**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY**

**A blue and white logo

Description automatically generated**

# 

# Clasificación automática de descripciones de carga contenerizada en códigos del Sistema Armonizado (HS) usando modelos de Procesamiento de Lenguaje Natural

**Prof. Titular: Dra. Grettel Barceló Alonso**

**Juan Pablo Acosta López (A01794035)**

**Oscar Villa Cárdenas (A01794052)**

**Genaro Rodriguez Vázquez (A01150931)**

**29 septiembre 2024**

**Índice**

[Clasificación automática de descripciones de carga contenerizada en códigos del Sistema Armonizado (HS) usando modelos de Procesamiento de Lenguaje Natural](#_heading=h.crzpksqdu7ux)

[¿Què es Feature Engineering?](#_heading=h.j0432rcuia01)

[Selección / extracción de características](#_heading=h.ao0fgfb04ccy)

[Proceso de webscrapping](#_heading=h.5al8c2f4zwgx)

[Limpieza de datos](#_heading=h.utrksgavpb7y)

[Conclusiones](#_heading=h.xhep8orgpmfj)

[Referencias](#_heading=h.ud6c1uqe1vfv)

[(N.d.). Retrieved from https://www.dripcapital.com/hts-code/section/all](#_heading=h.yxuj5vdccito)

## ¿Què es Feature Engineering?

El Feature Engineering es un proceso clave en el desarrollo de modelos de Inteligencia Artificial (IA) y aprendizaje automático. Consiste en seleccionar, transformar y crear características (features) a partir de los datos crudos disponibles, con el objetivo de mejorar el rendimiento de los modelos.

En el contexto de Machine Learning tradicional, las características son las variables o atributos que se utilizan para que el modelo aprenda y haga predicciones, esto implica:

* Selección de características relevantes: Identificar qué variables del conjunto de datos son importantes para el problema en cuestión y eliminar aquellas que no aportan valor o son redundantes. Esto ayuda a reducir el ruido y evitar el sobreajuste.
* Transformación de características: A veces, las características originales no son lo suficientemente representativas y requieren ser modificadas o transformadas para capturar mejor los patrones en los datos. Esto puede implicar la normalización o estandarización de datos, el uso de transformaciones matemáticas, o la creación de variables categóricas a partir de variables numéricas (como el one-hot encoding).
* Creación de nuevas características: En muchos casos, los datos crudos no contienen toda la información necesaria, por lo que es útil generar nuevas características derivadas. Por ejemplo, si tienes datos de tiempo, podrías generar nuevas variables como "día de la semana", "hora del día", etc., que podrían mejorar la precisión del modelo.
* Reducción de dimensionalidad: Si el conjunto de datos tiene demasiadas variables, se puede aplicar técnicas como PCA (Análisis de Componentes Principales) para reducir el número de características, manteniendo la mayor cantidad de información posible.

El **Feature Engineering** en un **RAG (Retrieval-Augmented Generation)** es diferente al tradicional. En lugar de trabajar con datos numéricos o categóricos, se enfoca en procesar texto para optimizar la recuperación y generación de información.

1. **Generación de nuevas características** no aplica, ya que el enfoque es en seleccionar información textual relevante, no en crear variables.
2. **Discretización** y **codificación** (como one-hot) no se usan porque se trabaja con texto, no con variables numéricas o categóricas.
3. **Escalamiento** y **transformaciones matemáticas** no son necesarios ya que los modelos de lenguaje operan con embeddings de texto, no con datos numéricos.

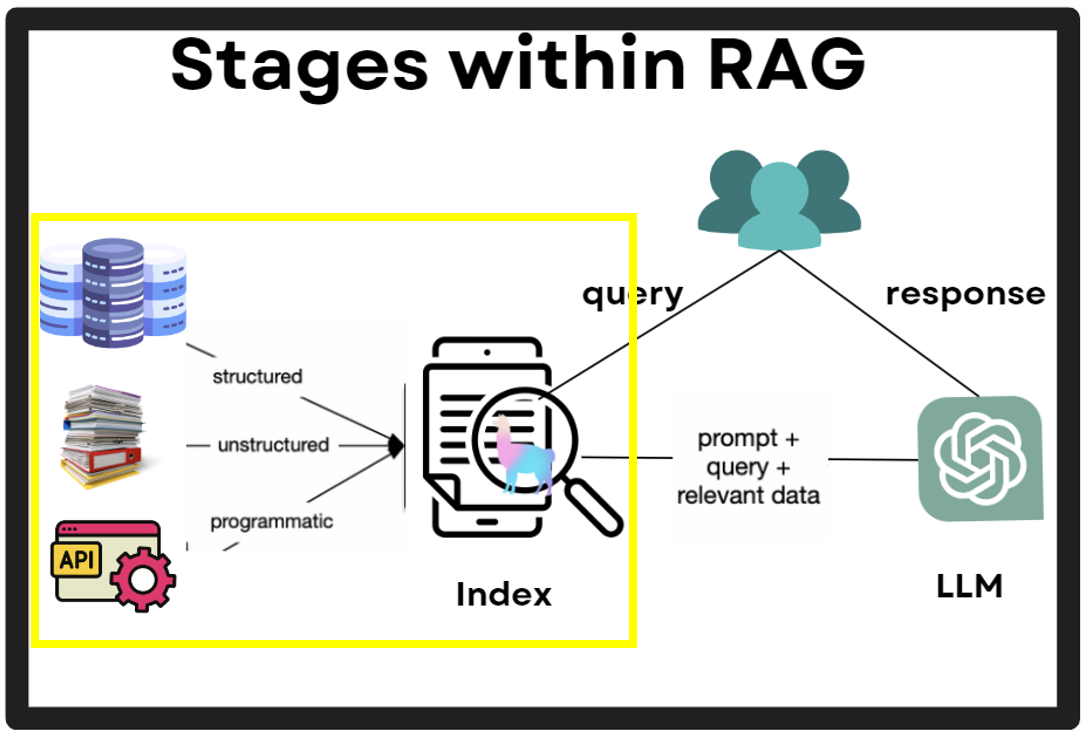
El Feature Engineering en RAG se centra en la **indexación**, **tokenización** y **selección de contexto** relevante para mejorar el rendimiento del modelo generativo.

La **normalización de texto** en un RAG es clave para mejorar la recuperación y generación de información. El proceso normalmente incluye:

1. **Limpieza**: Elimina caracteres especiales, HTML y puntuación innecesaria.
2. **Minúsculas**: Convierte todo el texto a minúsculas para dar consistencia.
3. **Eliminación de stop words**: Reduce palabras comunes sin valor informativo.
4. **Tokenización**: Divide el texto en palabras o frases para su procesamiento.
5. **Lematización o stemming**: Simplifica palabras a su forma base o raíz.

Esto asegura que el texto esté limpio y coherente, mejorando la precisión de la búsqueda y las respuestas generadas en el RAG.

Dicho proceso de normalización es un punto medular en la arquitectura de un RAG y se puede ilustrar para un mejor entendimiento de la siguiente manera:



***Fig 1****: Proceso de alimentación del índice en un sistema RAG (Retrieval-Augmented Generation). El índice se nutre de datos estructurados (como bases de datos y hojas de cálculo), datos no estructurados (como documentos y archivos de texto), y datos programáticos (provenientes de APIs y servicios automatizados). Estos datos se procesan y normalizan para ser utilizados en la fase de recuperación, donde se selecciona información relevante para la generación de respuestas.*

## Selección / extracción de características

El web scraping puede considerarse como un proceso de **extracción de datos**, ya que estamos seleccionando qué información recolectar de las páginas web y qué datos ignorar. En este sentido, el scraping involucra decisiones sobre qué datos son relevantes y cómo extraerlos eficientemente para nuestro proyecto. Por ejemplo, podríamos estar filtrando ciertos tipos de contenido o solo extraer ciertas etiquetas HTML o bloques de texto.

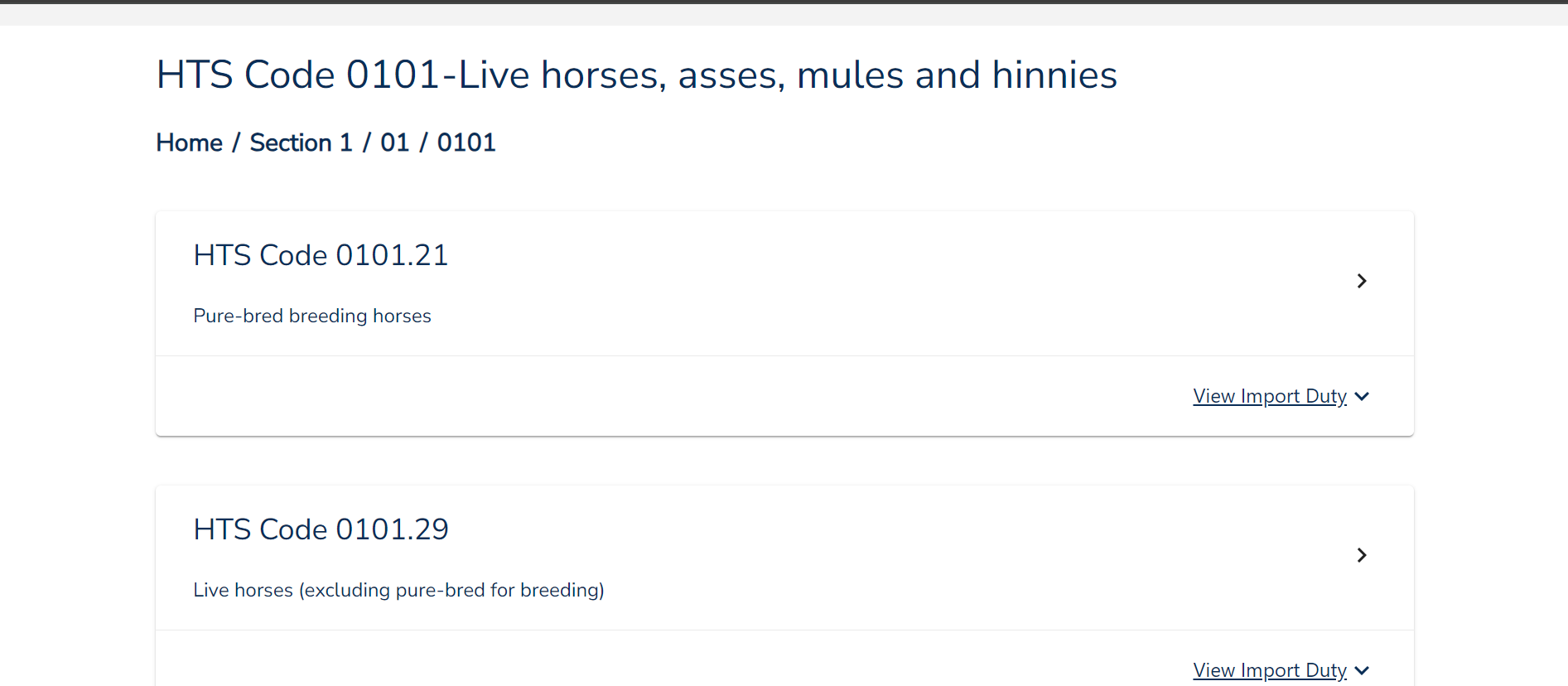


Figura 1: Página de la cual se hizo el web scrapping

En el scraping tomamos decisiones sobre la selección y extracción de la información adecuada que reducirá la complejidad del procesamiento posterior, facilitando la interpretación y el uso de los datos para el modelo LLM.

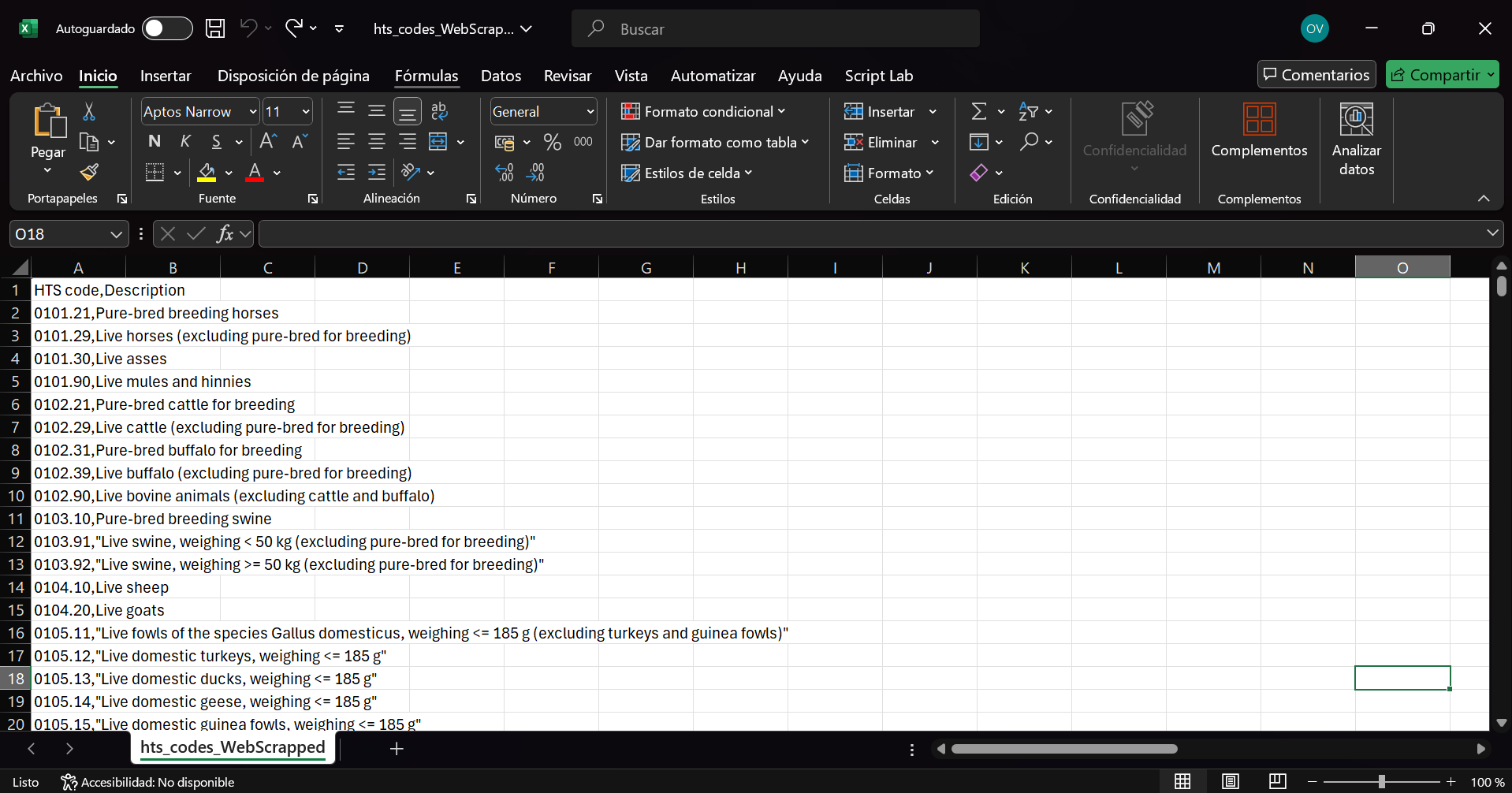


Figura 2: Como luce el archivo csv resultado del proceso de web scrapping

### Proceso de webscrapping

**Extracción de datos relevantes**:

* El código accede a una serie de secciones y capítulos dentro de una estructura en línea que organiza los códigos HTS. Para cada sección y capítulo, busca elementos HTML específicos que contienen los códigos y descripciones.

**Selección de contenido**:

* El código filtra los enlaces que contienen los números de capítulo a través de una comprobación de URL que incluye /hts-code/{section}/{chapter} para asegurarse de que solo se están procesando los enlaces correctos.

**Reducción de complejidad**:

* A lo largo del código, se eliminan prefijos y se limpia el texto para reducir el contenido no necesario y centrarse en los códigos y descripciones útiles. Por ejemplo, el prefijo "HTS Code " se elimina de cada fila antes de almacenar el código HTS en el dataframe.
* Este proceso de selección evita la recolección de información redundante o innecesaria, lo que **reduce la complejidad** del análisis posterior y optimiza los datos para su uso en el modelo.

**Facilitar la interpretación**:

* Al estructurar la salida en un **DataFrame de pandas**, los datos quedan listos para ser analizados o procesados más adelante. La extracción de solo los códigos y descripciones relevantes mejora la interpretabilidad de los datos obtenidos, lo que facilita la posterior carga o análisis de este contenido para fines de modelado o clasificación.

## Limpieza de datos

Debido a que los manifiestos aduanales suelen llenarse de manera manual por operadores humanos, las descripciones del contenido frecuentemente incluyen información irrelevante para la clasificación de la carga. Este problema se manifiesta en la inclusión de datos innecesarios, lo que dificulta una correcta categorización en función de los códigos del Sistema Armonizado (HS Code). En el marco de este proyecto, se considera evaluar la utilidad de implementar sub-modelos para la extracción de información relevante del manifiesto, tales como un modelo de traducción automática y el modelo TF-IDF, entre otros. El objetivo de esta evaluación futura es determinar si la identificación de información pertinente puede optimizarse mediante técnicas de procesamiento de lenguaje natural (NLP), reduciendo así el volumen de ruido presente en los datos y mejorando la precisión en la clasificación de las mercancías.

A continuación, se detallan los pasos propuestos para el pipeline de limpieza de datos dentro del contexto del presente estudio:

* Preprocesamiento de texto: El primer paso consiste en normalizar el texto de los manifiestos. Se propone convertir el texto a minúsculas para garantizar la uniformidad, seguido de la eliminación de signos de puntuación, caracteres especiales y espacios en blanco innecesarios. Esto se realiza con el objetivo de obtener una representación homogénea del texto que sea más manejable para las etapas subsiguientes.
* Traducción automática: Dado que los manifiestos pueden estar en diferentes idiomas, es fundamental traducir el contenido a un idioma común, como el inglés. Para ello, se propone el uso de un modelo de traducción automática que convierta de manera eficiente el texto de un idioma fuente (por ejemplo, español) a inglés. Esto facilita la posterior limpieza y análisis de los datos, asegurando que el pipeline funcione de manera coherente independientemente del idioma en que se encuentre el manifiesto original.
* Lematización y limpieza avanzada: Posteriormente, se lleva a cabo un proceso de lematización, en el cual las palabras se reducen a su forma base o raíz. Esto ayuda a eliminar variaciones morfológicas de las palabras que podrían añadir ruido al análisis, asegurando que términos con el mismo significado sean tratados como equivalentes. A continuación, se eliminan palabras irrelevantes para el proceso de clasificación, tales como stopwords, que son términos comunes pero no significativos (por ejemplo, "y", "el", "de"), así como palabras personalizadas que han sido identificadas como irrelevantes para este contexto específico (por ejemplo, "caja", "pallet", "manifiesto").
* Análisis de tokens clave: En esta etapa se identifican los tokens más importantes en el texto. Para ello, se combinan técnicas de análisis de frecuencias (como el conteo de palabras) con modelos de atención que permiten asignar puntuaciones de relevancia a los distintos tokens. Estas puntuaciones se calculan en función de su relevancia en el contexto del texto y de su aparición en la estructura global del manifiesto. De este modo, se filtran aquellos términos que no aportan valor significativo al análisis y se seleccionan los más representativos para la clasificación final.
* Extracción de tokens relevantes: Finalmente, con base en los resultados del análisis de tokens, se extraen las palabras o términos más relevantes para la correcta categorización de la carga. Este conjunto de tokens se utilizará como entrada para los modelos de clasificación posteriores, mejorando la precisión y la eficiencia en la asignación de códigos del Sistema Armonizado.
* Este pipeline representa un enfoque iterativo que busca optimizar tanto la calidad de los datos como la relevancia de la información utilizada para la clasificación. Al reducir el ruido y enfocarse en los elementos más significativos, se espera que los modelos de clasificación puedan operar con mayor precisión y eficiencia, alineándose con los objetivos del presente proyecto de automatización en la clasificación de mercancías.

## Conclusiones

Aunque es posible ofrecer una conclusión sobre el pre-procesamiento de los datos textuales (como tokenización, limpieza de texto, etc.), la metodología CRISP-ML está más orientada a proyectos de ciencia de datos tradicionales con modelos numéricos o categóricos. En los proyectos de LLM, la preparación de datos es diferente, ya que se centra más en la limpieza textual y menos en la transformación de variables numéricas. A su vez, es necesario utilizar una correcta arquitectura de transformación de sentencias dependiendo del caso de uso, que en este caso buscamos una mezcla entre clasificación, pero a su vez requerimos de similitud semántica.

Además, en el pipeline de limpieza propuesto, se incluyó la limpieza de texto (normalización, lematización, eliminación de stopwords), la traducción automática de los manifiestos a un idioma común y la extracción de tokens relevantes mediante análisis de frecuencias y técnicas de atención. Este enfoque no solo busca la clasificación correcta, sino también la captura de la similitud semántica, asegurando que descripciones textuales diferentes pero equivalentes se traten de forma coherente. Así, la metodología se adapta a la naturaleza específica de los modelos LLM, donde la comprensión semántica es clave para el éxito del proyecto.

## Referencias

Inmon, B. (2017). Turning text into gold: Taxonomies and textual analytics. Bradley Beach: Technics Publications

Manning, C., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. Cambridge: Cambridge University Press.

Wilbur, W. J., & Sirotkin, K. (1992). The automatic identification of stop words. Journal of Information Science, 18(1), 45-55

#### (N.d.). Retrieved from <https://www.dripcapital.com/hts-code/section/all>