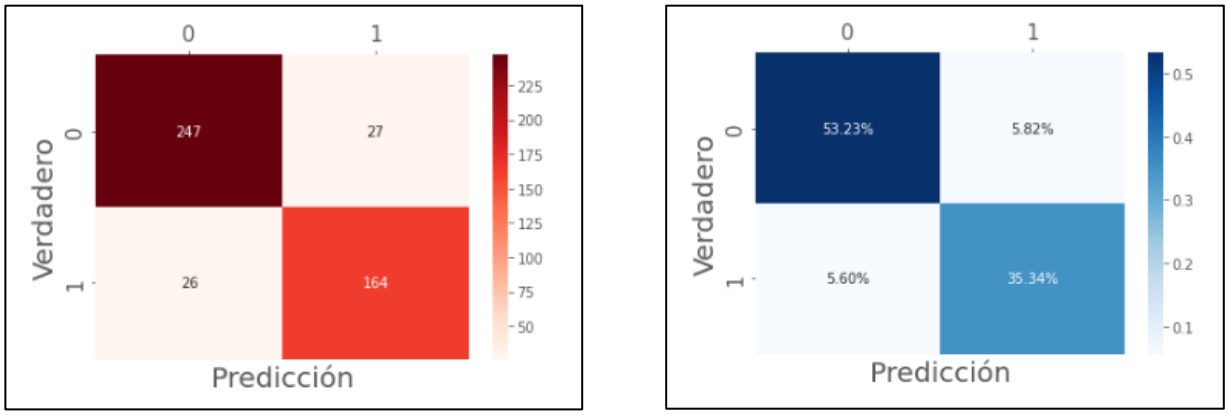


*Figura 50: Gráfico de la Matriz de Confusión.*

A partir de la Matriz de Confusión anterior, se genera la Tabla 33, que indica las principales métricas de desempeño del modelo.

| Accuracy | Recall | Precision | Specificity | F1 | Tasa Falsos Negativos | Valor Predictivo Positivo | Valor Predictivo Negativo |
|---|---|---|---|---|---|---|---|
| Exactitud | Sensibilidad | Precisión | Especificidad | | | | |
| 0,89 | 0,90 | 0,90 | 0,86 | 0,90 | 0,10 | 0,90 | 0,86 |

*Tabla 33: Métricas de Desempeño del Modelo.*

De los resultados obtenidos presentados, se evidencia que la **Exactitud** del modelo para la clasificación obtuvo un valor de **0,89**. Se realiza una comparación entre los resultados finales de los tres modelos, como se aprecia en la Figura 51.

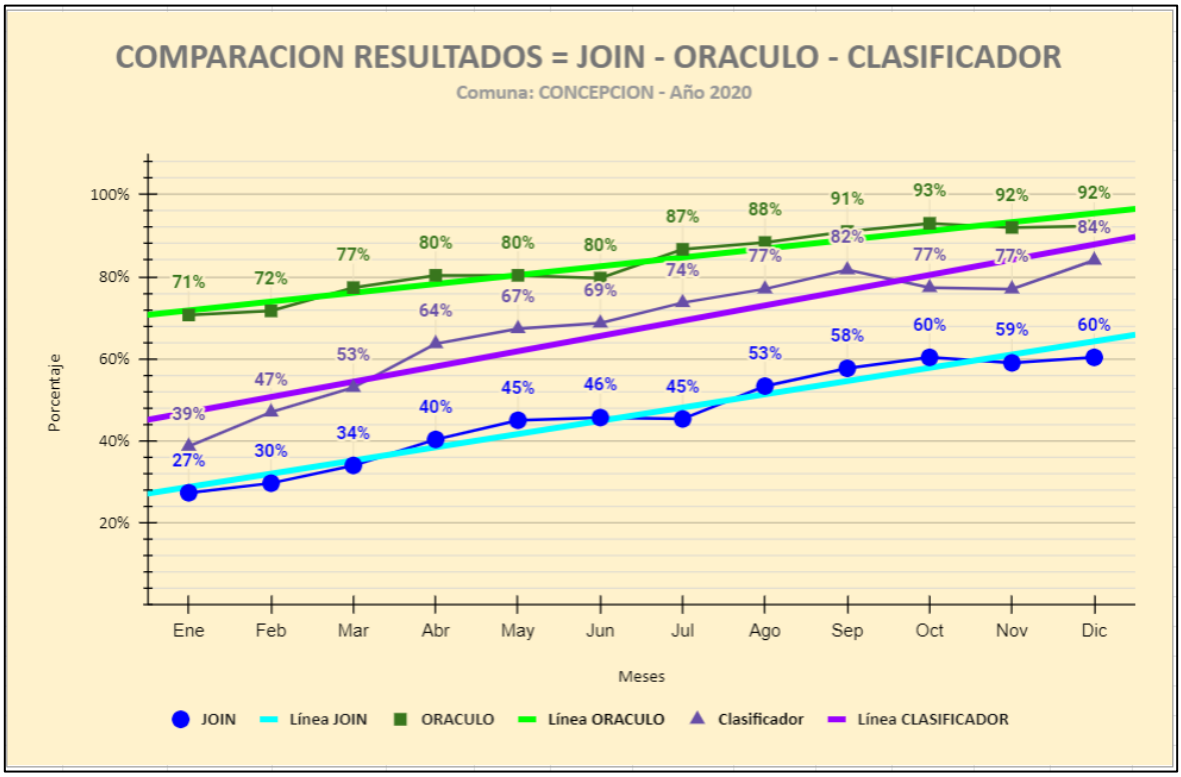


*Figura 51: Gráfico Comparativo de los Resultados de Todos los Algoritmos.*

Analizando el gráfico comparativo anterior al término de 12 meses (diciembre), se aprecia que el "**Algoritmo Join**" (como se trabaja actualmente en la Empresa), obtuvo un **60%** de éxito en la identificación de Clientes y que el algoritmo propuesto "**Algoritmo Clasificador**", obtuvo un **84%** de éxitos, y el "**Algoritmo Oráculo**" un **92%**, que es el máximo teórico (medida hipotética) en la identificación de Clientes (todos los clientes que ya se han visitado antes). Como se aprecia en la Tabla 34.

| Join | Clasificador | Oráculo |
|---|---|---|
| 60% | 84% | 92% |

*Tabla 34: Comparativo de los Algoritmos.*

## 4.4. Limitaciones de la Evaluación

Las limitaciones que se encontraron en esta evaluación y que son puntos a analizar en una futura implementación, fueron las siguientes.

- El universo de datos que fue considerado en esta evaluación, ya que se limitó a extraer información solamente de la comuna de "**Concepción"** que no es la más grande del país.

- El producto que se escogió "**Paquetería Internacional**", el cual no es el producto más vendido en la Empresa. Pero si era el que cumplía con la calidad en los datos.

- Se detectó que un Cliente puede tener más de una dirección dentro de la Comuna y lo más probable que también en otras comunas del país.

Dentro de las limitaciones que pueden surgir al implementar este modelo en producción, pueden ser las siguientes.

- Mala calidad de los datos, aunque esto fue levantado a las áreas correspondientes de la Empresa, el proceso no es rápido y requiere mucho esfuerzo y costo en las actualizaciones de los sistemas que involucran ingreso de Clientes.

- Queda la duda, si se consideran todos los otros productos o solo los más importantes y que generen un valor en la Empresa.

- El manejo de la historia de la data, puede repercutir en el desempeño del "**Algoritmo Clasificador**", si consideramos involucrar más variables.

- Posibilidad de redefinir prioridades ya programadas en el plan de desarrollo anual o sencillamente posponerlo para este año o el próximo.

# 5. Propuesta de Implementación

En esta sección se ilustra una propuesta de cómo se llevaría a producción en Correos de Chile el Algoritmo de Identificación de Clientes. Primero se describen las etapas de entrenamiento y determinación de Clientes, luego una breve descripción de la etapa actual de identificación de Clientes y la modificación que se debiese aplicar y por último se describe el proceso de reentrenamiento del Clasificador.

## 5.1. Algoritmo de Entrenamiento del Clasificador

En la Figura 52, se describe en forma gráfica como se entrenó el Algoritmo Clasificador con los datos de entrenamiento. Específicamente el proceso que involucró un etiquetado manual que abarcó aproximadamente **9.000** datos, los cuales tomaron como **20** días de trabajo dedicándole **4** horas diarias.

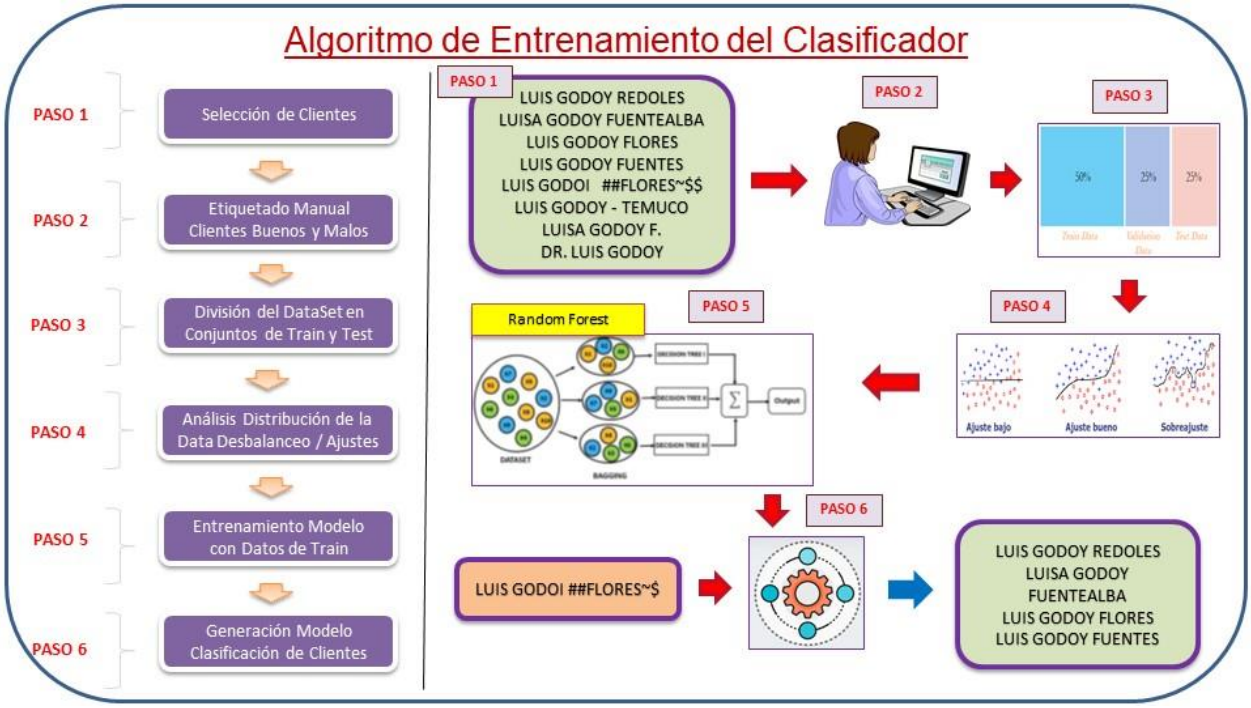


*Figura 52: Algoritmo de Entrenamiento del Clasificador.*

## 5.2. Algoritmo de Determinación de Clientes sin RUT

En la Figura 53, se describe en forma gráfica como se podría aplicar el Algoritmo de Determinación de Clientes. Cabe destacar que, en esta etapa de inferencia, solo se utiliza el Clasificador ya entrenado, es decir, totalmente automático.

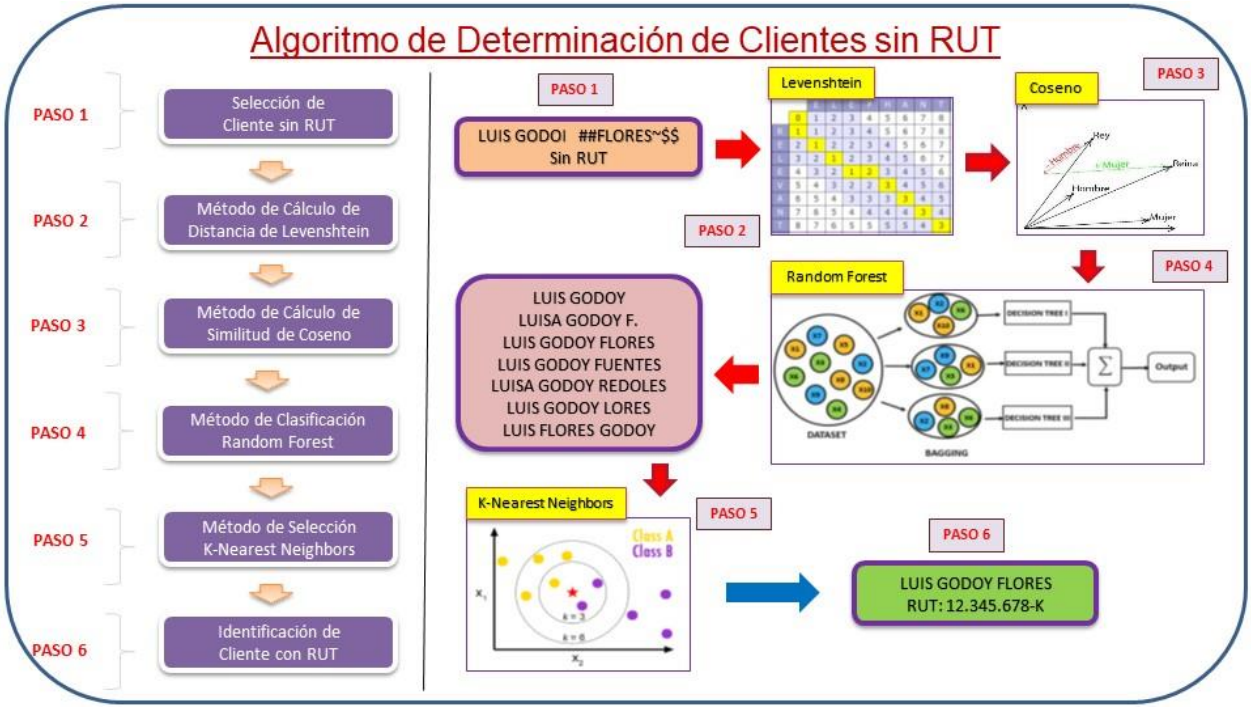


*Figura 53: Algoritmo de Determinación de Clientes sin RUT.*

## 5.3. Proceso de Identificación de Clientes (Actual)

En la Figura 54, se describe en forma gráfica como se realiza actualmente el proceso de Identificación de Clientes en Correos de Chile. Cabe destacar que los factores **Tiempo/Costo/Pasos/Errores** para identificar un cliente puede ser alto, pero al aplicar las mejoras beneficiaría a varios de los factores antes descritos.

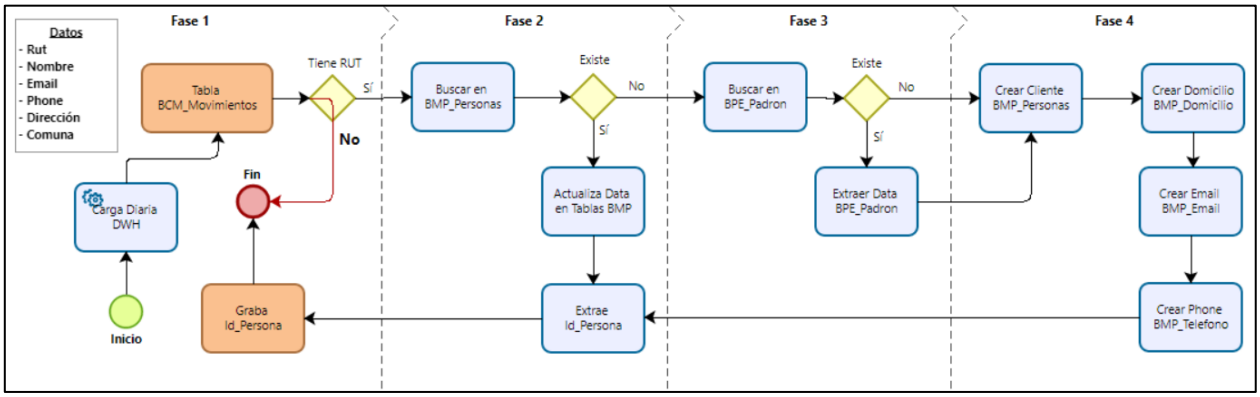


*Figura 54: Proceso (BPMN) de Identificación de Clientes (Actual).*

## 5.4. Proceso de Identificación de Clientes (Propuesto)

En la Figura 55, se describe en forma gráfica donde se alteraría de manera mínima el proceso de Identificación de Clientes sin RUT. Se estima un beneficio considerable en el corto plazo, ya que estadísticamente el ingreso mensual de nuevos Clientes en Correos de Chile es de **50 mil** aproximadamente, ya sea, en calidad de Clientes Remitentes y/o Clientes Destinatarios.

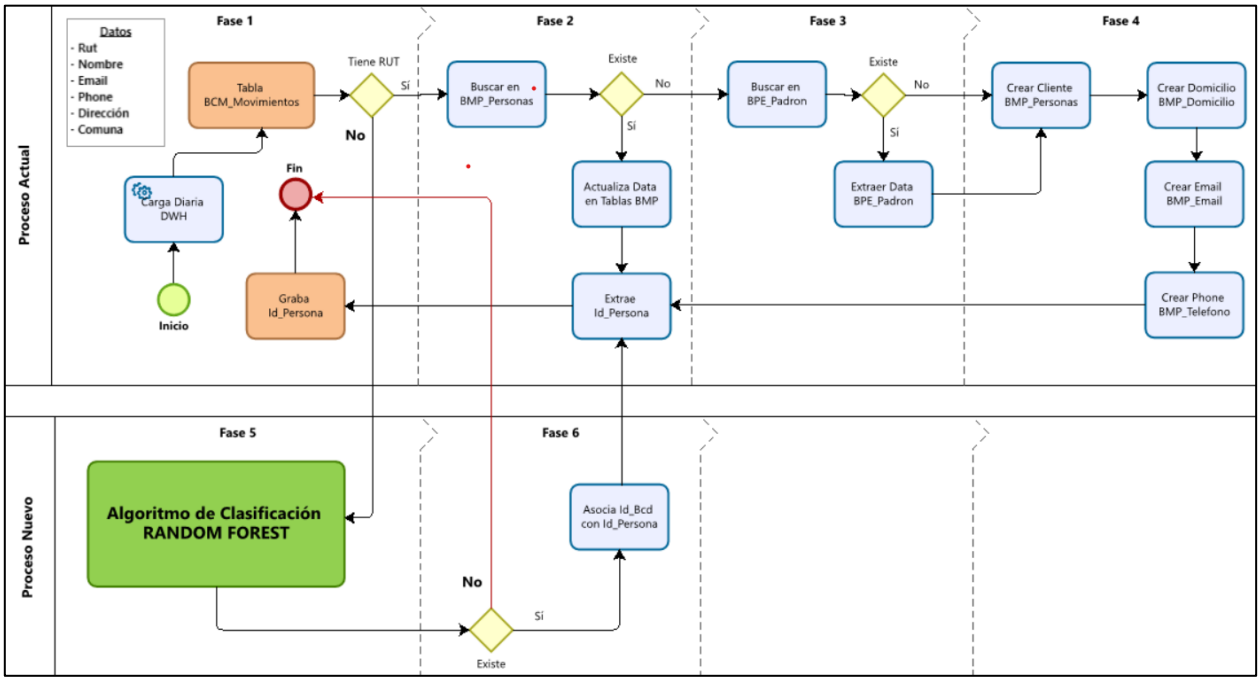


*Figura 55: Proceso (BPMN) de Identificación de Clientes (Propuesto).*

## 5.5. Proceso de Re-Entrenamiento del Clasificador

En la Figura 56, se describe en forma gráfica como se debe actualizar el Algoritmo Clasificador. Proceso que debe ser recurrente en el tiempo. Se estima que este reentrenamiento podría realizarse mensualmente, y ver si realmente hay variación, y si no la hay, se puede ir realizando en forma más esporádica. Pero si la variación es considerable, se puede realizar dos veces al mes.

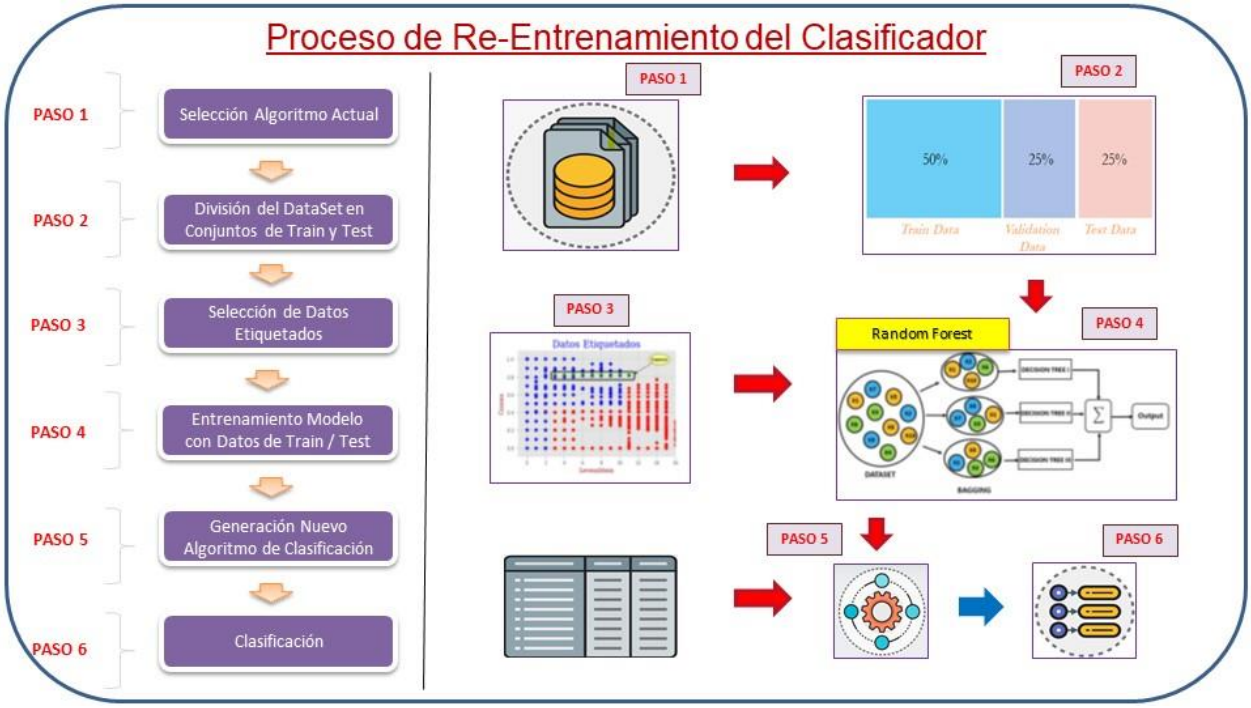


*Figura 56: Proceso de Re-Entrenamiento del Clasificador.*

# 6. Conclusiones y Trabajo a Futuro

## 6.1. Conclusiones

Los resultados obtenidos basados en Métodos de Distancia en donde se abordaron los problemas de los errores ortográficos, tipográficos, espacios en blancos, caracteres especiales, prefijos y sufijos en el campo **Nombre,** necesitaron una preparación y limpieza de la data que no fue visualizada en un comienzo. Aun así, los resultaron no fueron buenos cuando se trabajó por separado los cálculos de **Distancia de Levenshtein** y de **Similitud de Coseno**, en cambio cuando se aplicaron en conjunto tuvo mejores resultados.

Los resultados obtenidos basado en el "**Algoritmo Clasificador"** propuesto, en donde se abordaron los problemas de los tokens en desorden, tokens faltantes y abreviaturas o truncamiento de tokens en el campo **Nombre**, tuvieron resultados bastante buenos con respecto al "**Algoritmo JOIN"** que es la forma en cómo se trabaja actualmente en la Empresa.

Según los objetivos que fueron planteados en el Capítulo 1 de esta Tesis, el primer objetivo fue crear una Base de Clientes centralizada y diseñar procesos de extracción de todos los Clientes de los diferentes sistemas con que opera la Empresa, junto con lo anterior, también se realizaron procesos de limpieza de data, para aumentar la calidad de la información. El objetivo 1 fue cumplido con éxito.

El segundo objetivo fue clasificar e identificar los Clientes usando técnicas tradicionales, en lo cual los Clientes que tenían un Rut válido, su identificación y almacenamiento fue directo, además que su información se validó con la base de datos del SERVEL. Pero los que no poseían Rut fueron clasificados como "**indefinidos**" para así poder aplicarles el método del "**Algoritmo JOIN**", el cual llegó a un **60%** de identificación exitosa al cabo de 12 meses. Esto principalmente sucedió porque nos encontramos con diferentes formas de escritura en el **Nombre** lo que hace casi imposible determinar si un Cliente es el mismo que otro. El objetivo 2 fue cumplido con éxito.

El último objetivo, fue clasificar e identificar los Clientes usando técnicas de Machine Learning por medio del uso del Clasificador ***Random Forest***, el cual se cumplió, llegando al **84%** de identificación al cabo de 12 meses. Esto podría aumentar significativamente al incorporar nuevas variables al Clasificador, tal como, el Correo Electrónico y el Teléfono.

Las perspectivas de los resultados obtenidos en la empresa Correos de Chile, son abordable con un esfuerzo en costo y tiempo que puede ser sopesado por los beneficios de tener identificado al Cliente, lo que nos lleva a disminuir los errores, la nula gestión con ellos y lo principal, que el Cliente sienta que puede seguir confiando en la Empresa.

## 6.2. Trabajo a Futuro

Este trabajo sigue en constante evolución, ya que cada día surgen nuevas fuentes de datos que vienen con datos de mejor calidad. En lo próximo, se ha iniciado el levantamiento de interconexiones con otros sistemas que consumirán esta información.

En lo inmediato, se plantea aumentar la cantidad de datos, incorporando los datos del año 2021, ya que los años 2020 y 2022 están incorporados. Por otra parte, añadir otros atributos, ya sea, Email y/o Teléfono, al conjunto de propiedades de los Clientes, para explorar mejoras en los porcentajes de identificación de clientes.

Desde enero 2022, a la empresa de Correos de Chile está llegando el campo Rut de origen internacional (mundo privado), lo cual ayudará bastante a bajar la tasa de Clientes no identificados y contribuye a ser información fundamental para la Clasificación. En este escenario, las técnicas propuestas en esta memoria de todas maneras complementarán el tratamiento de datos en el ámbito internacional privado, y nacional.

# Bibliografía


Amón, I., & Jiménez, C. (2010). *Funciones de Similitud sobre Cadenas de Texto: Una Comparación Basada en la Naturaleza de los Datos*. https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxhbGdvcml0bW9zc2ltaWxhcmlkYWRkaXN0YW5jaWF8Z3g6NmU5YTJiNmY4ZTBhZDJj&pli=1.

Barrios, J. I. (2019). *La Matriz de Confusión y sus Métricas*. https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/. [Ultima Visita: 26-07-2022].

Breiman, L. (2001). *Random Forests. Machine Learning.* Springer, 1ra. Edición. https://doi.org/10.1023/A:1010933404324.

Campos, D. (2019). *Métricas de similitud para cadenas de textos*. https://medium.com/soldai/métricas-de-similitud-para-cadenas-de-texto-parte-i-introducción-154e4b724a27. [Ultima Visita: 26-07-2022].

Campos, J. (2021). *Una herramienta para afinar criterios de decisión*. http://www.debatesiesa.com/una-herramienta-para-afinar-criterios-de-decision/. [Ultima Visita: 26-07-2022].

Correos de Chile. (2018). *Memoria Correos de Chile de 2018*. https://correostransparente.correos.cl/memorias.html. [Ultima Visita: 27-07-2022].

Correos de Chile. (2019). *Memoria Correos de Chile de 2019*. https://correostransparente.correos.cl/memorias.html. [Ultima Visita: 10-12-2021].

Del Valle, A. (2017). *Curvas ROC y sus aplicaciones*. https://idus.us.es/bitstream/handle/11441/63201/Valle Benavides Ana Rocío del TFG.pdf?sequence=1&isAllowed=y. [Ultima Visita: 27-07-2022].

Dinov, I. D. (2018). *Data Science and Predictive Analytics. Springer, 1ra Edición,* https://doi.org/10.1007/978-3-319-72347-1.

Elmagarmid, A. K., Ipeirotis, P. G., & Verykios, V. S. (2006). *Duplicate record detection: A survey*. *IEEE Transactions on Knowledge and Data Engineering*, *19*(1), 1–16. https://ieeexplore.ieee.org/abstract/document/4016511.

Grootendorst, M. (2021). *9 Distance Measures in Data Science*. https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa. [Ultima Visita: 26-07-2022].

IBM Analytics. (2021). *¿Qué es el algoritmo de k vecinos más cercanos?*. https://www.ibm.com/cl-es/topics/knn. [Ultima Visita: 26-07-2022].

Invarato, R. (2016). *Hamming*. https://jarroba.com/hamming/. [Ultima Visita: 26-07-2022].

Kotu, V., & Deshpande, B. (2018). *Data science: concepts and practice. Morgan Kaufmann, 2da Edición*. https://www.elsevier.com/books/data-science/kotu/978-0-12-814761-0.

Kroese, D. P., Botev, Z. I., Taimre, T., & Vaisman, R. (2019). *Data science and machine learning: mathematical and statistical m*ethod*s. Chapman and Hall/CRC, 1ra Edición.* https://doi.org/10.1201/9780367816971.

Levenshtein, V. I. (1966). *Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 10(8), 707–710.*

Liaw, A., & Wiener, M. (2002). *Classification and regression by randomForest. R News*, *2*(3), 18–22.

Liu, C. (2020). *More Performance Evaluation Metrics for Classification Problems You Should Know*. https://www.kdnuggets.com/2020/04/performance-evaluation-metrics-classification.html. [Ultima Visita: 26-07-2022].

Markam, K., & DataSchool. (2014). *ROC Curves and Area Under the Curve explained*. https://www.dataschool.io/roc-curves-and-auc-explained/. [Ultima Visita: 26-07-2022].

Mitchell, T. (1997). *Machine learning (Vol. 1, Issue 9). McGraw-hill New York, 1ra Edición.*

Mucherino, A., Papajorgji, P. J., & Pardalos, P. M. (2009). *k-Nearest Neighbor Classification. In A. Mucherino, P. J. Papajorgji, & P. M. Pardalos (Eds.), Data Mining in Agriculture* (pp. 83–106). Springer, 1ra Edición. https://doi.org/10.1007/978-0-387-88615-2_4.

Orellana, J. (2018). *Arboles de decision y Random Forest*. https://bookdown.org/content/2031/. [Ultima Visita: 27-07-2022].

Qian, C. (2020). *Cálculo de distancia y similitud para el aprendizaje automático*. https://www.biaodianfu.com/distance.html#余弦相似度. [Ultima Visita: 27-07-2022].

Rodríguez, D. (2020). *La similitud de Jaro–Winkler*. https://www.analyticslane.com/2020/06/24/la-similitud-de-jaro-winkler/. [Ultima Visita: 27-07-2022].

Singh, S., & Gupta, P. (2014). Comparative Study ID3, CART and C4.5 Decision Tree Algorithm: a Survey. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.4929&rep=rep1&type=pdf. [Ultima Visita: 27-07-2022].

Slim, C. (2018). *Fundamentos de Machine Learning*. https://capacitateparaelempleo.org/pages.php?r=.tema&tagID=12544&load=12911&brandID=capacitate. [Ultima Visita: 27-07-2022].

Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Data mining introduction. Pearson 2da Edición*.

Vapnik, V. N. (1999). *An overview of statistical learning theory. IEEE Transactions on Neural Networks*, *10*(5), 988–999. https://ieeexplore.ieee.org/document/788640.

# Anexos

**Anexo A. Algoritmo Cálculo Similitud de Coseno.**

En la Imagen 1, se despliega la fórmula para el cálculo de similitud de Coseno, desarrollada en lenguaje Python.

```
[ ]  # Definiciones para usar Similitud de Coseno
     WORD = re.compile(r'\w+')

     def get_cosine(vec1, vec2):
         # print vec1, vec2
         intersection = set(vec1.keys()) & set(vec2.keys())
         numerator = sum([vec1[x] * vec2[x] for x in intersection])

         sum1 = sum([vec1[x]**2 for x in vec1.keys()])
         sum2 = sum([vec2[x]**2 for x in vec2.keys()])
         denominator = math.sqrt(sum1) * math.sqrt(sum2)

         if not denominator:
             return 0.0
         else:
             return float(numerator) / denominator

     def text_to_vector(text):
         return Counter(WORD.findall(text))

     def get_similarity(a, b):
         a = text_to_vector(a.strip().lower())
         b = text_to_vector(b.strip().lower())

         return get_cosine(a, b)
```

*Imagen 1: Rutina Python del Cálculo de Similitud de Coseno.*

**Anexo B. Algoritmo Cálculo Distancia de Haversine**.

En la Imagen 2, se despliega la fórmula para el cálculo de distancia de Haversine, desarrollada en lenguaje Python.

```
[ ]  # Calculo de la Distancia entre 2 Puntos Geográficos usando la Formula Haversine
     def CalculaDistancia(lat1, lon1, lat2, lon2, medida):

         rad = math.pi/180
         dlat = lat2-lat1
         dlon = lon2-lon1
         R = 6372.795477598
         a = (math.sin(rad*dlat/2))**2 + math.cos(rad*lat1)*math.cos(rad*lat2)*(math.sin(rad*dlon/2))**2
         distancia = 2*R*math.asin(math.sqrt(a))

         if medida == 'M':
           distancia = distancia * 1000

         return distancia
```

*Imagen 2: Rutina Python del Cálculo de la Distancia Haversine.*

## Anexo C. Algoritmo Método Oráculo.

En la Imagen 3, se aprecia una muestra del código desarrollado en lenguaje Python usando librerías Pandas y la fórmula del Cálculo de Distancia Haversine.

```
[ ]  # Calcula las Entregas Reales con Formula de Distancia
     acumula = 0
     i = 0
     inicio = time.strftime("%H:%M:%S")
     print('Registros :',i,'Coinciden:',acumula,'Tiempo:',time.strftime("%H:%M:%S"))

     for index, mes in df_Entregas_Mes.iterrows():
         for index, acu in df_Entregas_Acu.iterrows():
           if acu[6] < mes[6]: # x FECHA
               distancia = CalculaDistancia(mes[4],mes[5],acu[4],acu[5],'M')
               if distancia <= 10:
                 acumula += 1
                 print('Registros:',i,' - Coinciden:',acumula,acu[0],mes[0],' - Haversine:',distancia)
                 break
         i += 1
```

*Imagen 3: Rutina Python del Cálculo del Algoritmo Oráculo.*

## Anexo D. Algoritmo Método Join.

En la Imagen 4, se aprecia una muestra del código desarrollado en lenguaje Python usando librerías Pandas, la condición de igualdad y la fórmula de Cálculo de Distancia Haversine.

```
[ ]  # Calcula las Entregas Reales con Formula de Distancia
     acumula = 0
     i = 0
     inicio = time.strftime("%H:%M:%S")
     print('Tiempo:',time.strftime("%H:%M:%S"))

     for index, mes in df_Entregas_Mes.iterrows():
         for index, acu in df_Entregas_Acu.iterrows():
           if acu[6] < mes[6]: # x FECHA
             if acu[1] == mes[1]:  # x NOMBRE para JOIN
               distancia = CalculaDistancia(mes[4],mes[5],acu[4],acu[5],'M')
               if distancia <= 10:
                 acumula += 1
                 print(i,'-',acumula,'- Metros:',distancia,acu[1],'-',mes[1])
                 break
         i += 1
```

*Imagen 4: Rutina Python del Cálculo del Algoritmo Join.*