

Tecnológico de Monterrey, Campus Guadalajara



Desarrollo de Proyectos de Ingeniería Matemática

---

## **Grayproject: Fill missing data based in semantic analysis**

---

June 3, 2023

Ingeniería en Ciencia de Datos y Matemáticas

José Iván Álvarez Güémez A01638794

Ricardo Kaled Corona Romero A01635604

Renata Uribe Sánchez A01274629

Rodrigo Hiroshi Argandar Nishizawa A01630005

## Contents

<b>1</b>	<b>Introducción</b>	<b>3</b>
1.1	Definición del problema . . . . .	3
1.2	Herramientas utilizadas . . . . .	3
<b>2</b>	<b>Fase 1: Aspectos generales de la planeación</b>	<b>3</b>
2.1	Metodologías ágiles . . . . .	4
2.1.1	Objetivo del proyecto . . . . .	4
2.1.2	Diagrama de Gantt . . . . .	5
2.1.3	Alcance . . . . .	5
2.1.4	Riesgos . . . . .	5
2.1.5	Stakeholders . . . . .	6
<b>3</b>	<b>Fase 2: Análisis exploratorio, pre-procesamiento y limpieza del corpus.</b>	<b>6</b>
3.1	Análisis exploratorio . . . . .	6
3.2	Limpieza del corpus . . . . .	8
<b>4</b>	<b>Fase 3: Desarrollo de metodología, modelación y realización de código.</b>	<b>9</b>
4.1	Desarrollo de la metodología . . . . .	9
4.2	Modelación . . . . .	10
4.2.1	Word embedding . . . . .	10
4.2.2	Decision Tree . . . . .	10
4.2.3	Support Vector Classifier . . . . .	11
4.2.4	K-nearest Neighbor . . . . .	11
4.2.5	Clustering (K means) . . . . .	11
4.3	Entrenamiento . . . . .	11
<b>5</b>	<b>Fase 4: Validación y presentación de resultados</b>	<b>12</b>
<b>6</b>	<b>Conclusiones</b>	<b>15</b>
6.1	Trabajos futuros . . . . .	15
<b>7</b>	<b>Anexos</b>	<b>17</b>
7.1	Código fuente . . . . .	17
7.2	Tablero Miró . . . . .	17

## 1 Introducción

La avasallante era tecnológica en la que nos desenvolvemos hoy en día supone distintos desafíos y disyuntivas éticas a partir de los avances revolucionarios que se generan progresivamente de la mano de la inteligencia artificial y otras tecnologías emergentes. Por un lado representa un indicador de prosperidad por su apresurada capacidad para resolver problemas pero en contraria se considera también un motivo de preocupación, pues se teme que este inmensurable desarrollo tecnológico eventualmente genere dilemas éticos al incorporar una concepción errónea del conocimiento. Tal es el caso de los modelos del lenguaje, programas ahora considerados como los primeros retazos hacia la inteligencia artificial general que parten del procesamiento del lenguaje natural, un campo de estudio que pretende estudiar la interacción entre el lenguaje humano y las computadoras al comprender y generar texto.[\[4\]](#)

En los últimos años, el procesamiento del lenguaje natural (NLP) ha representado avances importantes en el campo de la investigación de marketing, al ser utilizado para extraer información de un conjunto de textos, sobretodo para problemas que involucran la clasificación textual de datos relativos a la publicidad, lo que podría significar publicaciones en redes sociales, comentarios de clientes y campañas de marketing. [\[7\]](#)

### 1.1 Definición del problema

Como una de las más importantes empresas tecnológicas, HP, representa una vasta línea de productos y servicios operando alrededor del mundo. Para comercializar eficazmente estos productos, es necesario comprender las necesidades del mercado y contrastarlas con diversas tendencias estacionales de modo que puedan generarse ofertas y promociones para los distintos segmentos de clientes. Al tratarse de una tarea que trabaja directamente con registros de texto, el NLP podría implementarse para encontrar patrones en las diversas bases de datos relativas al marketing de la empresa.

Es así que el presente estudio explora el uso del NLP para clasificar los programas de marketing de HP. A través del uso de técnicas relacionadas a modelos del lenguaje podrían detectarse patrones en función a palabras asociadas con la categoría o incluso dentro de la misma descripción.

### 1.2 Herramientas utilizadas

Dentro del desarrollo del proyecto, con la intención de facilitar trabajo colaborativo y el control de versiones en términos de la confección del código, se utilizó *Google Colaboratory* de modo que todos los integrantes del equipo tuvieran acceso en tiempo real a todas las modificaciones del código. Finalmente, una vez concluida la fase de modelación comprendida por la creación del código, las diversas libretas del código fuente se guardaron en *Git Hub* (Ver [7](#) para una versión completa del código)

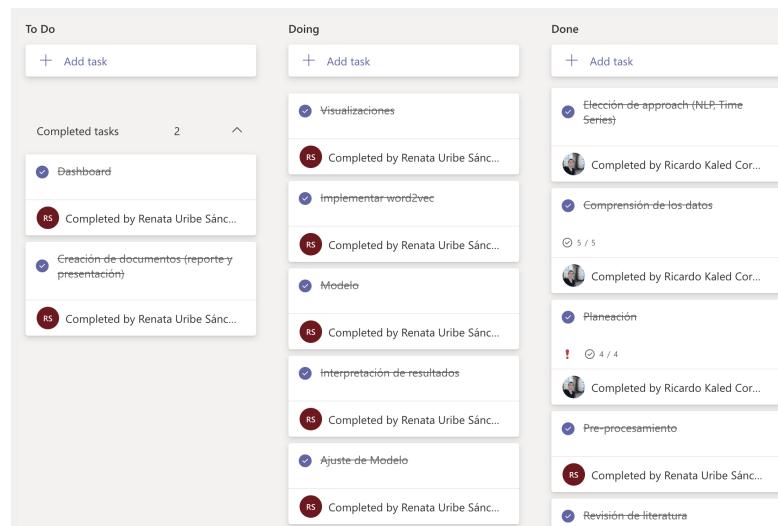
Por su parte, dentro de la creación del código se utilizaron diversas librerías propias del lenguaje de programación python, siendo las más importantes *Scikit-learn*, *NLTK* (concretamente para el procesamiento del lenguaje natural), *Gensim* en la vectorización de palabras, *Regex* para la búsqueda específica de expresiones y finalmente la herramienta *Tableau* para la creación de visualizaciones dentro de un primer análisis exploratorio.

## 2 Fase 1: Aspectos generales de la planeación

Con la tentativa de guiar el desarrollo del proyecto hacia una gestión correcta en términos de repartición de tareas y fechas de entrega para la obtención de resultados efectivos, se incorporaron prácticas de *Project Management* previas al análisis de los datos, de modo que la planeación siguió la estructura de la metodología ágil *Scrum*.

## 2.1 Metodologías ágiles

Partiendo de la metodología *Scrum* [6], el flujo de trabajo comenzó con la definición del *Product Backlog*, enlistando todas las funcionalidades y características que se deseaban incluir en la entrega final. En función de las características se crearon actividades y posteriormente se asignaron a cada uno de los miembros del equipo con fechas de entrega que se alinearan a las fechas clave, correspondientes a los avances y a la entrega final del proyecto. Toda comunicación interna del equipo se generó a través de la plataforma *Teams* además del tablero Scrum con todas las actividades que comprendieron el proyecto y su estado de progreso. Adicionalmente a estas herramientas, el registro de todo progreso se plasmó en el tablero Miró relativo al Equipo 4 (Véase 7). En la Figura 1 se observa un ejemplo del *Scrum Board* en proceso con algunas de las actividades que comprendieron el desarrollo del proyecto



**Figure 1:** Ejemplo del Scrum Board en proceso dentro del desarrollo del proyecto

### 2.1.1 Objetivo del proyecto

Desarrollar un modelo de clasificación que permita automatizar el proceso, utilizando técnicas de procesamiento del lenguaje natural para identificar el tipo de programa de marketing de un producto determinado en el que se invierte el capital, con el fin de reducir la dependencia de la clasificación manual, mejorando la eficiencia y precisión del proceso.

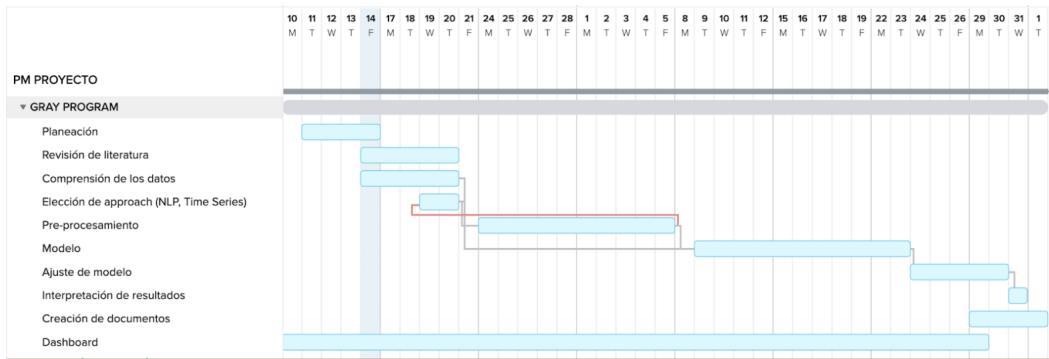
A continuación se observa la Figura 2 en donde se describen concretamente los objetivos de este estudio siguiendo los criterios SMART.

Objetivos SMART				
Específico (S)	Medible (M)	Alcanzable (A)	Relevante (R)	Tiempo (Ti)
Desarrollar un modelo de clasificación automatizado que permita identificar el tipo de programa de marketing.	Evaluar la precisión del modelo mediante una métrica de clasificación, además de la precisión, el f1-score.	Se tiene acceso directo a la base de datos además de los conocimientos necesarios gracias a las clases complementarias.	Será relevante para la organización pues ya no existirá dependencia a la clasificación manual.	Dentro de las próximas ocho semanas se implementará el modelo de clasificación.

**Figure 2:** Objetivos SMART

### 2.1.2 Diagrama de Gantt

Partiendo de Scrum para nuestra gestión del proyecto, se realizó un Diagrama de Gantt en donde fue posible visualizar los tiempos de entrega así como la secuenciación de tareas con su respectiva dependencia para comenzar con una nueva actividad. (Véase Figura 3)



**Figure 3:** Diagrama de Gantt

### 2.1.3 Alcance

Los siguientes alcances del proyecto se basan en las principales actividades y resultados que se esperan alcanzar a lo largo del desarrollo de este estudio, enfocándonos en generar una solución automatizada y precisa para la clasificación de programas de marketing.

1. Desarrollar un modelo de clasificación utilizando técnicas de procesamiento del lenguaje natural (NLP) para automatizar el proceso de asignación de categorías a los programas de marketing de HP de modo que se imite el proceso manual al reducir su dependencia.
2. Utilizar una limpieza adecuada del corpus de texto para aumentar la precisión de los modelos implementados a través de librerías como Regex y NLTK, además de eliminar stop words y aplicar algún proceso de lematización.
3. Seguir una metodología basada en MLops para asegurar la efectividad del modelo y la conclusión del proyecto.
4. En términos de entregables, realizar un tablero interactivo en Miró que registre el avance del proyecto así como un reporte final que documente la metodología y herramientas utilizadas.
5. Exponer los resultados del proyecto a través de una presentación ejecutiva con el socio formador.

### 2.1.4 Riesgos

Dentro de la realización de un proyecto, es importante identificar y gestionar riesgos de manera proactiva, es decir, con la certeza de contar con un plan de acción en caso de que algún incidente se presente. A continuación, se enumeran algunos de los posibles riesgos que podrían impactar en el proyecto:

1. **Datos incompletos** Los datos podrían no estar completos, con errores de clasificación y falta de coherencia, lo que podría dificultar la evolución del proyecto. De la misma forma, las diversas clases podrían no estar balanceadas, por ejemplo, con muchas más ocurrencias en alguna en particular.
2. **Técnicas de NLP** Debido a que el conjunto de datos es de una naturaleza técnica, es importante considerar su confección para la selección de técnicas de procesamiento del texto pues su contenido

está estructurado con códigos de productos y palabras claves, es decir, carece de enunciados completos, lo que podría dificultar su manipulación

3. **Sobreajuste del modelo** Después de la implementación del modelo, el algoritmo podría resultar en un sobreajuste de los datos de entrenamiento, de modo que no sea capaz de generalizar para nuevas observaciones, por ello, sería necesario realizar una validación adecuada.
4. **Falta de comunicación** Si no se establece una comunicación efectiva con el socio formador, los profesores encargados y entre los miembros del equipo, se podría incidir en una mala alineación de los objetivos.

#### 2.1.5 Stakeholders

Los principales involucrados en el presente proyecto son los profesores encargados del proyecto, quienes brindarán orientación y apoyo en el desarrollo del modelo de clasificación. Además, Iván, el representante de HP, juega un rol fundamental, ya que además de proporcionara los datos, nos dará una constante retroalimentación del proyectos para que pueda alinearse con sus expectativas.

### 3 Fase 2: Análisis exploratorio, pre-procesamiento y limpieza del corpus.

#### 3.1 Análisis exploratorio

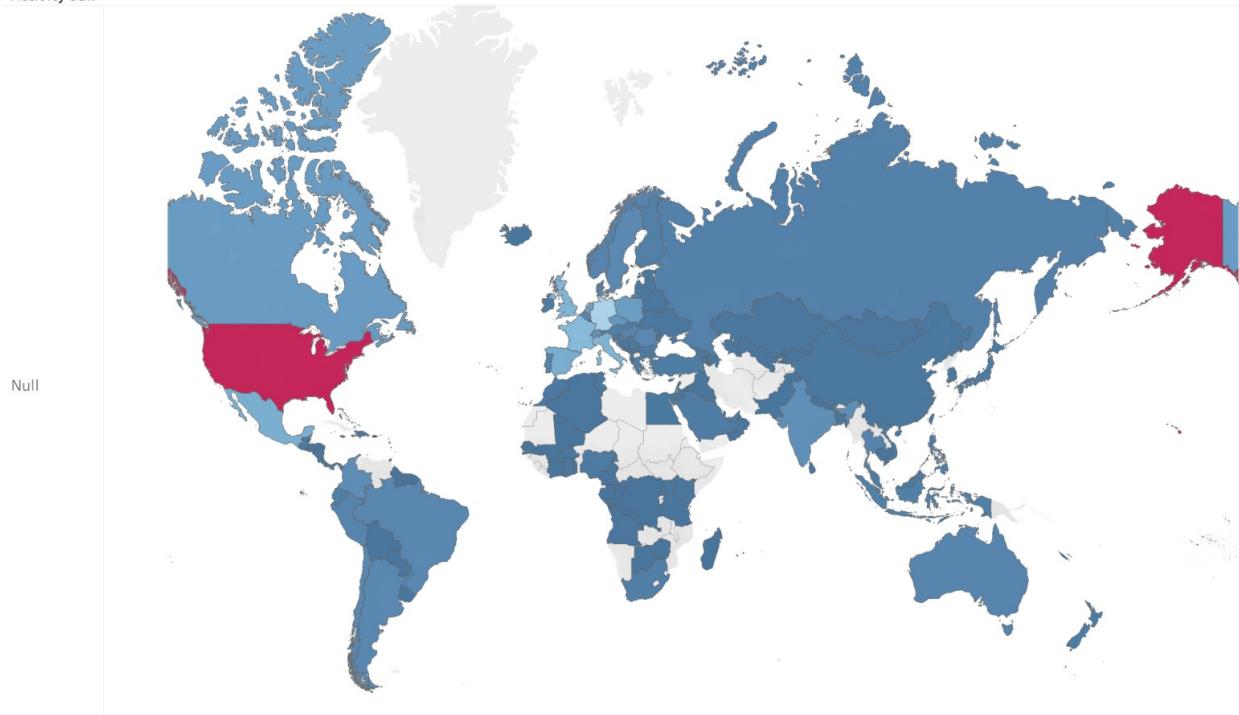
Los gastos de las campañas de marketing de HP son capturados en una tabla junto con la descripción del programa en donde el flujo de gastos se calcula agrupando las campañas por categorías de programa y subprograma. La asignación de la categoría requiere intervención humana por lo que se busca imitar la inteligencia humana en el momento de asignación de la categoría. Este dataset que fue proporcionado contaba con distintas columnas que daban insights de como se llevaba este control, desde la fecha donde empezaba, el modelo de ciertos productos, el tipo de actividad, el subtipo de las mismas entre otros. Al tener tanta diferencia entre los datos almacenados, los datos tenian celdas que contaban con valores nulos o desconocidos, lo cual era importante para el socio formador arreglar mediante la implementación de este proyecto.

Al obtener el dataset que se pretendía analizar, se hicieron distintas visualizaciones para obtener la cantidad de valores que tendrían que limpiarse o removerse, al igual que analizar cómo estaba comprendido el conjunto de datos, corrigiéndolo antes de comenzar con la modelación.

Para comprobar dónde estaba originado, se hizo una búsqueda rápida con la intención de saber en qué países se ubicaba la mayor cantidad de datos desconocidos en el mundo, donde finalmente se localizó que EE.UU. era el mayor problema, justificando así la raíz del problema para clasificar los datos faltantes, como puede observarse en la Figura 4.

## Activity Subtype Null by country

Activity Su..

**Figure 4:** Cantidad de datos desconocidos por país

La primera visualización creada fue para conocer la cantidad de datos para cada *activity subtype*, variable dependiente que debía de limpiarse.

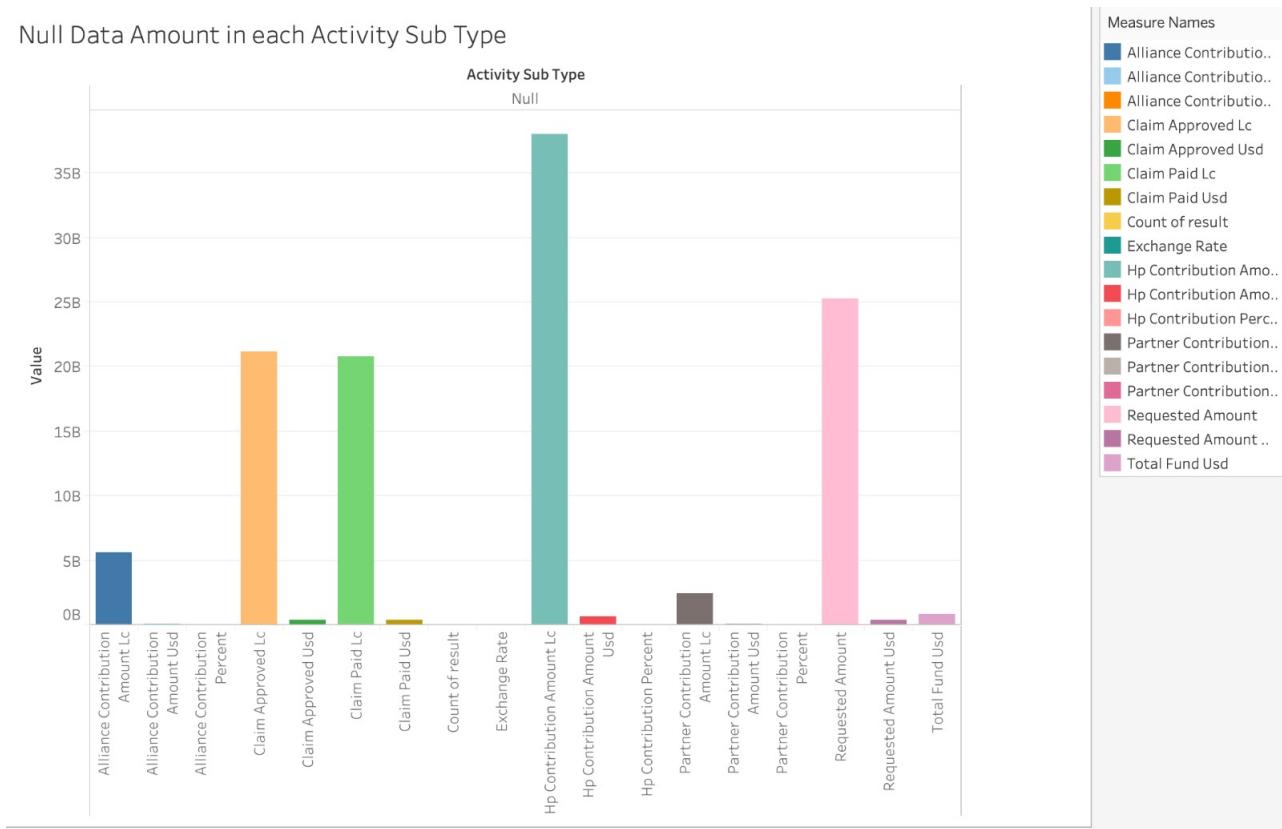
Tras obtener este número, se visualizó la distribución de estos valores "UNKNOWN" para ver dónde se encontraba la mayoría de estos (Figura 5).

Activity Subtype count

Activity Subtype Id	
Advertising	1,721
Amplification	27
Banner	284
Billboard	151
Black_Friday	151
Brand_Building	29
BTB	1,006
BTS	688
Choice	30
Circular	85
Digital	156
Display	1,314
Email	304
Endcap	179
Holiday	618
Landing_page	7
Launch	7
Other	3,814
pla	6
Prime	8
Print	51
Rewards	52
Search	533
Showcase	175
Social	20
Sponsored_product	333
TIN	31
UNKNOWN	9,679
Vendor	69
Video	1

**Figure 5:** Count for each Activity Subtype

Null Data Amount in each Activity Sub Type

**Figure 6:** Unknown values for each activity subtype

En las visualizaciones anteriores (Figura 5 y Figura 6) se denota como hay bastante variedad de datos desconocidos en las diferentes categorías que tenemos dentro del conjunto de datos. Con estos conceptos en mente se generó la limpieza de estos datos para la implementación que se describe posteriormente en este documento.

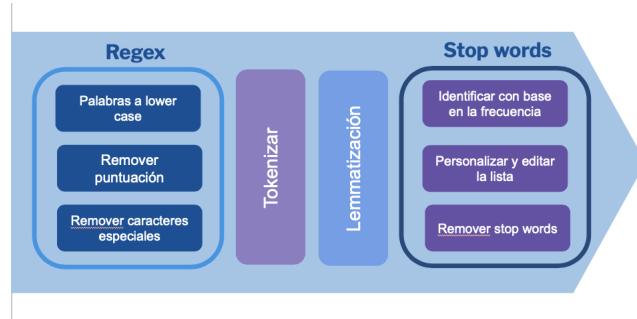
### 3.2 Limpieza del corpus

Previo al proceso de limpieza propio del procesamiento del lenguaje natural, el conjunto de datos también se tuvo que procesar en términos de datos nulos. En el caso particular de la descripción, cuando existían observaciones vacías, se optó por tomar como descripción el valor de la columna *Name*, de modo que, en los datos vacíos existían dos mismas observaciones. Esto no supuso ningún problema para el modelo de vectorización pues al obtener el promedio de la oración para explicar su contexto seguía formándose una combinación lineal de aquel vector, en otras palabras, su dirección no era afectada.

Para comenzar el proceso de limpieza de los datos, después de que seleccionaron las columnas *Program Name* y *Program Description* con la tentativa de comprender el corpus, se convirtió el conjunto de datos a minúsculas, lo que facilitó su manejo y análisis. A continuación, se procedió eliminando la puntuación y empleando la librería de expresiones regulares *Regex* para remover caracteres especiales, tales como URLs.

Una vez con el conjunto de datos limpio de caracteres innecesarios, se procedió a crear tokens, dividiendo el corpus por palabras, y se optó por utilizar la lematización en lugar de la stemmatización. La razón detrás de esta elección es debido a que en un primer intento de utilizar la stemmatización, algunas palabras resultaron cortadas a la mitad debido a su naturaleza basada en reglas, lo que resultó en una precisión deficiente. En cambio, con la lematización, la limpieza del texto mejoró significativamente al reducir correctamente su extensión a su lema, lo que significó que se eliminaron prefijos y sufijos de manera acertada. En cuanto a las *stop words*, se decidió personalizar la lista y no limitarse a utilizar únicamente las incluidas en los paquetes de la librería NLTK.

Para identificar las *stop words* más relevantes, se analizó su frecuencia en el conjunto de datos y se determinó que las más comunes eran menos relevantes siendo preposiciones o artículos en su mayoría, por lo tanto, fueron eliminadas. Posteriormente, se revisó la lista de *stop words* para realizar una lista de exclusión y se incluyeron algunas palabras que, aunque consideradas como palabras sin significado, eran importantes para nuestro análisis. Una vez con ambas listas, se procedió a eliminar las *stop words* y se obtuvo un conjunto de datos limpio y listo para ser analizado. En la Figura 7 puede observarse la metodología recién descrita.



**Figure 7:** Metodología de la limpieza del corpus

A continuación (Figura 8) se presenta un ejemplo del conjunto de datos limpio con las palabras tokenizadas.

	index	Name	Descr	Target
0	20494	['asp', 'meijer', 'bts', 'endcap']	['market', 'development']	BTS
1	23268	['costco']	['costco']	Display
2	23440	['costco']	['costco']	Display

**Figure 8:** Primeras ocurrencias del corpus limpio

## 4 Fase 3: Desarrollo de metodología, modelación y realización de código.

### 4.1 Desarrollo de la metodología

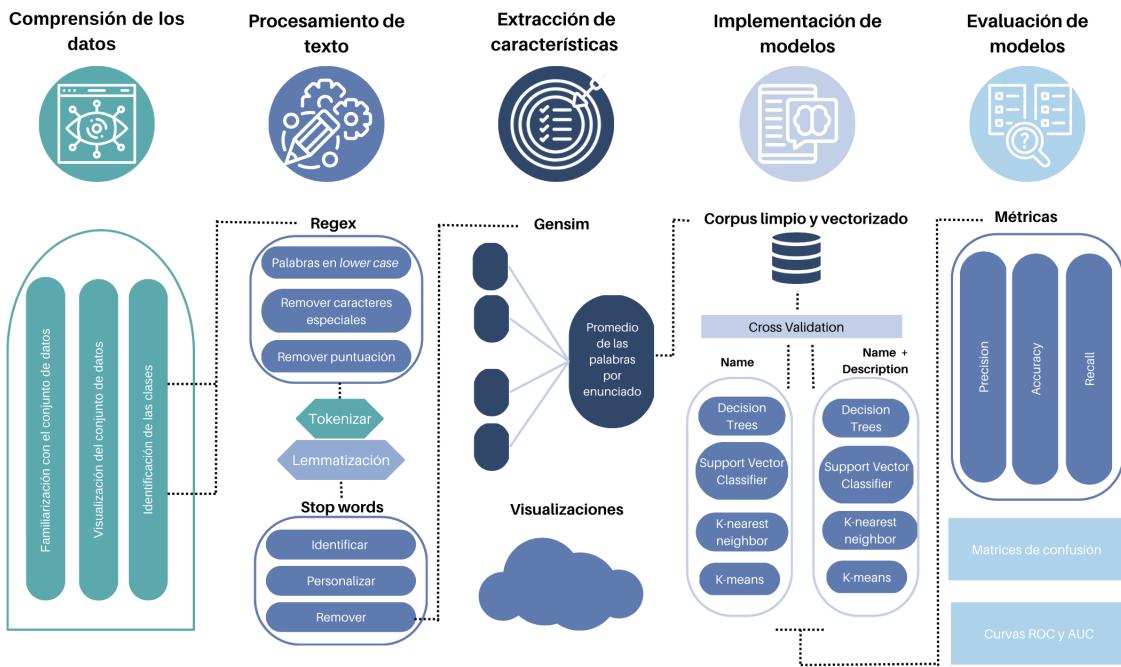
Previo a la construcción de modelos se plantearon dos hipótesis para conducir y posteriormente evaluar el desempeño del proyecto, concretamente de la clasificación.

**Hipótesis 1:** Se plantea que la inclusión de la columna *Description* junto con la columna *Name* en el modelo, resultará en una mejora significativa para la precisión de la clasificación. Se espera que, al incorporar más columnas, brindando información adicional para la descripción del corpus, se permita la captura de mejores características del contexto y finalmente una clasificación más acertada.

**Hipótesis 2:** Se postula que aplicar un enfoque de limpieza de texto más completo, que además de incluir el uso de expresiones regulares (regex), también considere la eliminación de stopwords y la lematización, resultará en un mejor rendimiento del modelo en comparación con un enfoque que solo utilice expresiones regulares. La eliminación de palabras vacías y la reducción de estas a su lema original ayudará a mejorar la calidad del procesamiento del texto reduciendo el ruido y finalmente generando una clasificación precisa.

Antes de comenzar a abordar las soluciones para la clasificación de los programas de marketing, dentro de la gestión del proyecto, se implementó un plan de trabajo más específico a procesos de análisis de datos, siguiendo la mayoría de los aspectos importantes de la metodología MLOps como proceso estándar para

el aprendizaje automático. Combinando prácticas y herramientas de desarrollo de software sumadas a la implementación de modelos de machine learning. Su objetivo principal parte de una implementación confiable de modelos de ML a través de un control de versiones, la automatización y el trabajo colaborativo. Este flujo de trabajo se ejemplifica en la Figura 9.



**Figure 9:** Metodología del flujo de trabajo

## 4.2 Modelación

Una vez realizada la limpieza del corpus, como dicta la metodología, se preparó el texto para ser introducido a los distintos modelos implementados. Esto solo es posible a través de una vectorización de las palabras que comprenden el corpus de modo que se utiliza un modelo de red neuronal para aprender asociaciones de palabras a partir de un corpus de texto.

### 4.2.1 Word embedding

Word2Vec es un algoritmo de procesamiento del lenguaje natural que representa las palabras en función de vectores numéricos. Está basado en analizar el contexto en el que aparecen las palabras para conocer su significado. Durante su entrenamiento, ajusta los vectores de modo que las palabras similares se agrupen en una distancia cercana dentro del plano. Para este estudio, se utilizó particularmente la librería *Gensim*, una biblioteca de procesamiento de lenguaje natural que proporciona algoritmos para implementar esta vectorización de palabras según su contexto.

### 4.2.2 Decision Tree

Un Decision Tree es un modelo de aprendizaje automático que utiliza una estructura en forma de árbol para representar decisiones y sus consecuencias. Cada nodo interno representa una característica relevante y cada rama representa una opción posible. Las hojas del árbol representan las decisiones finales. Se construye dividiendo los datos en subconjuntos basados en atributos y se utiliza para predecir resultados basados en nuevas instancias. Los árboles de decisión son fáciles de interpretar, pueden manejar datos numéricos y categóricos, y son útiles en clasificación, toma de decisiones y otras aplicaciones.

#### 4.2.3 Support Vector Classifier

Un SVC (Support Vector Classifier) es un algoritmo de aprendizaje automático utilizado para la clasificación de datos. Se basa en el concepto de vectores de soporte y busca encontrar un hiperplano óptimo que separe las diferentes clases de datos en un espacio de características. Utiliza el "kernel trick" para mapear los datos a un espacio de mayor dimensionalidad y puede manejar conjuntos de datos linealmente y no linealmente separables.

#### 4.2.4 K-nearest Neighbor

KNN (k-nearest neighbors) es un algoritmo de clasificación y regresión que se basa en encontrar los vecinos más cercanos a una nueva instancia para determinar su etiqueta o valor. En clasificación, se asigna la etiqueta más común entre los vecinos cercanos, mientras que en regresión se calcula el promedio de los valores de los vecinos. El valor de k representa el número de vecinos considerados en la decisión. KNN es simple pero efectivo y se utiliza en diversas aplicaciones, como reconocimiento de patrones y recomendación de productos.

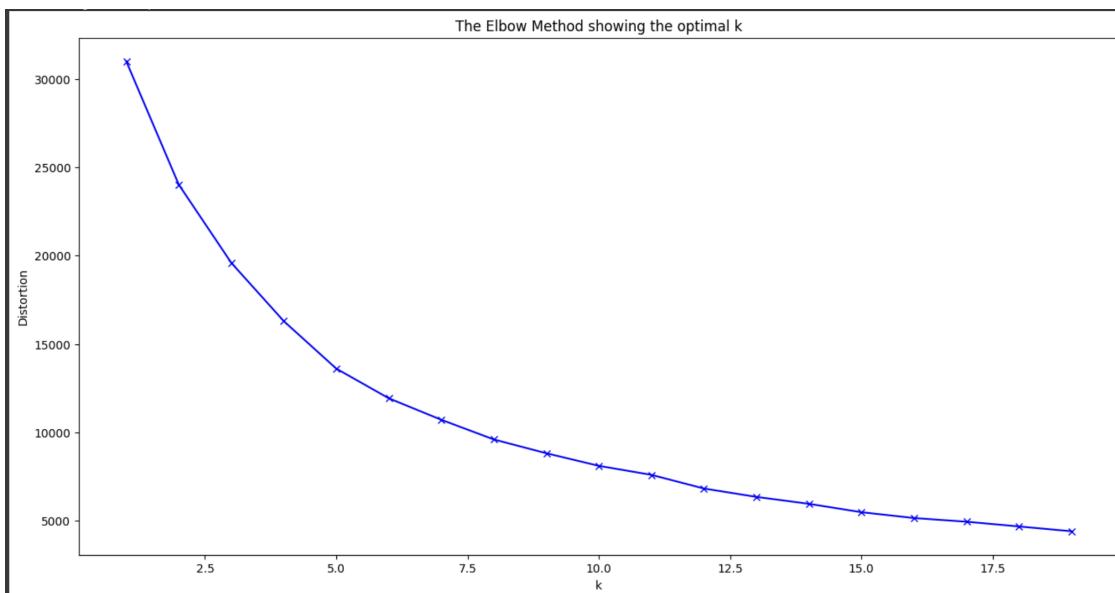
#### 4.2.5 Clustering (K means)

El Clustering es una técnica de aprendizaje automático no supervisado que agrupa instancias de datos similares entre sí en clusters. No requiere etiquetas previas y se basa en medidas de similitud o distancia entre los datos. El objetivo es encontrar patrones y estructuras subyacentes en los datos. El clustering se utiliza en diversas aplicaciones para segmentación y análisis de datos no etiquetados. Es una herramienta poderosa para descubrir información oculta en conjuntos de datos.

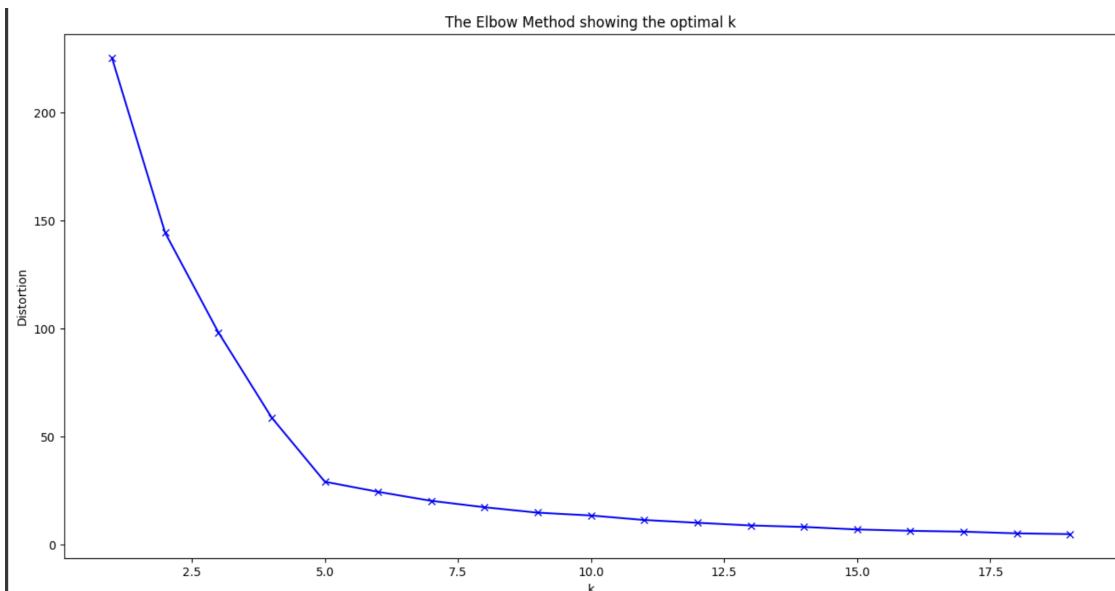
### 4.3 Entrenamiento

Las palabras fueron vectorizadas con el modelo word2vec, utilizando la técnica de skipgram. Una vez generado nuestro espacio vectorial, se realizó el "embedding" de las oraciones. Para esto se tomaron los vectores de las palabras que conformaban cada oración y se promediaron, encontrando así un punto medio geométrico que contextualizaba a la oración dentro del universo de palabras. Una vez, con todas las oraciones, se exploraron los datos con el objetivo de encontrar una manera de separar las categorías linealmente, pero la superposición de las categorías en el espacio hacia imposible categorizarlos con este acercamiento de una manera precisa.

Para atacar este problema se derivaron sub-etiquetas de cada una de las categorías ya existentes con técnicas de clusterización. Con un modelo de K-means y basados en la suma de las varianzas que este arroja, se observó que para varias categorías resulta viable llevar a cabo la subcategorización. Esta se restringe a las categorías mismas, por lo que resultan conjuntos (idealmente) mutuamente exclusivos.



**Figure 10:** Gráfico de codo suma de varianzas K-means categoría "Program"



**Figure 11:** Gráfico de codo suma de varianzas K-means categoría "Digital"

El resultado de esta clusterización, son datos que se pueden separar linealmente de una manera más efectiva, ya que aunque sigan perteneciendo a la misma categoría, y estos se encuentren distribuidos en el mismo sector espacial, encontramos que cada subcategoría tiene una mayor densidad. Preparados de esta manera los datos, se entreno un modelo Support Vector Classifier con kernel lineal el cual, utilizando cross validation y técnicas para optimizar hiperparametros, mostró un desempeño significativamente mejor, si comparamos al etiquetado anterior.

## 5 Fase 4: Validación y presentación de resultados

Alrededor de la interpretación de los modelos y funcionando como sustento para comprender cuáles son las palabras que reinciden con más frecuencia para cada clase, es decir, las más representativas, se generaron nubes de palabras como una representación visual. En una nube de palabras, el tamaño de las palabras crece en función de su frecuencia de ocurrencia como se muestra en la [12](#) para las diversas clases del conjunto de

datos. Es posible identificar la recurrencia de ciertas palabras que a primera vista parecen bastante obvias para pertenecer a cada clase. Sin embargo, si se realiza un análisis más profundo, palabras un poco más pequeñas como *Black Friday* en la clase *Holiday* podrían determinar la clasificación de una categoría, pues a través del utilizado *Word2Vec* estas palabras son el contexto en el que normalmente ocurre cada una de las distintas categorías del conjunto de datos.



**Figure 12:** Nubes de palabras para cada una de las categorías

En un intento de aplicar un primer algoritmo a la base de datos, se implementó un Random Forest, al ser uno de los algoritmos más recomendados cuando existe un importante desbalance entre clases, como es el caso de este estudio. Es importante mencionar que en este primer modelo, el corpus del texto fue mínimamente manipulado, incorporando únicamente la limpieza con expresiones regulares y la librería general de stop words para el idioma inglés además de utilizar solamente el nombre del producto para su clasificación.

Como se puede observar en la Figura 1 , las métricas son muy pobres inicialmente, aunque el rendimiento mejora cuando se aplica la mínima limpieza.

Primer acercamiento del modelo		
	Random Forest sin limpieza, columna <i>Name</i>	Random Forest con limpieza, columna <i>Name</i>
<i>Accuracy</i>	0.632	0.710

**Table 1:** Resultados del primer acercamiento

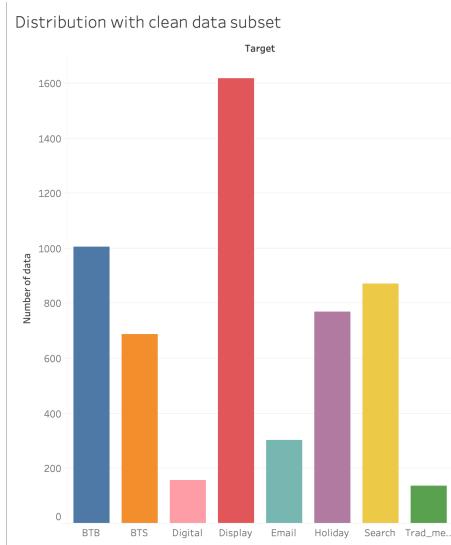
Debido al pobre rendimiento de los modelos, el texto debió ser procesado exhaustivamente, incorporando técnicas avanzadas de procesamiento del lenguaje natural como la lemmatización, descrita previamente y la personalización de stopwords. Por otro lado, se incorporó la columna *Description* como predictor de la variable dependiente, además del predictor ya utilizado *Name*. Los resultados de los cuatro modelos pueden verse en la Tabla 2.

Accuracy			
Modelo	Name	Description	Name + Description
Decision Tree	0.864	0.752	0.858
Support Vector Classifier	0.888	0.541	0.882
K-nearest Neighbor	0.895	0.672	<b>0.932</b>

**Table 2:** Accuracy de los modelos según las columnas predictoras

Como se puede ver en la Tabla 2, el rendimiento de los modelos mejora en su mayoría tras realizar una limpieza exhaustiva pues el eliminar ruido y otros datos irrelevantes permite que el modelo pueda enfocarse en los patrones y otras características importantes. Del mismo modo, al utilizar más columnas como características, se proporciona al modelo una información más completa y contextual, lo que puede mejorar su capacidad para capturar patrones complejos en los datos. En el caso del clasificador Support Vector Classifier (SVC), obteniendo el mejor desempeño entre los modelos, destaca por su capacidad para manejar conjuntos de datos no balanceados lo hace especialmente útil, ya que puede encontrar un equilibrio entre las clases minoritarias y mayoritarias, evitando sesgos y mejorando la precisión general del modelo.

Al tener el resultado de los modelos de como generaron el nuevo data set, las clases quedaron con la distribución representada en la Figura 13.



**Figure 13:** Distribución de clases con la implementación de los clasificadores

En la cual se nota el cambio de distribución a una mas equilibrada entre las clases mas importantes, cuando anteriormente la clase program era la que destacaba de manera alarmante en los datos.

## 6 Conclusiones

El presente estudio explora el uso del análisis semántico para clasificar programas de marketing de forma automática para la empresa HP. Durante el proyecto, se evidenció el uso de metodologías ágiles y un estrategias de manejo de proyecto que llevó a la realización satisfactoria del entregable. El objetivo fue desarrollar un modelo de clasificación que permitiera automatizar el proceso, usando NLP, para identificar el tipo de programa de marketing con la finalidad conocer el flujo de capital sobre cada programa. Se generaron dos hipótesis: La primera consiste en identificar si la precisión del modelo aumenta al utilizar la descripción del programa o si permanece igual. La segunda hipótesis postula que una limpieza basada en stopwords personalizadas y lematización, mejora el rendimiento del modelo. Se ha de notar que aunque estas hipótesis pueden parecer triviales, al utilizar el álgebra de palabras la solución del word embedding, la respuesta a la hipótesis no se ve a primera vista. Se clasificó probando los modelos Decision tree, support vector machine, k-nn y k-means, donde se encontró que el mejor modelo fue SVC con un precisión de 0.932. Como trabajo futuro, se planteó utilizar nuevas técnicas de clustering basados en análisis de datos topológicos. Se presentó el algoritmo mapper para visualizar y clusterizar las nubes de palabras. Y se expuso de forma hipotética como sería más facil clasificar las oraciones si se tiene un álgebra de la forma cualitativa de cada clasificación. En conclusión, se aceptó la hipótesis alternativa de ambas pruebas, donde al utilizar la columna descripción y ser limpiada con lematización y stopwords personalizadas dió un mejor resultado.

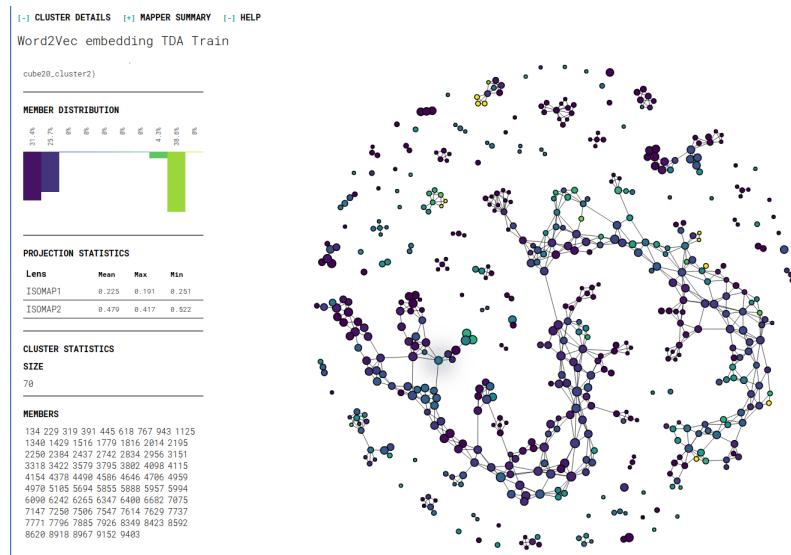
### 6.1 Trabajos futuros

Como trabajos futuros se busca analizar si las categorías ya asignadas representan un contenido semántico total y único. Por total nos referimos a que dos categorías que refieran al mismo tema no se encuentren separadas, sino se unan en una sola. Y por único nos referimos a que dentro de una categoría, su contenido sea disjunto de otras categorías. El beneficio hipotético de expandir tal expresividad conlleva una mejor diferenciación de las clases al entrenar los modelos, debido a que se elimina la contaminación de las etiquetas al tener observaciones mal clasificadas de una categoría asignada a otra.

Se propone utilizar el análisis topológico de datos (TDA; Topological Data Analysis) para identificar las relaciones entre categorías, y generar nuevas etiquetas al segmentar los datos por grupos. TDA engloba métodos computacionales basados en topología algebraica para el cálculo eficiente de los invariantes de un objeto, poder medirlos y compararlos. Profundizando, el análisis topológico de datos consiste en explorar la forma cualitativa de los datos para formar agrupaciones representativas. La noción base parte de los algoritmos de agrupación no supervisados (ej. k-means). Sin embargo, su fortaleza recae en representar relaciones complejas de alta dimensionalidad (ej. el conjunto de células que forma tejidos, y estos órganos, y estos sistemas). Estas representaciones con alta relación se encapsulan en los complejos simpliciales. Los complejos simpliciales son una abstracción de los grafos. De estos se parten para extraer las características homológicas [1]. Pensar de forma topológica consiste en olvidarnos de los datos y pensar en las relaciones entre ellos.

La principal herramienta para extraer características es la homología simplicial persistente. Describe las características de los datos mediante el número de hoyos, ciclos, y vacíos en diferentes dimensiones [3]. Ejemplificando con un conjunto de oraciones que pertenecen a una clase; las oraciones embebidas en un vector generan un espacio vectorial n-dimensional, cada oración se representa como un punto en ese espacio; oraciones con significado cercano corresponderán a puntos cercanos. Estos puntos se calculan los complejos simpliciales. Esto se hace ya que la topología solo se puede calcular sobre objetos geométricos como un Manifold. Mediante la construcción de complejos simpliciales, el objeto construido son homotópicamente equivalentes a un Manifold, por lo que el cálculo de la topología ya es posible.

Como objetivo para encontrar las relaciones entre categorías se utiliza el algoritmo MAPPER. MAPPER es un algoritmo de visualización y clusterización suave para datos que expresan alta relacionalidad [5]. Este algoritmo utiliza un acercamiento de análisis topológico de datos que le permite tener mayor flexibilidad que otros visualizadores no permiten. En el algoritmo, la nube de puntos se divide en subconjuntos, donde cada subconjunto se representa como un nodo en un grafo. Los enlaces entre nodos representan objetos en común entre dos o más nodos. Para filtrar los datos primero se utiliza un análisis de norma, centralidad, PCA y ecentricidad. Una desventaja es que es difícil de tunear, se tiene que realizar multiples iteraciones para llegar a la visualización requerida.



**Figure 14:** El algoritmo MAPPER implementado en el conjunto de entrenamiento.

Como objetivo final se consideraría oportuno generar los invariantes topológicos de cada categoría. Esta base da datos de la forma en esencia de cada categoría, nos permitiría, en base a una nueva oración no clasificada.

ficada calcular su homología persistente y poder compararla contra las categorías ya dadas en nuestra base de datos. Este álgebra de categorías hipotéticamente facilitaría la clasificación de nuevas oraciones ya que en lugar de modelos complejos, todo se resume a comparar invariantes [2, 8], y mejora la interpretabilidad (una vez pasando la curva de aprendizaje del tema) ya que se podría apreciar porque tal oración (ejemplificando vagamente: que su invariante tiene forma de estrella) se encuentra clasificado en tal categoría (que también todos sus elementos tienen forma de estrella).

Entre los recursos computacionales para cumplir tal objetivo se encuentran las librerías de TDA GUDHI, Scikit-tda, Dionysus, PHAT, SIPHA y Giotto. Como referencia en la literatura se pueden consultar a Gunnar Carlsson, Frédéric Chazal y Raúl Rabadán, entre otros igual de relevantes.

## 7 Anexos

### 7.1 Código fuente

El repositorio de GitHub que incluye todas las libretas de código mencionadas en este documento se encuentra en el siguiente link: [https://github.com/renatauribes/GrayProject\\_NLPmarketing.git](https://github.com/renatauribes/GrayProject_NLPmarketing.git)

### 7.2 Tablero Miró

El siguiente link muestra el Tablero Miró perteneciente al Equipo 4 en donde fueron registradas las distintas etapas de la realización del presente proyecto: [https://miro.com/app/board/uXjVMX0x1LA=?share\\_link\\_id=952508584737](https://miro.com/app/board/uXjVMX0x1LA=?share_link_id=952508584737)

## References

- [1] Gunnar Carlsson. "Topology and data". In: *Bulletin of the American Mathematical Society* 46.2 (Jan. 29, 2009), pp. 255–308. ISSN: 0273-0979. DOI: [10.1090/S0273-0979-09-01249-X](https://doi.org/10.1090/S0273-0979-09-01249-X). URL: <http://www.ams.org/journal-getitem?pii=S0273-0979-09-01249-X> (visited on 02/25/2023).
- [2] Gunnar Carlsson et al. "Topological Data Analysis and Machine Learning Theory". In: () .
- [3] Frédéric Chazal and Bertrand Michel. *An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists*. Feb. 25, 2021. arXiv: [1710.04019\[cs, math, stat\]](https://arxiv.org/abs/1710.04019). URL: <http://arxiv.org/abs/1710.04019> (visited on 02/25/2023).
- [4] Dominik Cvetek et al. "APPLICATION OF NLP ALGORITHMS IN ITS MARKET ANALYSIS". In: May 2018.
- [5] Gretchen Langenbahn. *Topological Data Analysis with Mapper*. Apr. 27, 2022. URL: <https://scholarworks.bgsu.edu/honorsprojects/732>.
- [6] Ken Schwaber and Mike Beedle. *Agile Software Development with Scrum*. Upper Saddle River, New Jersey: Prentice Hall, 2002. ISBN: 0130676349. URL: <http://portal.acm.org/citation.cfm?id=559553&dl=ACM&coll=portal>.
- [7] Sanjeev Verma et al. "Artificial intelligence in marketing: Systematic review and future research direction". In: *International Journal of Information Management Data Insights* 1.1 (2021), p. 100002. ISSN: 2667-0968. DOI: <https://doi.org/10.1016/j.jjimei.2020.100002>. URL: <https://www.sciencedirect.com/science/article/pii/S2667096820300021>.
- [8] Ali Zia et al. "Topological Deep Learning: A Review of an Emerging Paradigm". In: (2023). Publisher: arXiv Version Number: 1. DOI: [10.48550/ARXIV.2302.03836](https://doi.org/10.48550/ARXIV.2302.03836). URL: <https://arxiv.org/abs/2302.03836> (visited on 02/23/2023).