Pontificia Universidad Católica del Ecuador

Facultad de Ingeniería

Escuela de Sistemas

TEMA

Modelo de aprendizaje para clasificación de correos electrónicos.

AUTOR:

VELEZ ZAMBRANO JOFFRE GABRIEL

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS DE INFORMACIÓN

QUITO, NOVIEMRBE DE 2022

DEDICATORIA

A mi madre, por enseñarme los valores del respeto, del altruismo, de la responsabilidad y la gratitud,

A mi padre, por nunca darse por vencido a pesar de las adversidades que la vida le ha puesto, haciéndose presente sin importar la distancia,

A mi hermana, por estar a mi lado en los buenos y en los malos momentos, aconsejándome y guiándome en cada oportunidad,

A mi hermano, por ser mi confidente durante este viaje, a quién le puedo contar todo sin necesidad de usar palabras,

A Maxwell, que me ayudo a encontrar la paz y la felicidad en los momentos más tensos de mi vida,

Y finalmente, a mí, que supo aprovechar todo lo que mi familia me ha dado, agradeciendo lo bueno y aprendiendo de lo malo. Que este sea el comienzo de una vida de éxitos y felicidad junto a todas las personas en esta dedicatoria.

AGRADECIMIENTO

Agradezco a mi director Mtr. Charles Escobar, por sus consejos y su guía, un profesional admirable y de gran calidad humana.

RESUMEN

El correo electrónico es usado para fines comunicativos en la época actual, pero eso no lo exonera de ser usado con fines malignos. Por esta razón ha sido necesario implementar medidas de seguridad que filtren que tipos de correos son los deseados y cuáles no. Sin embargo, las medidas utilizadas por las empresas proveedoras del servicio de mensajería tiene fallos y no se aprovecha tecnologías con potencial como el aprendizaje de máquina o la minería de datos.

Este trabajo de titulación analiza distintas fuentes de datos para encontrar una idónea que permita una buena aplicación de estos. Así mismo, se compara diversas metodologías de minería de datos, de forma que la seleccionada permita un desarrollo eficaz y eficiente según el contexto del proceso. Para que finalmente, se realice el proceso de minería de datos adquiriendo información y sugiriendo soluciones basados en los resultados que genere el modelo.

ÍNDICE

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO I: INTRODUCCIÓN

1. MARCO DE REFERENCIA

1.1. Justificación

En la actualidad, las personas cuentan con diversos canales para comunicarse: servicios de mensajería instantánea, redes sociales, entre otros. Pero, el que destaca entre todos es el email, existen estudios que dicen que “el uso del correo electrónico se ha masificado de forma que en el año 2018 se enviaron alrededor de 236 billones de emails, sin embargo, el 53.5% de estos fueron correos no deseados” (Karim et al., 2019). Esto supone que los correos basura son la mitad de los emails enviados durante el año.

Los correos no deseados aparecieron junto a los correos inofensivos, aunque estos cuentan con un objetivo distinto que el de comunicar un mensaje. Por ejemplo, el virus informático, que se originó en 1999, Melissa como se explica en un artículo, “se propagaba por medio de un correo electrónico, conteniendo un archivo que al abrirlo contagiaría a la computadora. También un virus originado en el año 2000 y escrito en Visual Basic, ILoveYou era un troyano que se propagó por correo electrónico conteniendo un archivo capaz de registrar contraseñas, direcciones IP, entre otros datos” (Herrero Sanz, 2019).

Así mismo Karim et al. detallan que “las perdidas en el año 2018 por estos correos no deseados fueron aproximadamente de 12.5 billones de dólares americanos”. Se estima que los valores de daño sigan creciendo a través de los años.

Paralelamente, la minería de datos ha permitido realizar predicciones basadas en eventos pasados, por lo que, ayuda a automatizar procesos de forma eficaz y eficiente. Este proceso de predicciones es sumamente acogido en empresas, de manera que se han establecido distintas metodologías para realizar el trabajo.

Específicamente el periódico El Universo relata que, “en la fecha de 9 de octubre del año pasado, la empresa Banco Pichincha recibió un ataque que saturó los servicios que la compañía disponía, dejando a los clientes sin posibilidad de uso” («Ciberataque a Banco Pichincha fue realizado por atacantes internacionales, se revela en Comisión de Desarrollo Económico», 2021). El periódico El Comercio brinda el ejemplo de “La Corporación Nacional de Telecomunicaciones, o CNT, en la cual el día 15 de julio del 2021 los servicios en agencias, atención al cliente y otros más, sufrían de intermitencias por culpa de un secuestro de información” (Díaz, 2021). A pesar de que en ambos casos no se detalló de manera oficial el origen de los ataques, no se puede descartar que se generaron por correos con contenido malicioso.

Por las razones antes mencionadas, este proyecto de disertación busca comparar los distintos modelos de minería de datos, seleccionar el que destaque y entrenarlo para que clasifique correos electrónicos inofensivos y no deseados. De esta forma, aplicar el método que se adapte de mejor forma a las necesidades establecidas y analizar los resultados que brinde este.

1.2. Planteamiento del problema

Los correos electrónicos o emails son una herramienta imprescindible para la comunicación entre las personas, siendo usado con todo tipo de fines como el laboral, el personal o para noticias y notificaciones. Estos servicios de mensajería permiten el intercambio de todo tipo de mensajes e información: textos planos, archivos, enlaces, entre otros.

No obstante, el email también es utilizado con fines nocivos. Existen registros de correos electrónicos que contienen archivos con código malicioso que infectan computadores. De misma forma, se puede encontrar correos falsos que personifican individuos y entidades. Este tipo de correos se denomina <<spam>>.

Para combatir el spam, los proveedores de correo electrónico utilizan diversos sistemas, sin embargo, estos no son completamente precisos. Como consecuencia de este fallo, distintas

personas han sido víctimas de estafas en línea perdiendo datos personales, información bancaria o una suma de capital definida por el victimario.

Dado esto, se puede definir la siguiente problemática principal:

- No existe un modelo de minería de datos que clasifique los correos electrónicos.

Y los siguientes problemas secundarios:

- No se usan distintas fuentes de datos para el análisis de correos electrónicos fraudulentos.
- No se ha identificado los modelos que permitan la clasificación de correos electrónicos.
- No existe una comparativa que indique qué modelo de minería de datos sobresale para clasificar correos electrónicos.

1.3. Objetivo General

- Diseñar un modelo de minería de datos para la clasificación de correos electrónicos.

1.4. Objetivos Específicos

- Analizar distintas fuentes de datos de correos electrónicos asociados a modelos de aprendizaje.
- Identificar modelos de minería de datos para la clasificación de correos electrónicos.
- Aplicar el modelo de minería de datos que destaque para la clasificación de emails.

1.5. Antecedentes

Por el alto tráfico de correos electrónicos no deseados que hay diariamente, la búsqueda de una solución eficiente se ha intensificado. Diversas personas proponen un modelo de aprendizaje

por medio de la minería de datos como el caso del trabajo del PhD Muhammad donde “compara 14 modelos de clasificación diferentes como árboles aleatorios o perceptrón multicapa” (Muhammad Abdulhamid et al., 2018).

Sin embargo, no se ha encontrado un trabajo que compare los distintos modelos que existen para minar datos. El proceso de seleccionar el modelo que se adapte a los datos es vital debido a que cada metodología comprende distintas fases y se difieren en su complejidad haciendo que la minería se facilite o se complique según los datos que se dispongan.

Por lo que este trabajo busca aportar a la sociedad con una comparativa para establecer que modelo se adapta de mejor forma a los datos extraídos de los emails basura.

1.6. Alcance

Este trabajo de titulación tiene como alcance analizar correos electrónicos para determinar un modelo de minería de datos para clasificar correo basura. Para ellos se contará con datos de distintas fuentes, por ejemplo: correos electrónicos inofensivos, emails no deseados, estructuras de datos ya existentes, entre otros.

Posteriormente se comparará diversos modelos para minería de datos para definir y seleccionar el que mejor se ajuste al conjunto de datos, para después aplicarlo por medio del entrenamiento de distintos algoritmos de clasificación para que cataloguen correos electrónicos. El resultado de este trabajo será un modelo de minería de datos para la clasificación de emails junto a un análisis de la eficacia y eficiencia de este.

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

2. Marco Teórico

2.1. Correo No Deseado

El correo no deseado, o simplemente spam, se define como “una comunicación no deseada con intención de ser entregada a un destinatario de forma directa o indirectamente a pesar de las medidas que prevengan su entrega” (Cormack, 2008).

De igual forma, Cormack define las siguientes características que disponen los correos spam:

- No solicitado: la mayoría de los receptores no precisan de este tipo de correos, sin embargo, existen individuos que contestan el mensaje y arrepintiéndose posteriormente.
- Indiscriminado: primigeniamente el spam se distribuye sin ningún tipo de relación entre el emisor y receptor. Por lo que, se envía sin diferenciar a las personas a quienes se les envía el correo.
- Falso: debido a que el correo es no solicitado e indiscriminado, este debe encubrirse con una fachada para aparentar legitimidad. De esta forma intenta asegurar sobrepasar los filtros de spam.
- Portador de carga: también llamado payload, la carga es la parte del spam que realiza la acción maliciosa. Esta puede tener distintas formas como un enlace, un archivo o un mensaje que debe llamar la atención del usuario para que pueda ejecutarla.

Con estas cualidades los correos no deseados buscan beneficiarse de todos los usuarios a quien le llegue el mensaje, produciendo estafas, robo de información, robo de identidad entre otros males.

2.2. Minería de Datos

“Son los procesos para clasificar grandes estructuras de datos con la finalidad de descubrir información útil según el caso de aplicación” (¿Qué es la minería de datos?, 2021). Para procesar los datos se utilizan técnicas de estadística, inteligencia artificial, aprendizaje de máquinas, entre otros temas.

Para realizar estos procesos se debe “utilizar un lenguaje de programación, este puede ser cualquiera debido a que en la actualidad la mayoría de los lenguajes cuentan con librerías que permiten trabajos de aprendizaje de máquina y minería de datos” (Jamsa, 2020). Para este trabajo se estableció el lenguaje de programación Python, un lenguaje enfocado mayormente al aprendizaje de máquina y a la inteligencia artificial.

2.3. Modelo de Minería de Datos

Debido a la longevidad que tiene la minería de datos se han establecidos distintos modelos para poder realizarla de forma sistemática y eficiente. Cada modelo cuenta con cierto número de fases y las detalla de forma clara para ejecutar el trabajo.

2.3.1. CRISP-DM

“CRISP-DM proviene de **CR**oss-**I**ndustry **S**tandard **P**rocess for **D**ata **M**ining y fue desarrollado en el año 1996 por Daimler-Benz, ISL y NCR” (Shafique & Qasier, 2014). Este modelo define seis fases que se deben realizar para completar el proceso de minería. Estas son:

- Entendimiento del negocio: se centra en comprender la finalidad con la que se hará la minería y otros criterios como el éxito del trabajo, requerimientos y terminologías.
- Entendimiento de los datos: esta fase consta en un primer acercamiento con los datos que se utilizarán. Se revisa los tipos de datos con los que se trabajarán y se plantea que variables brindan más información al resultado.
- Preparación de los datos: para que el algoritmo funcione de la mejor forma es necesario limpiar y preparar el conjunto de datos. Esto supone categorizar, vectorizar, transformar, entre otras tareas.
- Modelado: se seleccionan modelos de aprendizaje de máquina para aplicar, así mismo se usan varios con distintos parámetros o directamente distintos algoritmos.
- Evaluación: para evaluar los resultados que dan los modelos es necesario establecer el tipo de métrica que usa. Posteriormente se da la interpretación de los resultados recolectados.
- Despliegue: esta última fase busca darle un uso al conocimiento que se genera al interpretar los resultados. Para esto se establecen conclusiones, recomendaciones y planes de acción.

2.3.2. KDD

KDD o conocido también como Descubrimiento de Conocimiento en Bases de Datos (**K**nowledge **D**iscovery **D**atabases) es “un modelo iterativo e interactivo que cuenta con siete pasos” (Shafique & Qasier, 2014).

- Entendimiento del dominio de la aplicación: el cliente explica los objetivos y uso de la aplicación.

- Selección de los datos: durante esta fase se crea el conjunto de datos desde distintas fuentes de datos.
- Preprocesamiento de los datos: es necesario que no existan inconsistencias en los datos. Por esto se realizan varias tareas para limpiar imperfecciones que tenga el conjunto de datos.
- Transformación: el conjunto de datos se somete a trabajos de transformación y reducción para que el algoritmo se adapte de la mejor forma posible.
- Minería de datos: para esta fase se debe escoger las tareas y algoritmos que se realizarán.
- Evaluación e interpretación: como indica su nombre, se da una interpretación a los resultados que genera la minería de datos.
- Conocimiento: con la interpretación de los resultados, es necesario aplicar acciones para darle un uso adecuado a los resultados.

2.3.3. SEMMA

SEMMA son las siglas de **S**ample, **E**xplore, **M**odify, **M**odel y **A**ccess, o Muestreo, Exploración, Modificación, Modelado y Acceso en español. “SEMMA es un modelo de minería de datos desarrollado por el instituto SAS” (Shafique & Qasier, 2014). Como se indica en su nombre, SEMMA cuenta con cinco fases para el trabajo.

- Muestreo: se la define como una fase opcional donde se extrae un pequeño conjunto de datos del conjunto mayor para poder explorar y manipular la data sin que haya complicaciones.
- Exploración: como indica su nombre, se explora el grupo de datos para tener mejor comprensión del problema y poder descubrir anomalías y tendencias.

- Modificación: se transforman, reducen y limpian los datos para que el modelo de aprendizaje se adapte de la mejor forma.
- Modelado: en esta fase hay que seleccionar y aplicar un modelo de aprendizaje de máquina que mejor se ajuste al conjunto de datos que se está usando.
- Acceso: se evalúa la utilidad que tuvo el modelo además de estimar el rendimiento de este para evaluarlo, tomando en cuenta todos los aspectos posibles.

2.3.4. Microsoft's Team Data Science Process

La empresa Microsoft ha desarrollado su propio modelo para minar datos, lo ha llamado TDSP que proviene de Team Data Science Process, y la empresa lo define como "una metodología ágil e iterativa que brinda soluciones analíticas predictivas" (What is the Team Data Science Process? - Azure Architecture Center, s. f.). El TDSP se compone de cinco fases las cuales son:

- Entendimiento del negocio: se trabaja junto al cliente para establecer que objetivos se quiere completar y cuál es el problema del negocio y se identifican las fuentes de datos que pueden ayudar a cumplir las metas.
- Adquisición y entendimiento de los datos: se busca producir un conjunto de datos limpios y que estos permitan alcanzar una respuesta deseable. Por lo cual se revisa y se limpian los datos.
- Modelado: en esta fase se debe escoger los datos que aporten al modelado de aprendizaje, así mismo se establece que modelo se usará y se entrenará al algoritmo.
- Despliegue: el modelo se empieza a producir por medio de APIs para implementarlo en páginas web, aplicaciones back-end, entre otros.
- Aceptación del cliente: finalmente el cliente da el grado de satisfacción con el modelo en un ambiente implementado junto al sistema.

2.3.5. SCRUM-DS

Para adaptar SCRUM a la ciencia de datos, fue necesario basarse en los pasos de CRISP-DM, formando SCRUM-DS fusionando características de ambos modelos. “SCRUM se forma de cinco fases que se pueden adaptar a las fases de CRISP-DM” (Baijens et al., 2020). Para adaptar se redujo a tres fases únicamente.

- Refinamiento: siempre debe realizarse antes de un sprint, aquí se encuentra el equipo y se habla de las prioridades y necesidades del cliente. Así mismo se establece que hará el producto final.
- Sprint y Stand-Up diario: el sprint es un periodo donde se realizan todas las actividades para el desarrollo e implementación del software. Por otro lado, el stand-up son reuniones que realiza el equipo de forma diaria para revisar el trabajo hecho el día anterior y establecer el trabajo futuro.
- Retrospectiva y revisión del sprint: la retrospectiva se realiza al finalizar un sprint donde se hace una reunión para usar medidas de mejora en los siguientes sprints. Finalmente, en la revisión hay otra junta con el cliente el cual dará su opinión respecto al sprint que se le presente.

2.4. Algoritmo de Clasificación

“La clasificación es de las tareas más comunes en el aprendizaje de máquina y el tipo de variable que se predice debe ser categórica.” (Novaković et al., 2017). Existen diversos algoritmos de clasificación, cada uno cuenta con sus fortalezas y debilidades, y para saber cuál aplicar es necesario conocer la problemática a resolver.

Entre los modelos más conocidos se tiene: árboles de decisión, vecinos cercanos, entre otros modelos. Así mismo estos cuentan con sus métricas de evaluación como son la matriz de confusión que nos da la cantidad de aciertos y fallos que ha tenido el modelo, la precisión que

permite conocer el porcentaje de aciertos y otras métricas para conocer la eficiencia del algoritmo.

CAPÍTULO III: METODOLOGÍA

3. Metodología de desarrollo del plan de tesis

3.1. Investigación Cualitativa

“La investigación Cualitativa genera datos no numéricos, brinda un grado de entendimiento a las creencias, experiencias, actitudes, comportamientos e interacciones humanas” (Pathak & Jena, 2013). Este tipo de investigación permite trabajar con datos que no pueden ser medidos en forma numérica. Para esto se utiliza métodos de recolección de datos como entrevistas, encuestas y observaciones.

De esta forma se asegura que cada parte que participe en la investigación tenga un rol activo dentro de la misma. Generalmente se usa en trabajos psicológicos y psiquiátricos ya que permite tener una mayor comprensión de los pacientes. Una vez se mezcla con características numéricas los resultados de la investigación serán optimizadas y, por lo tanto, más comprensibles.

3.2. Investigación Aplicada

“La investigación aplicada tiene la meta de generar conocimiento con una aplicación al momento y en medio plazo” (Lozada, 2014). Las aplicaciones son parte de un adicional de otro tipo de investigaciones debido a que tienen que estar enfocadas a un grupo real, sea una empresa o un grupo social.

Al terminar el trabajo investigativo se cuenta con un prototipo o un producto prácticamente terminado para su revisión y, en el mejor de los casos, su producción. Una vez descritos estos puntos, este trabajo de titulación se realizará basándose en la investigación aplicada ya que se generará un modelo de aprendizaje de máquina para clasificar correos electrónicos.

3.3. Metodología de desarrollo de software

3.3.1. Modelo que aplica

Como se explica en el capítulo dos, existen diversos modelos para realizar minería de datos, como Crisp-DM, KDD, TDSP, entre otros. Sin embargo, según se estableció en el primer capítulo, el modelo aplicado no será decidido hasta un análisis profundo de las metodologías establecidas anteriormente.

CAPÍTULO IV: DISEÑO DE UN MODELO DE MINERÍA DE DATOS PARA LA CLASIFICACIÓN DE CORREOS ELECTRÓNICOS

4.1. Análisis de distintas fuentes de datos de correos electrónicos asociados a modelos de aprendizaje

Para el análisis del origen de los datos se considera 3 tipos de fuentes distintas: de orden público, privadas y personales. Cada grupo de datos disponen de características que pueden ponderarse y clasificarse.

Las fuentes de datos públicas se caracterizan por ser de libre acceso y se encuentran comúnmente en repositorios en línea como kaggle, sitios gubernamentales, o el repositorio de aprendizaje de máquina de la universidad de California en Irvine (UCI). Usualmente son usados con fines educativos y prácticas de entrenamiento.

Por otro lado, los orígenes de datos privados se distinguen por ser propios de una empresa o compañía, para acceder a ellos es necesario cumplir con un rol que autorice dicha acción sobre este conjunto de datos. El uso que tienen estos datos es únicamente interno en función de generar un beneficio a la empresa. De esta forma, estos conjuntos de datos son más complicados de obtener sin la aprobación de una compañía de por medio.

Finalmente, los datos personales son los que generan de forma inherente al humano, por ejemplo, la bandeja de correo personal como Gmail o Outlook. El problema de esta fuente de datos es el volumen de datos, debido a ser una sola persona los datos generados son significativamente menores al resto de las otras fuentes, aunque estos son los más sencillos de conseguir.

Con los tipos de orígenes aclarados existen diversos grupos de datos, también conocidos como datasets, que pueden ser utilizados para el desarrollo del modelo de aprendizaje de máquina para la clasificación de correos electrónicos.

Por conveniencia de este proyecto de titulación los correos que se encuentren en el idioma español tienen un mayor grado de importancia a otro tipo de idiomas, sin embargo, no se niega la posibilidad de trabajar con mails traducidos a español.

Antes de realizar el trabajo es necesario un análisis profundo sobre las ventajas y desventajas que tienen los datos.

Los grupos de datos con los que se trabajan son los siguientes:

- *spam_ham_dataset.csv* (Barish, 2020)
- *spambase.data* (Hopkins et al., s. f.)
- *Enron1.txt* (Metsis et al., 2006)
- *dataset_spam_personal.csv*

Tabla 1 *Comparación de las características principales de los datasets*

	spam_ham_dataset	**spambase**	**Enron1**	**dataset_spam_personal**
Núm. de registros	5171	4601	5975	156
Núm. de variables	4	57	2	5
Idioma	Inglés	Inglés	Inglés	Español
Tipo	Público	Público	Público	Personales

A partir de lo explicado anteriormente y de la Tabla 1, se ponderará las características que tienen cada dataset para establecer con cuál trabajar. Para la calificación se estableció un rango donde 2 es *bueno*, 1 representa un *neutro* y 0 es *malo*.

De esta forma la tabla cuantitativa tiene la siguiente forma:

Tabla 2 *Comparación cuantitativa de los datasets*

	spam_ham_dataset	**spambase**	**Enron1**	**dataset_spam_personal**
Cantidad de registros	2	1	2	0
Facilidad para encontrar los datos	2	1	0	1
Facilidad para entender las variables	2	0	1	2
Idioma	1	1	1	2
TOTAL /8	**7**	**3**	**4**	**5**

Una vez evaluado los distintos datasets, se concluye que el *spam_ham_dataset* es el adecuado para utilizar con el modelo de aprendizaje debido a su alto puntaje en la valoración siendo siete sobre ocho posible puntos. El dataset seleccionado tiene características que se adaptan de mejor forma a las necesidades de este trabajo, con su único defecto siendo que los correos se encuentran en inglés. Sin embargo, se utilizarán librerías de programación para traducir el contenido de las variables y poder trabajar con correos en español.

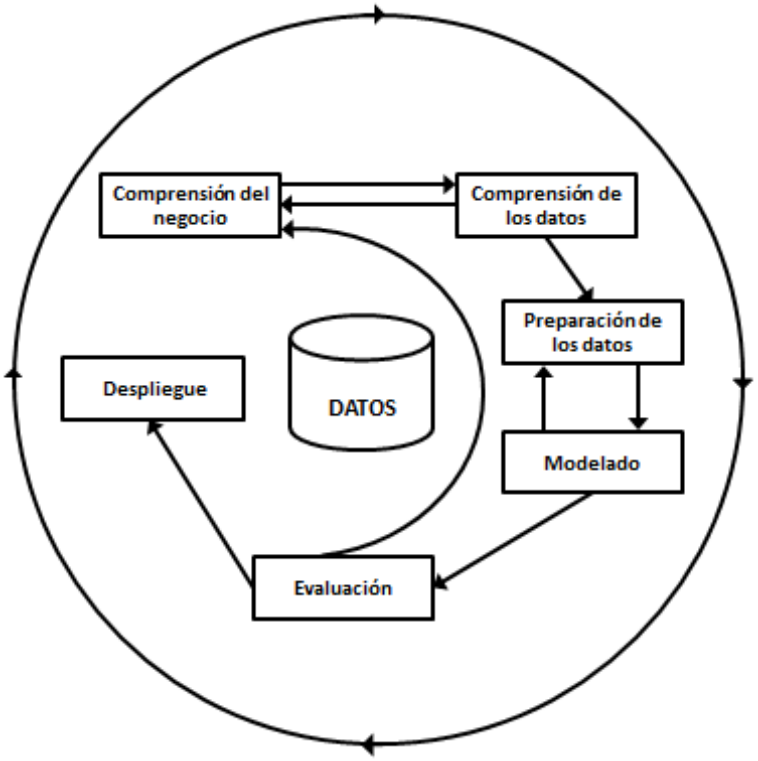
De igual forma, el *dataset_spam_personal* es una buena opción con la debilidad de que su número de registros es diminuto lo que puede ocasionar un sobre entrenamiento del modelo provocando que al darle más correos el modelo no será capaz de clasificarlos de forma correcta.

4.2. Comparativa de modelos de minería de datos para la clasificación de correos electrónicos

Por la antigüedad del proceso para minar datos se desarrollaron varios modelos que apoyan con una guía para que el resultado del trabajo sea el mejor, sin derrochar recursos ni dinero. Sin embargo, no se puede utilizar todas las guías existentes por la cantidad de tiempo y recursos disponibles, por lo que se debe seleccionar uno según los requerimientos que el cliente establezca. Para la comparación que se realizará los modelos seleccionados fueron los siguientes:

- CRISP-DM

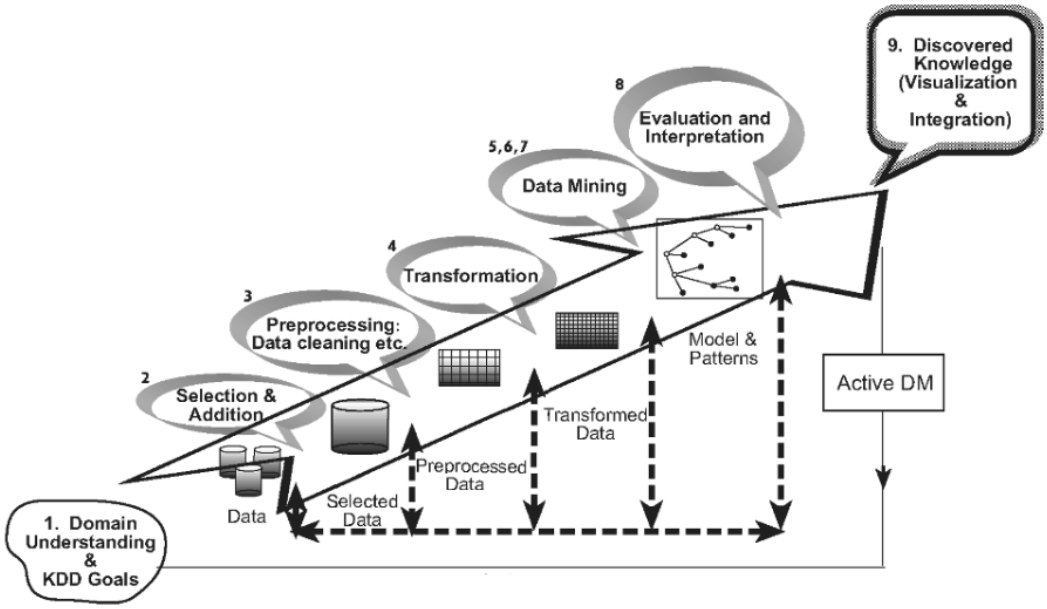
Ilustración 1 *Fases del modelo CRISP-DM*



Nota. Tomado de Peralta, F. C. (2014). Proceso de Conceptualización del Entendimiento del Negocio para Proyectos de Explotación de Información. *Revista Latinoamericana de Ingenieria de Software*, 2(5), 273. https://doi.org/10.18294/relais.2014.273-306

- KDD

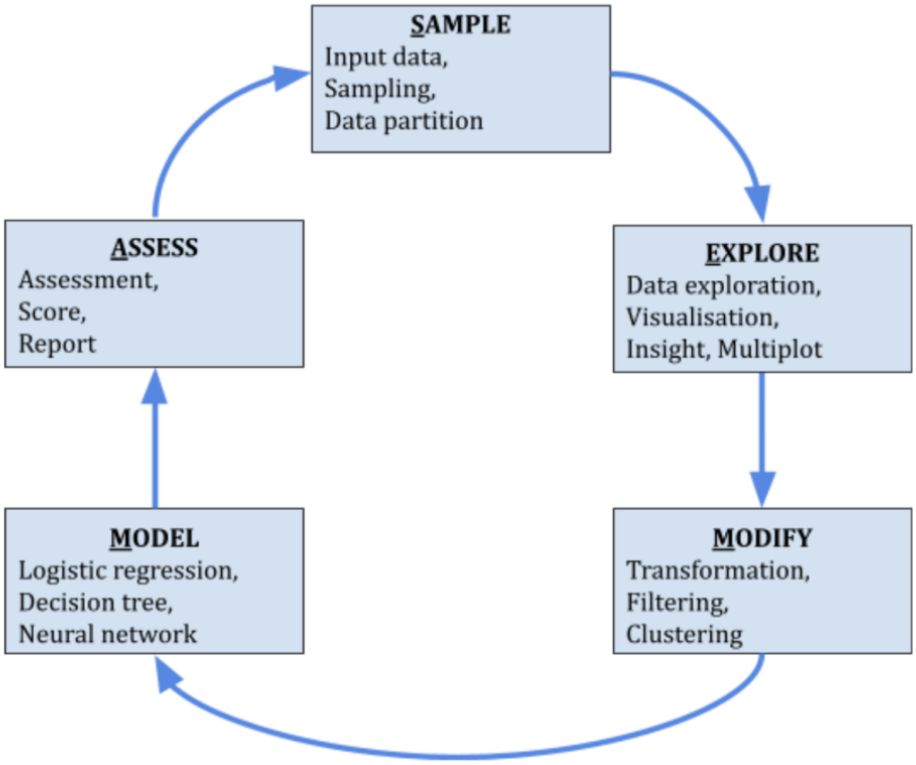
Ilustración 2 *Fases del modelo KDD*



Nota. Tomado de Maimon, O. & Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook* (2nd 2010 ed.). Springer.

- SEMMA

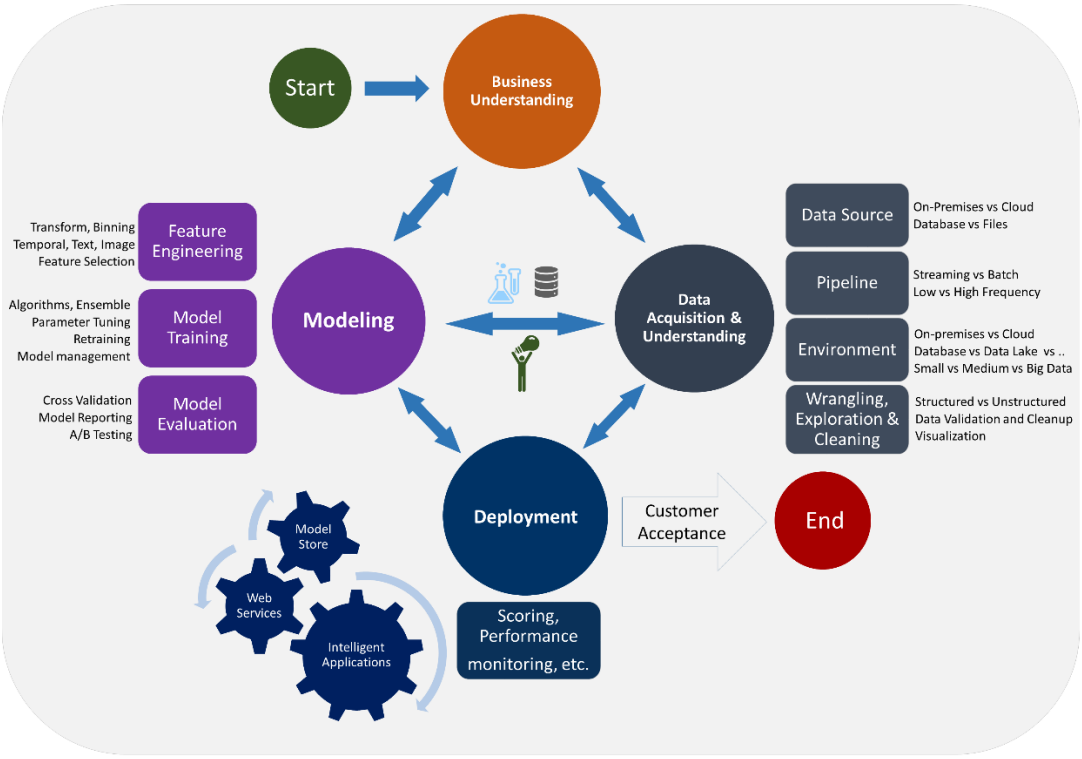
Ilustración 3 *Fases del modelo SEMMA*



Nota. Tomado de Zubchenko, A. (2022, 27 octubre). *Data Collection for Machine Learning: The Complete Guide*. Waverley. https://waverleysoftware.com/blog/data-collection-for-machine-learning-guide/

- TDSP

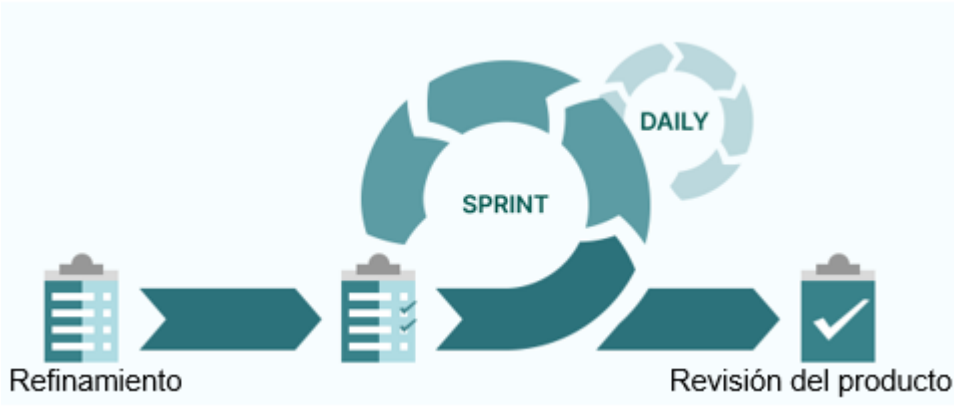
Ilustración 4 *Fases del modelo TDSP*



Nota. Tomado de *What is the Team Data Science Process? - Azure Architecture Center*. (s. f.). Microsoft Learn. Recuperado 6 de octubre de 2022, de https://learn.microsoft.com/en-us/azure/architecture/data-science-process/overview

- SCRUM-DS

Ilustración 5 *Fases del modelo SCRUM-DS*



Nota. Adaptado de Stsepanets, A. (2022, 16 septiembre). *Marco Scrum:¿Cómo gestionar trabajo y proyectos complejos?* Gantt Chart GanttPRO Blog. https://blog.ganttpro.com/es/marco-scrum-metodologia-agil/

Los modelos anteriormente mencionados fueron escogidos por la cantidad de menciones en los trabajos investigativos y popularidad respecto a cursos de aprendizaje sobre estos.

Tabla 3 *Fases de los modelos de minería de datos*

<table>
<tr><th>CRISP-DM</th><th>KDD</th><th>SEMMA</th><th>TDSP</th><th>SCRUM-DS</th></tr>
<tr><td>Entendimiento del negocio</td><td>Entendimiento del dominio de la aplicación</td><td>NO APLICA</td><td>Entendimiento del negocio</td><td>Refinamiento</td></tr>
<tr><td>NO APLICA</td><td>NO APLICA</td><td>Muestreo</td><td>NO APLICA</td><td>NO APLICA</td></tr>
<tr><td>Entendimiento de los datos</td><td>Selección de los datos</td><td>Exploración</td><td rowspan="3">Adquisición y entendimiento de los datos</td><td rowspan="4">Sprint y Stand-Up diario</td></tr>
<tr><td rowspan="2">Preparación de los datos</td><td>Preprocesamiento de los datos</td><td rowspan="2">Modificación</td></tr>
<tr><td>Transformación</td></tr>
<tr><td>Modelado</td><td>Minería de datos</td><td>Modelado</td><td>Modelado</td></tr>
<tr><td>Evaluación</td><td>Evaluación e interpretación</td><td>Acceso</td><td>Aceptación del cliente</td><td rowspan="2">Retrospectiva y revisión del sprint</td></tr>
<tr><td>Despliegue</td><td>Conocimiento</td><td>NO APLICA</td><td>Despliegue</td></tr>
</table>

Como se puede visualizar en la tabla anterior, algunos modelos comparten ciertas fases y existen varias similitudes en la mayoría de estas, algunos combinan fases y otros modelos dividen ciertos trabajos.

Para determinar cuál será usado, es necesario ponderar los modelos tomando en cuenta aspectos como el tiempo, la dificultad, la aceptación, entre otros. Por lo cual se establece un sistema de valoración donde 2 representa *bueno*, 1 significa *regular* y 0 es *malo*.

Tabla 4 *Comparación cuantitativa de los modelos de minería de datos*

	CRISP-DM	**KDD**	**SEMMA**	**TDSP**	**SCRUM-DS**
Facilidad de uso	2	2	2	1	1
Detalle de sus fases	2	1	2	2	1
Popularidad de uso	2	1	0	1	0
Tiempo empleado	1	1	2	1	2
TOTAL /8	**7**	**5**	**6**	**5**	**4**

Finalizada la evaluación y comparación de cada modelo seleccionado para minar datos, se concluye que el ideal para trabajar en este proyecto de investigación es CRISP-DM. Sin embargo, dado el contexto que rodea este trabajo también es posible utilizar SEMMA y KDD oTDSP en menor medida.

A pesar de los bajos valores de SCRUM-DS, es necesario aclarar que estos modelos no son adecuados para el ambiente en el que se encuentra este proyecto, pero no se descartan para otros tipos de trabajos existentes en el ambiente laboral.

4.3. Aplicación del modelo de minería de datos para la clasificación de emails

4.3.1. Entendimiento del negocio

El primer paso para minar datos es comprender el contexto y la necesidad que existe. Para esto se debe definir el alcance, los objetivos, antecedentes, entre otros. Varios de estos puntos se han descrito en el primer capítulo del presente trabajo de titulación.

Los correos electrónicos son un medio de comunicación sumamente importante en esta era, usado con fines benignos y malignos. Los correos malignos tienen el objetivo de molestar y perjudicar a los usuarios. Estos correos malignos denominados spam producen perdidas monetarias y vulnerabilidades en la información por lo que es necesario un mecanismo de seguridad para que no lleguen al usuario final.

4.3.2. Entendimiento de los datos

El dataset escogido fue el *spam_ham_dataset* proveniente del repositorio de datos Kaggle. Para realizar la primera exploración de datos se utiliza la función *head()* de Python.

Ilustración 6 *Exploración inicial del dataset*

```
print(datasetInicial.head())
```

```
   Unnamed: 0 label                                               text  \
0         605   ham  Subject: enron methanol ; meter # : 988291\r\n...
1        2349   ham  Subject: hpl nom for january 9 , 2001\r\n( see...
2        3624   ham  Subject: neon retreat\r\nho ho ho , we ' re ar...
3        4685  spam  Subject: photoshop , windows , office . cheap ...
4        2030   ham  Subject: re : indian springs\r\nthis deal is t...

   label_num
0          0
1          0
2          0
3          1
4          0
```

Como se puede visualizar en la ilustración 6, existen 5 columnas: *unnamed: 0* que es una variable numérica discreta utilizada simplemente como un índice para asegurar que no exista registros repetidos; la variable *label* que es categórica nominal para indicar si el correo es inofensivo (*ham*) o es no deseado (*spam*); *text* es una variable categórica nominal en la que se coloca el asunto del correo y el contenido del mismo, en html o texto plano; *label_num* es una variable numérica discreta que al igual que *label* indica si el correo electrónico es inofensivo (*0*) o no deseado (*1*) de forma numérica. Seguidamente es necesario investigar si el dataset cuenta con datos nulos o vacíos para evitar complicaciones con el algoritmo. Para eso se puede usar la siguiente función:

Ilustración 7 *Revisión de datos vacíos en el dataset*

```
print(datasetInicial.isna().sum())
print(datasetInicial.shape)
```

```
Unnamed: 0    0
label         0
text          0
label_num     0
dtype: int64
(5171, 4)
```

En la ilustración 7 se visualiza que el dataset tiene un tamaño de 4 columnas con 5171 registros. Afortunadamente no existe datos vacíos o nulos, por lo que se puede proseguir con la siguiente fase de preparación de los datos.

4.3.3. Preparación de los datos

Como se explicó en el capítulo 4.1, los datos en el archivo están en el idioma inglés por lo que es necesario traducirlos a español. Para esto se utilizará una librería de Python que utiliza la API de Google Translate denominada *Googletrans*, añadiendo los correos traducidos al conjunto de datos. Y para un mejor control, se procede a renombrar las variables para que sean más entendibles.

Ilustración 8 *Traducción de inglés a español de los correos electrónicos*

```
transMail = datasetInicial['text'].apply(lambda x: traductor.translate(x, dest='es').text)

dataset = datasetInicial
dataset.insert(1, 'correo', transMail)
dataset.columns = ['index', 'correoES', 'tipo', 'mailEN', 'tipoNum']

print(dataset.head())
```

```
   index                                          correoES  tipo  \
0    605  Asunto: metanol enron; metro # : 988291\r\nest...   ham
1   2349  Asunto: hpl nom para el 9 de enero de 2001\r\n...   ham
2   3624  Asunto: retiro de neón\r\njo jo jo, estamos ce...   ham
3   4685  Asunto: photoshop, ventanas, oficina. barato ....  spam
4   2030  Asunto: re: manantiales indios\r\neste trato e...   ham

                                        mailEN  tipoNum
0  Subject: enron methanol ; meter # : 988291\r\n...        0
1  Subject: hpl nom for january 9 , 2001\r\n( see...        0
2  Subject: neon retreat\r\nho ho ho , we ' re ar...        0
3  Subject: photoshop , windows , office . cheap ...        1
4  Subject: re : indian springs\r\nthis deal is t...        0
```

Nota. De igual forma se aplica el renombramiento de las variables para que sean explícitas respecto a su contenido.

Seguidamente es necesario eliminar variables que no sean de utilidad para el resultado, en este caso se eliminará la columna de *index* debido a que no es útil para determinar si el correo es benigno o maligno, la columna *tipo* porque ya cuenta con una columna *tipoNum* que nos indica que clasificación tiene el correo y finalmente, la columna *mailEN* ya que el dataset cuenta con la columna *correosES* que está en idioma español.

Ilustración 9 *Eliminación de columnas del dataset*

```
dataset.drop('index', axis=1, inplace = True)
dataset.drop('tipo', axis=1, inplace = True)
dataset.drop('mailEN', axis=1, inplace = True)

print(dataset.head())
```

```
                                              correoES  tipoNum
0  Asunto: metanol enron; metro # : 988291\r\nest...         0
1  Asunto: hpl nom para el 9 de enero de 2001\r\n...         0
2  Asunto: retiro de neón\r\njo jo jo, estamos ce...         0
3  Asunto: photoshop, ventanas, oficina. barato ....         1
4  Asunto: re: manantiales indios\r\neste trato e...         0
```

Finalmente, para que el modelo trabaje de forma eficiente es requerido que todas las columnas se encuentren transformadas en variables numéricas. La variable *tipoNum* ya cumple con el requisito, por lo que ahora toca procesar la variable *correoES*.

Para esto es necesario el procesamiento de lenguaje natural o nlp por sus siglas en inglés. En Python existe la librería denominada *nltk* que brinda interfaces que comprende artículos, pronombres, preposiciones, entre otros componentes del lenguaje que no brindan un significado al mensaje, mas es usado para un mejor entendimiento.

Ilustración 10 *Procesamiento del lenguaje natural*

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

nltk.download('stopwords')
nltk.download('punkt')
stop_words = set(stopwords.words('spanish'))

dataset['correoES'] = dataset['correoES'].apply(lambda x: ' '.join([ word for word in word_tokenize(x)

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!

X = dataset.loc[:, 'correoES']
y =dataset.loc[:, 'tipoNum'].values

X[1]

'Asunto : hpl nom 9 enero 2001 ( ver archivo adjunto : hplnol 09. xls ) - hplnol 09 . xls'
```

Nota. Se aplica un filtro de palabras vacías y la librería *word_tokenize* permite separar palabra a palabra todas las oraciones del correo.

Como se visualiza en la ilustración 10, los correos han sido filtrados de palabras utilizadas para una mejor comprensión del contenido. Una vez aplicado esto, es necesario llevar las palabras a datos numéricos. Para hacerlo se usará una librería de *sklearn* que pondera cuanto se repite una palabra, al tener varias instancias el número será mayor por lo que tendrá un mayor peso en el modelo de aprendizaje.

Ilustración 11 *Vectorización de los correos electrónicos*

```
from sklearn.feature_extraction.text import CountVectorizer
cv=CountVectorizer()

X=cv.fit_transform(X).toarray()

X[1]

array([0, 0, 0, ..., 0, 0, 0])
```

Al comparar la ilustración 10 con la ilustración 11, se puede notar como se ha transformado una oración en un arreglo de números discretos. Una vez procesado los datos, es necesario dividirlos en un grupo de entrenamiento y otro de prueba para el modelo, siendo la proporción de 80% y 20% respectivamente, y de esta forma se procede a la creación de los algoritmos de aprendizaje.

Ilustración 12 *División de datos en grupos de entrenamiento y prueba*

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(4136, 54868)
(1035, 54868)
(4136,)
(1035,)
```

4.3.4. Modelado

Debido a que la variable a predecir solo cuenta con dos estados, se utilizará cuatro modelos de clasificación y se comparará sus resultados. Los algoritmos seleccionados fueron el clasificador de Naive Bayes, que es un clasificador que aplica el teorema de Bayes junto a distribuciones gaussianas o normales; árboles aleatorios, que trabajan de forma parecida a árboles de decisión solo que los aleatorios se combinan con el modelo de predicción de bagging; árboles de decisión, un algoritmo estructurado por nodos y ramas representando atributos y probabilidades respectivamente; y vecinos cercanos, que predice los valores por la cercanía de los atributos. Todos los modelos se encuentran dentro de la librería de *sklearn* que brinda distintos modelos de regresión y clasificación para Python.

Ilustración 13 Entrenamiento y predicción de los algoritmos escogidos

```
from sklearn.naive_bayes import GaussianNB
clasificadorNB = GaussianNB()
clasificadorNB.fit(X_train, y_train)
y_predNB = clasificadorNB.predict(X_test)

from sklearn.ensemble import RandomForestClassifier
clasificadorRF = RandomForestClassifier()
clasificadorRF.fit(X_train, y_train)
y_predRF = clasificadorRF.predict(X_test)

from sklearn.tree import DecisionTreeClassifier
clasificadorDT = DecisionTreeClassifier()
clasificadorDT.fit(X_train, y_train)
y_predDT = clasificadorDT.predict(X_test)

from sklearn.neighbors import KNeighborsClassifier
clasificadorKN = KNeighborsClassifier()
clasificadorKN.fit(X_train, y_train)
y_predKN = clasificadorKN.predict(X_test)
```

Como se puede visualizar en la ilustración 13 se utilizan los cuatro modelos anteriormente seleccionados. Estos trabajan con los parámetros predeterminados por la librería, sin embargo, es necesario mencionarlos. Para Naive Bayes “la e^{-9} parte de la varianza más grande de las variables de entrada se añaden a la varianza para estabilidad del cálculo.”; en árboles aleatorios “el número de árboles son 100, el criterio que mide la impureza de distribución de datos es *Gini*, no tiene límite de profundidad, el mínimo de hojas es 0 y no hay límite para un máximo.”; el árbol de decisión “tiene un criterio de impureza tipo *Gini*, no tiene un límite de profundidad, el mínimo de hojas es 0 y no hay límite para un máximo.”; y vecinos cercanos “que utiliza 5 vecinos, todos los vecinos tienen el mismo peso y utiliza la distancia *Minkowski*” (scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation, s. f.).

4.3.5. Evaluación

De igual forma que los modelos, la librería *sklearn* cuenta con funciones de métricas para medir la eficacia de los algoritmos. Al estar tratando con modelos de clasificación, es posible utilizar métricas como la matriz de confusión que permite ver que datos acertó y cuantos fueron falsos positivos o falsos negativos.

Las métricas utilizadas son: la precisión, que ayuda a comprender que porcentaje de predicciones positivas son correctas; la sensibilidad, que indica la cantidad de casos positivos predichos; la especificidad, que indica la cantidad de casos negativos predichos; y la exactitud, que brinda la cantidad de predicciones correctas. Para calcular todas las métricas se usan las siguientes fórmulas:

$$Precisión = \frac{TP}{(TP + FP)} * 100$$

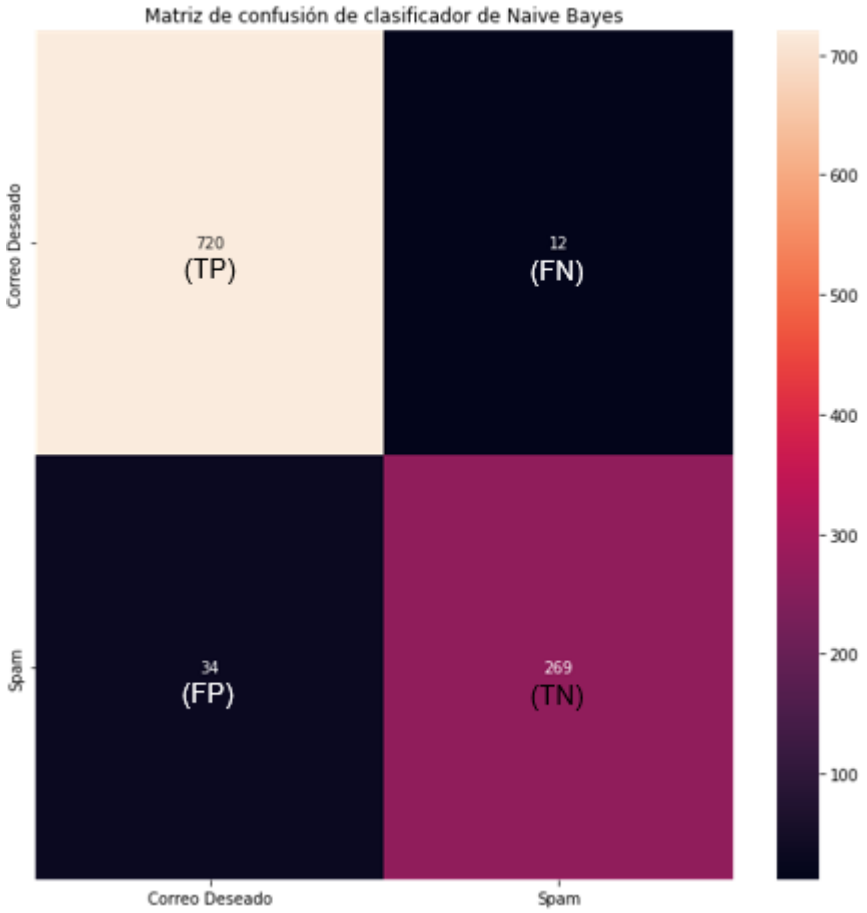
$$Sensibilidad = \frac{TP}{(TP + FN)} * 100$$

$$Especificidad = \frac{TN}{(TN + FP)} * 100$$

$$Exactitud = \frac{(TP + TN)}{(TP + FP + FN + TN)} * 100$$

Evaluando el modelo de Naive Bayes Gaussiana, se utiliza una matriz de confusión y a partir ella se mide la exactitud que tuvo.

Ilustración 14 *Matriz de confusión del clasificador de Naive Bayes*

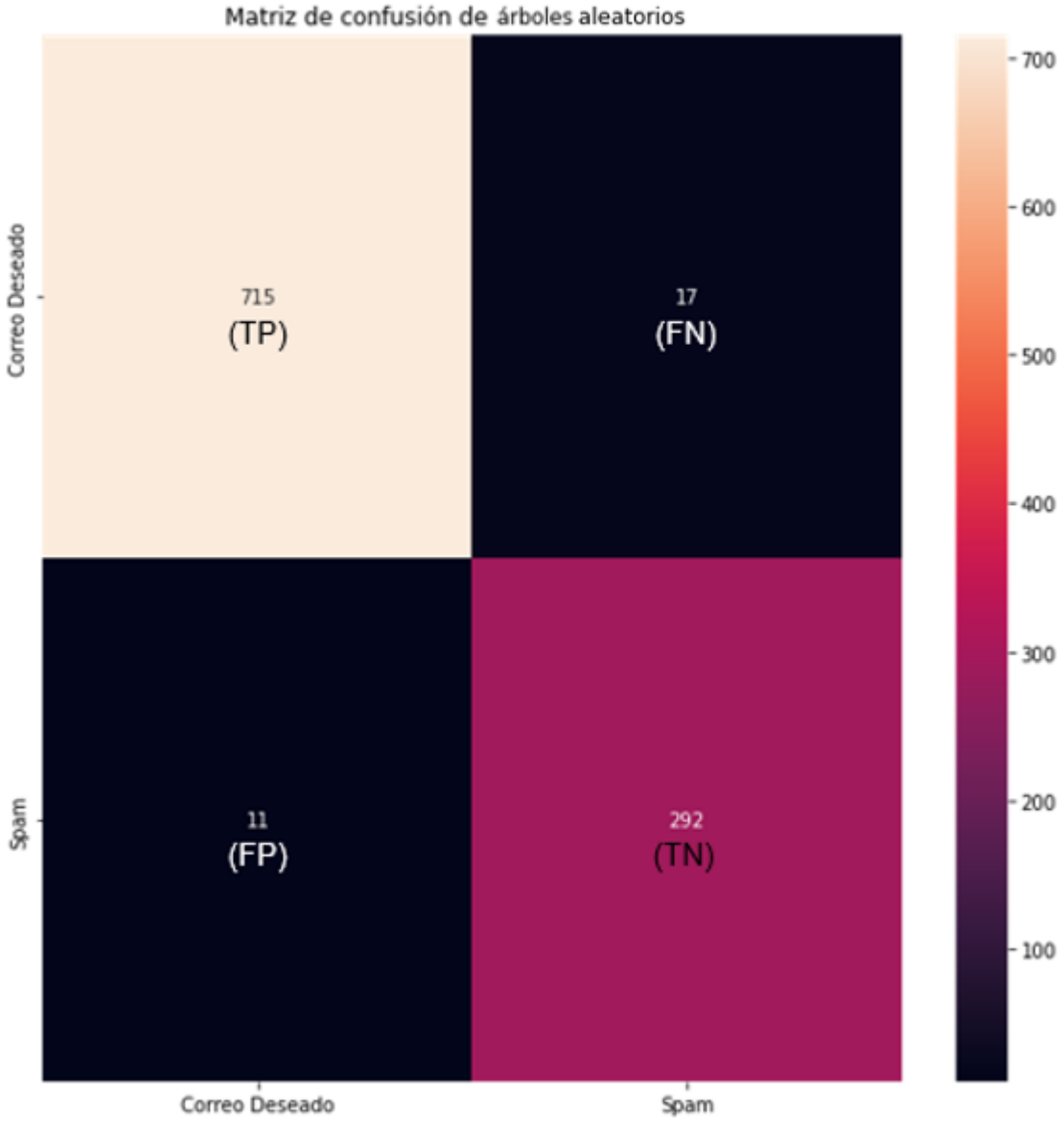


Para entender la ilustración 14 y calcular la precisión es necesario comprender individualmente los valores. En el eje X de la ilustración se encuentra la clasificación del algoritmo, mientras que en el eje Y son las clasificaciones reales. Los correos deseados que fueron predichos como deseados son verdaderos positivos (*TP*), los correos deseados que fueron predichos como spam adquieren el nombre de falso negativo (*FN*), correos que sean spam, pero su predicción fue de deseados son falsos positivos (*FP*) y correos spam que sean predichos como spam son verdaderos negativos (*TN*).

Al reemplazar los valores de la fórmula, se obtiene una precisión de 72.8%, una sensibilidad de 98.36%, una especificidad de 88.78% y una exactitud de 95.56%. Estos valores son altos para el modelo, por lo que ha funcionado de forma correcta el algoritmo de Naive Bayes Gaussianas.

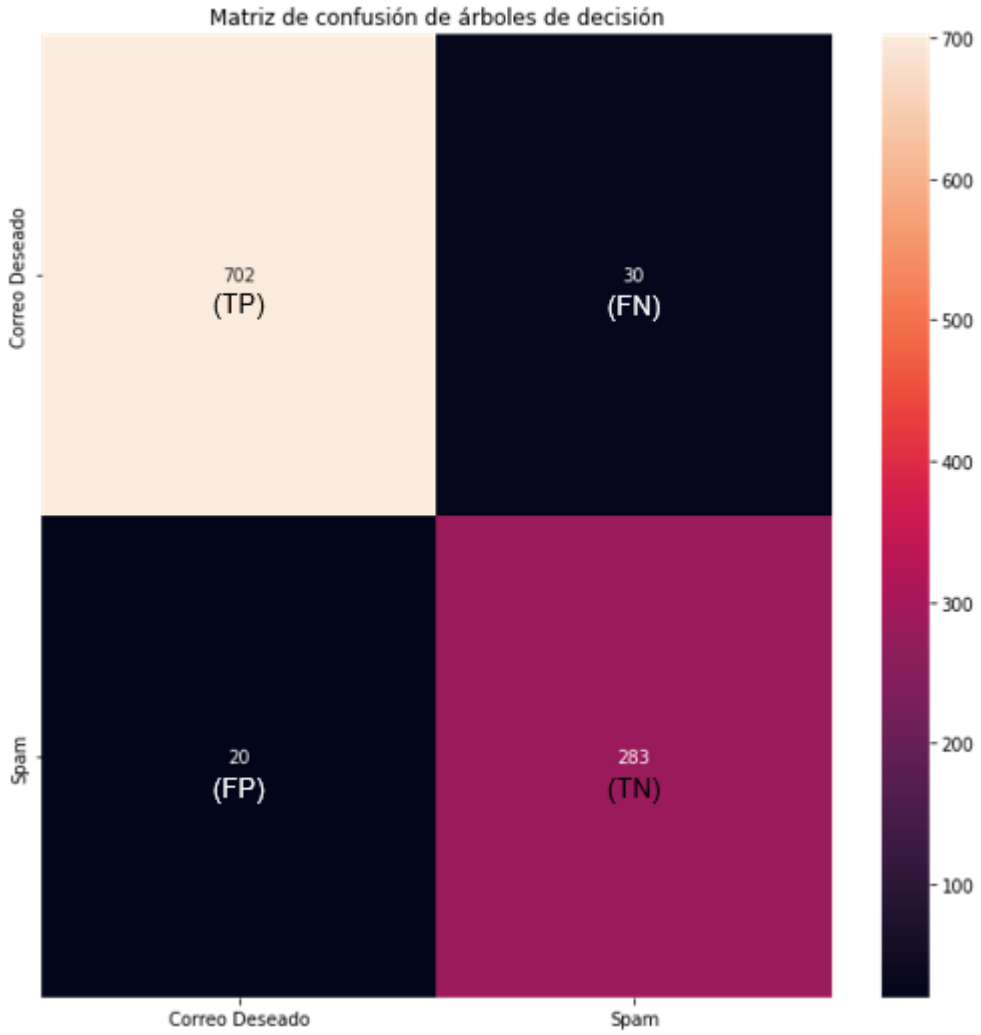
De igual forma, se saca la matriz de confusión del modelo de árboles aleatorios.

Ilustración 15 *Matriz de confusión de árboles aleatorios*



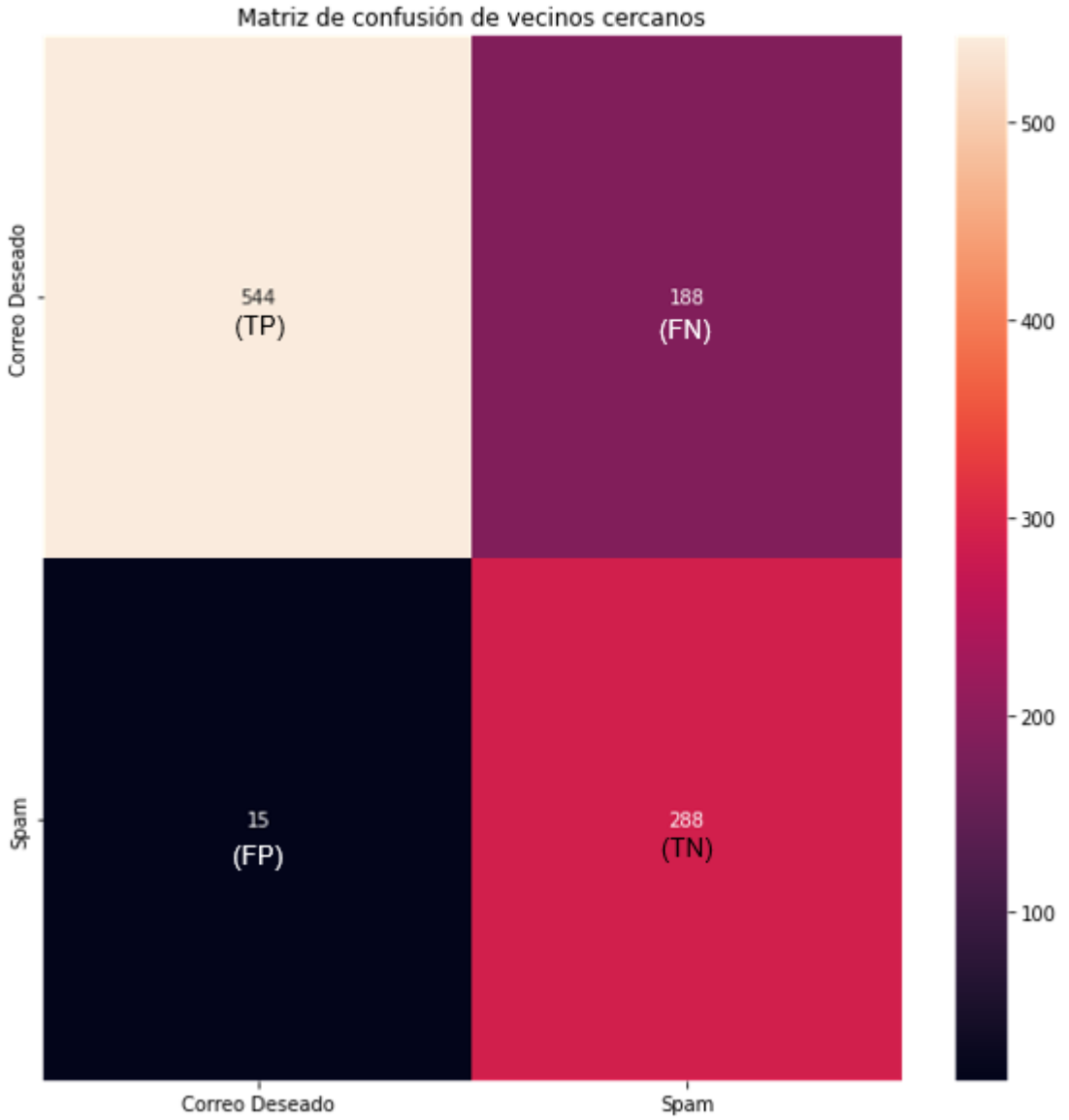
Las indicaciones dadas en la ilustración 14 se mantienen en la ilustración 15. Por lo que, el siguiente paso es calcular sus métricas. Sus resultados son de una precisión de 98.49%, una sensibilidad de 97.68%, una especificidad de 96.37% y una exactitud de 97.3%. Lo siguiente es medir el rendimiento del modelo de árboles de decisión.

Ilustración 16 *Matriz de confusión de árboles de decisión*



Al igual que en los otros casos lo que se debe realizar es el cálculo de las métricas. Árboles de decisión logró una precisión de 97.23%, una sensibilidad de 95.9%, una especificidad de 93.4% y una exactitud de 95.17%. Finalmente, hay que evaluar al modelo de vecinos cercanos.

Ilustración 17 *Matriz de confusión de vecinos cercanos*



De igual forma, por medio de la matriz de confusión se obtiene los resultados en las métricas selectas. Vecinos cercanos logró una precisión de 97.32%, una sensibilidad de 74.32%, una especificidad de 95.05% y una exactitud de 80.39%.

Para comprender que algoritmo rinde mejor es necesario tabular la información para una comparativa clara.

Tabla 5 *Comparación de resultados de los modelos de aprendizaje supervisado*

	Naive Bayes	**Árboles Aleatorios**	**Árboles de Decisión**	**Vecinos Cercanos**
Precisión	72.8%	**98.49%**	97.23%	97.32%
Sensibilidad	**98.36%**	97.68%	95.90%	74.32%
Especificidad	88.78%	**96.37%**	93.40%	95.05%
Exactitud	95.96%	**97.30%**	95.17%	80.39%

Nota. Los valores con **negrillas** representan al valor más alto adquirido en la métrica.

Como se muestra en la tabla 5, a pesar de que todos los modelos tienen un valor deseable de exactitud, árboles aleatorios es que sobresale por tener los mejores valores en la mayoría de las métricas. Por este motivo, en este caso, aplicar este modelo es lo recomendable.

4.3.6. Despliegue

Para poder aplicar el conocimiento generado con los modelos de aprendizaje de máquina se puede aplicar el código por medio de APIs, en el caso de utilizar Gmail de Google, la compañía tiene un servicio para poder implementar scripts a la cuenta de Google. De igual forma, si se cuenta con un correo suministrado por Microsoft se utilizaría la API de Outlook. Finalmente, para otros tipos de correo es necesario, al igual que con los casos anteriores, utilizar una API desarrollada desde el principio para ejecutar el algoritmo y que filtre los correos de forma casi perfecta.

CONCLUSIONES

- Se uso las fuentes de datos de tipo públicas debido a que son útiles para fines educativos, mientras que las de orden privado se desarrollan en un ambiente laboral por estar bajo políticas de uso de una organización, así mismo, la utilización de datos personales puede ser complicado por la ínfima cantidad de datos que había.
- El modelo de minería de datos CRISP-DM fue el seleccionado por su facilidad y popularidad de uso como también por el nivel de detalle en sus seis fases, a pesar de eso, también era posible utilizar SEMMA por como describe sus fases, la facilidad que tienen y la cantidad de tiempo que utilizan, su defecto siendo que no es tan utilizado.
- Entre todos los algoritmos, el de mejor rendimiento fue árboles aleatorios, seguido por árboles de decisión que pueden los modelos implementados, mientras que el algoritmo de vecinos cercanos fue el peor de todos.
- Existen diversas fuentes de datos de distinto orden y con distintas características, por lo que un análisis a profundidad es necesario ya que brinda información útil para aplicar el modelo minería de datos.
- La longevidad de la minería de datos ha permitido que se desarrollen distintas metodologías para realizar el trabajo. Para seleccionar el que mejor se ajuste, es necesario estudiar el contexto como los datos que se trabajarán, el tiempo disponible, el presupuesto, entre otros valores.
- Al igual que las metodologías de minería de datos, existen diversos modelos de aprendizaje de máquina, cada uno dispone de ciertas características. Cada modelo tiene fortalezas y debilidades por lo que se debe conocer en qué casos se aplica cada algoritmo.

RECOMENDACIONES

- Reunir varios datos de orden personal para utilizarlos con el modelo, de forma que se compare el impacto que tiene sobre los modelos.
- Realizar un caso de estudio con modelos que se centren en el trabajo con el cliente como TDSP o SCRUM-DS aplicando lo aprendido en un ambiente laboral.
- Implementar diversos modelos de aprendizaje de máquina por medio de APIs para la diferenciación de los resultados.
- Integrar distintas fuentes de datos, generando un almacén de datos para que exista un incremento de valor en los resultados generados. Así mismo, se simula un ambiente laboral dado a que los datos de una empresa suelen provenir de distintas áreas.
- Trabajar de forma cercana con el cliente para conseguir un mejor contexto respecto a diversos ámbitos que rodeen el trabajo. En caso de no disponer de un cliente, simular uno haciendo un levantamiento de requerimientos.

BIBLIOGRFÍA

¿Qué es la minería de datos? (2021, 13 enero). latam.kaspersky.com. Recuperado 1 de octubre de 2022, de https://latam.kaspersky.com/resource-center/definitions/data-mining

Baijens, J., Helms, R. & Iren, D. (2020, junio). Applying Scrum in Data Science Projects. *2020 IEEE 22nd Conference on Business Informatics (CBI)*. https://doi.org/10.1109/cbi49978.2020.00011

Barish, A. (2020). spam_ham_dataset [Conjunto de datos]. En *Kaggle* (Versión V4). https://www.kaggle.com/code/ayhampar/spam-ham-dataset

Ciberataque a Banco Pichincha fue realizado por atacantes internacionales, se revela en Comisión de Desarrollo Económico. (2021, 20 octubre). *El Universo*. Recuperado 25 de septiembre de 2022, de https://www.eluniverso.com/noticias/economia/ciberataque-a-banco-pichincha-fue-realizado-por-atacantes-internacionales-se-revela-en-comision-de-desarrollo-economico-nota/

Cormack, G. V. (2008). Email Spam Filtering: A Systematic Review. *Foundations and Trends® in Information Retrieval*, *1*(4), 335-455. https://doi.org/10.1561/1500000006

Díaz, V. (2021, 17 julio). CNT apaga todas sus computadoras tras fuerte ataque informático. *El Comercio*. Recuperado 25 de septiembre de 2022, de https://www.elcomercio.com/actualidad/negocios/cnt-ataque-informatico-ransomware-fiscalia.html

Herrero Sanz, P. (2019). *Malware lab* [Trabajo final de grado, Universidad de Barcelona]. Dipòsit Digital.

Hopkins, M., Reeber, E., Forman, G. & Suermondt, J. (s. f.). Spambase Data Set [Conjunto de datos]. En *UCI Machine Learning Repository*. https://archive.ics.uci.edu/ml/datasets/spambase

Jamsa, D. (2020, 17 febrero). *Introduction to Data Mining and Analytics*. Jones & Bartlett Publishers.

Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K. & Alazab, M. (2019). A Comprehensive Survey for Intelligent Spam Email Detection. *IEEE Access*, *7*, 168261-168295. https://doi.org/10.1109/access.2019.2954791

Lozada, J. (2014). Investigación Aplicada: Definición, Propiedad Intelectual e Industria. *CienciAmérica*, *3*(1), 47-50. https://dialnet.unirioja.es/servlet/articulo?codigo=6163749

Martinez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernandez-Orallo, J., Kull, M., Lachiche, N., Ramirez-Quintana, M. J. & Flach, P. (2021, 1 agosto). CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories. *IEEE Transactions on Knowledge and Data Engineering*, *33*(8), 3048-3061. https://doi.org/10.1109/tkde.2019.2962680

Muhammad Abdulhamid, S., Shuaib, M., Osho, O., Ismaila, I. & K. Alhassan, J. (2018, 8 enero). Comparative Analysis of Classification Algorithms for Email Spam Detection. *International Journal of Computer Network and Information Security*, *10*(1), 60-67. https://doi.org/10.5815/ijcnis.2018.01.07

Novaković, J. D., Veljović, A., Ilić, S. S., Papić, Z. & Tomović, M. (2017). Evaluation of Classification Models in Machine Learning. *Theory and Applications of Mathematics & Computer Science*, *7*(1), 39-46. https://uav.ro/applications/se/journal/index.php/TAMCS/article/view/158

Pathak, V. & Jena, B. (2013). Qualitative research. *Perspectives in Clinical Research*, *4*(3), 192. https://doi.org/10.4103/2229-3485.115389

scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation. (s. f.). https://scikit-learn.org/

Shafique, U. & Qasier, H. (2014, noviembre). A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, *12*(1), 217-222. http://www.ijisr.issr-journals.org/abstract.php?article=IJISR-14-281-04

V. Metsis, I. Androutsopoulos & G. Paliouras. (2006). Enron1 [Conjunto de datos]. En *AUEB*. http://www.aueb.gr/users/ion/data/enron-spam/preprocessed/enron1.tar.gz

What is the Team Data Science Process? - Azure Architecture Center. (s. f.). Microsoft Learn. Recuperado 6 de octubre de 2022, de https://learn.microsoft.com/en-us/azure/architecture/data-science-process/overview