**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)**

**27 Octubre 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)

[Ensambles](#_heading=h.r4t8hiypfa34)

[Selección](#_heading=h.tdc9y98tyd4q)

[Los modelos deben ser ordenados por la métrica principal, pero el resumen debe incorporar otras métricas pertinentes](#_heading=h.hl3zznbkntu)

[Se deberán incluir también los tiempos de entrenamiento](#_heading=h.4ubzu8daejaf)

[Se elige el modelo final alineado con los objetivos y necesidades del negocio.](#_heading=h.74r50r36ybqn)

[Gráficos](#_heading=h.losvxh665u0w)

[Matriz de Confusion](#_heading=h.78gu1m82t0d2)

[Mistral7b](#_heading=h.h0adhs23zz9y)

[Llama3.2](#_heading=h.lrmgzl1wddgs)

[Conclusiones](#_heading=h.xhep8orgpmfj)

[Referencias](#_heading=h.ud6c1uqe1vfv)

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

## Ensambles

Los modelos de lenguaje grande (LLM), debido a su capacidad para procesar y entender patrones lingüísticos complejos, ofrecen una solución significativamente superior en el contexto de la clasificación de códigos HS. Estos modelos están diseñados para capturar relaciones semánticas profundas en grandes volúmenes de datos textuales, logrando con alta precisión y fiabilidad tareas complejas de clasificación, incluso en estructuras jerárquicas como las de los códigos HS. En pruebas y evaluaciones previas, los LLM han demostrado un rendimiento notablemente superior en comparación con otros modelos de clasificación estándar, al poder clasificar tanto a nivel general como específico sin requerir la combinación de modelos adicionales. Esto significa que un modelo LLM bien ajustado puede obtener resultados óptimos en clasificación, evitando la redundancia y sobrecarga de recursos asociada con los ensambles.

Además, en términos de eficiencia y precisión, los modelos de ensamble no ofrecen ventajas sustanciales en este caso, ya que los LLM ya son capaces de maximizar el rendimiento en esta tarea específica. Las características avanzadas de estos modelos, como los mecanismos de atención, permiten que un solo modelo pueda captar detalles y variaciones importantes sin la necesidad de recurrir a estrategias adicionales de ensamble, que, en este contexto, solo introducirían una complejidad innecesaria. Por lo tanto, un modelo de lenguaje grande representa la opción más robusta y rentable para clasificar códigos HS, dado que proporciona resultados de alta calidad mientras optimiza el uso de recursos y simplifica el desarrollo y mantenimiento del proyecto.

## Selección

### Los modelos deben ser ordenados por la métrica principal, pero el resumen debe incorporar otras métricas pertinentes

|  | **GPT4o + text-embedding-3-large NO CHUNKS** | **GPT3.5 + text-embedding-3-large NO CHUNKS** | **Mistral7b + all-mpnet-base-v2 NO CHUNKS** | **TestA (Llama3.2 + all-mpnet-base-v2 NO CHUNKS)** |
| --- | --- | --- | --- | --- |
| **Accuracy (2-digit)** | 73.00% | 70.00% | 50.00% | 43.00% |
| **Accuracy (4-digit)** | 53.00% | 51.00% | 23.00% | 24.00% |
| **Costo embeddings de entrenamiento (103,703 tokens)** | 0.01348 USD\* | 0.01348 USD\* | 0 USD | 0 USD |
| **Costo por consulta** | $0.03825 USD \*\* | $0.03825 \*\* | 0 USD | 0 USD |

\*Para calcular el costo de generar embeddings utilizando el modelo "text-embedding-3-large" de OpenAI, consideramos la siguiente información:

* **Cantidad de tokens en el dataset**: Aproximadamente 103,703 tokens.
* **Precio por token del modelo "text-embedding-3-large"**: $0.00013 por 1,000 tokens

\*\* Para calcular el costo de cada consulta de descripción de manifiesto, para obtener un HS code por parte de OpenAI, se calcula la cantidad de tokens enviado en cada template (255 tokens):

* **Cantidad de tokens en template**: 255 tokens
* **Precio por consulta GPT4o y/o GPT3.5**: $0.0015 por 1,000 tokens

### Se deberán incluir también los tiempos de entrenamiento

Dado que se utiliza una técnica de **Retrieval-Augmented Generation (RAG)**, donde el modelo de lenguaje se adapta mediante un **fine-tuning ligero**, los tiempos de entrenamiento son rápidos y eficientes. Esto implica que el modelo no necesita ser ajustado constantemente, sino solo cuando se actualiza el diccionario de códigos HS, algo que típicamente ocurre una vez al año.

Esta estructura de actualización ocasional reduce significativamente la carga de recursos y costos, ya que no se requiere un reentrenamiento frecuente o intensivo. Al mantener una capa de embeddings actualizada en lugar de rehacer el modelo completo, se facilita la respuesta precisa ante cambios en el catálogo de códigos, sin interrumpir el servicio o generar costes elevados. Además, esta estrategia permite mantener el rendimiento del sistema siempre alineado con la última versión de los códigos, maximizando así su precisión de forma eficiente y económica.

Los tiempos para crear las bases de datos vectoriales, donde se alojan los embeddings del diccionario de HS codes son:

* **text-embedding-3-large**: 36.43 minutos por medio de API
* **all-mpnet-base-v2**: 16.22 minutos / GPU Nvidia RTX3050.

Las diferencias en los tiempos se deben a:

**Infraestructura de Procesamiento**:

* *text-embedding-3-large* se está ejecutando a través de una API, lo que significa que el procesamiento depende del servidor remoto y su infraestructura, que puede estar limitada por la disponibilidad de recursos, congestión de red, y latencia de comunicación entre el cliente y el servidor. Estos factores pueden agregar un tiempo considerable.
* *all-mpnet-base-v2*, en cambio, se ejecuta localmente en una GPU Nvidia RTX3050, que tiene acceso directo al procesamiento de datos sin la latencia de red o dependencia en servidores externos.

**Arquitectura y Tamaño del Modelo**:

* *text-embedding-3-large* es un modelo más grande, lo que implica más parámetros y capas de atención, haciendo que cada procesamiento sea más pesado en términos computacionales. Estos modelos grandes son más lentos, especialmente si no están optimizados para GPU.
* *all-mpnet-base-v2* es una versión optimizada que usa menos parámetros en comparación con modelos “large” y suele ser más eficiente al ejecutar tareas de embeddings.

**Optimización para Hardware**:

* La GPU Nvidia RTX3050 proporciona una aceleración significativa en los cálculos en comparación con un procesamiento en CPU o en servidores que no necesariamente tienen la misma optimización para modelos grandes.
* Si el servidor donde corre *text-embedding-3-large* no usa aceleración GPU o esta está limitada, el tiempo será mayor en comparación con un modelo optimizado corriendo en una GPU moderna como la RTX3050.

**Bibliotecas y Optimizaciones Locales**:

* Al correr *all-mpnet-base-v2* localmente, puedes aprovechar optimizaciones específicas de bibliotecas como PyTorch o TensorFlow, que tienen mejoras de rendimiento específicas para GPUs. En la API de *text-embedding-3-large*, el control sobre estas optimizaciones depende de cómo esté configurado el backend de la API.

### Se elige el modelo final alineado con los objetivos y necesidades del negocio.

Si bien los modelos de OpenAI presentan un mayor porcentaje de precisión que el resto, seleccionar el modelo que mejor se adapte a las necesidades del negocio requiere de un análisis más profundo. El modelo **Mistral7b** combinado con los embeddings de **all-mpnet-base-v2** se presenta como una opción estratégica para un negocio que priorice tanto el **coste** como el **manejo privado de la información**.

**Coste Reducido**: En comparación con modelos avanzados como GPT-4 o GPT-3.5, el modelo Mistral7b con all-mpnet-base-v2 ofrece un gran beneficio económico al no tener costos asociados con los embeddings de entrenamiento ni por consulta. Esto resulta en una reducción significativa en los costos operativos, lo cual es particularmente ventajoso para aplicaciones que requieren consultas frecuentes o de gran volumen. Este ahorro de costos permite al negocio maximizar su retorno de inversión, dedicando recursos a otras áreas estratégicas sin sacrificar la viabilidad del sistema de recomendación.

**Cumplimiento de Normativas de Privacidad**: Sectores como logística, aduanas, finanzas, salud o servicios legales enfrentan regulaciones estrictas sobre la privacidad de los datos. Utilizar modelos cerrados como GPT-4 puede representar riesgos regulatorios al depender de servicios externos para el procesamiento de datos confidenciales. En cambio, Mistral7b permite cumplir con normativas más estrictas (como GDPR o CCPA) al mantener el procesamiento dentro de la infraestructura de la empresa, minimizando el riesgo de filtración de datos.

Sumado a estos 2 principales factores de decisión, algunas razones adicionales para optar por **Mistral7b + all-mpnet-base-v2** sobre modelos de mayor precisión son:

**Escalabilidad en Implementaciones Locales**: Mistral7b + all-mpnet-base-v2 se puede implementar en servidores propios o entornos controlados, permitiendo que el modelo escale internamente sin incurrir en altos costos por consultas. Esto es ideal para empresas que necesitan expandir el uso del modelo sin aumentar proporcionalmente el gasto operativo, algo que sería prohibitivo con modelos de pago por consulta como GPT-4 o GPT-3.5.

**Flexibilidad en Personalización del Modelo**: Como Mistral7b es de código abierto, las empresas tienen más libertad para ajustar y optimizar el modelo según sus necesidades específicas, mejorando potencialmente la precisión en los casos de uso concretos sin depender de los proveedores externos. Esta flexibilidad es valiosa para empresas que buscan adaptarse rápidamente a cambios en el negocio o en el tipo de datos que manejan.

**Predictibilidad de Costos a Largo Plazo**: A medida que el negocio crece, un modelo como GPT puede escalar rápidamente en costos debido a la estructura de cobro por consulta. Mistral7b, sin embargo, permite prever costos fijos, ya que su implementación y mantenimiento no depende de un número de consultas. Esto hace que el presupuesto sea más manejable y estable, facilitando una planificación financiera a largo plazo.

**Mejor Integración con Infraestructura Interna**: Al ser un modelo abierto y flexible, Mistral7b se integra mejor con sistemas internos de TI, lo que facilita la combinación de la inteligencia del modelo con bases de datos, herramientas de gestión o incluso otros modelos internos. Esto se traduce en una mayor eficiencia operativa y en una experiencia más fluida para los equipos de TI y datos.

## Gráficos

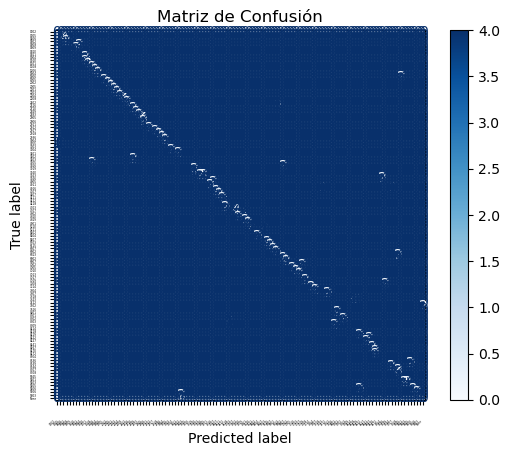
La mayoría de las gráficas solicitadas, excepto la matriz de confusión, no aplican al contexto del modelo seleccionado (GPT-4o + text-embedding-3-large sin fragmentación) debido a la naturaleza del modelo y la tarea específica de clasificación de códigos HS basada en procesamiento de lenguaje natural (NLP).

* Curva ROC y Curva de Precisión-Recall: Estas gráficas son útiles en problemas de clasificación binaria o multicategoría cuando se analizan probabilidades de pertenencia a diferentes clases. Sin embargo, en el caso del modelo elegido, la clasificación de códigos HS se realiza directamente a través de embeddings y generación de texto sin una probabilidad asociada a cada categoría. Al no operar con scores probabilísticos para cada clase, estas curvas no son aplicables ni útiles en la interpretación de su rendimiento.
* Análisis de Residuos: El análisis de residuos es más adecuado para modelos de regresión o clasificación en los que se puede evaluar el error como la diferencia entre el valor real y el predicho en términos cuantitativos. En este caso, al trabajar con un modelo generativo de lenguaje que clasifica directamente con base en contexto y patrones semánticos, no se genera un "residuo" clásico, ya que no hay valores numéricos continuos que comparar.
* Gráfico de Importancia de Características: Este tipo de gráfico es característico en modelos como árboles de decisión o algoritmos basados en árboles de ensamblaje, donde es posible medir la importancia de cada variable de entrada en el resultado final. En el modelo LLM elegido, el procesamiento se basa en embeddings textuales de frases completas y no en variables individuales, lo que impide descomponer el resultado en importancias de características específicas.
* Tendencia y Predicción (para series temporales): Este gráfico aplica únicamente a modelos de series temporales, donde se analiza la evolución de una variable a lo largo del tiempo. La clasificación de códigos HS no es un problema de series temporales, ya que el modelo no se basa en datos secuenciales por tiempo, sino en la clasificación de texto descriptivo en un único momento.
* Diagrama de Árbol: Este gráfico corresponde exclusivamente a algoritmos de árboles de decisión. Dado que el modelo elegido es un LLM, no emplea una estructura de árbol para tomar decisiones y, por lo tanto, no es posible ni relevante generar un diagrama de árbol.

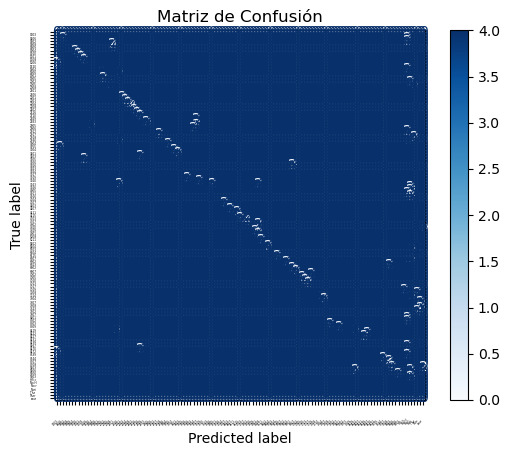
La matriz de confusión, en cambio, sí aplica en este contexto, ya que permite evaluar el rendimiento del modelo al mostrar la cantidad de clasificaciones correctas e incorrectas para cada categoría de códigos HS, ofreciendo una interpretación intuitiva del desempeño en clasificación multinivel.

### Matriz de Confusión

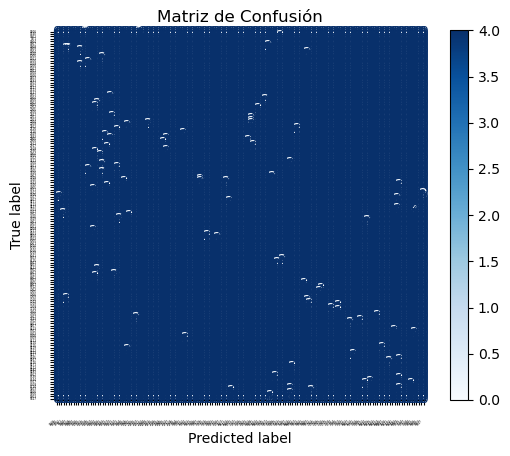
#### GPT4o



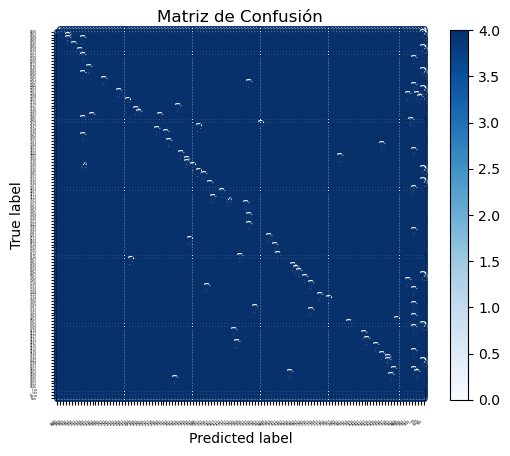
#### GPT3.5



#### Mistral7b



#### Llama3.2



## Conclusiones

En resumen, los modelos de lenguaje grande son opciones destacadas para la clasificación de códigos HS debido a su capacidad para procesar estructuras jerárquicas complejas y captar relaciones semánticas profundas en grandes volúmenes de texto. Entre los modelos evaluados, GPT-4o, junto con el text-embedding-3-large (sin segmentación), demostró el mejor rendimiento, superando a alternativas como GPT-3.5, Mistral-7B y Llama 3.2 en precisión tanto a nivel general (2 dígitos) como específico (4 dígitos). Esto se debe a la arquitectura avanzada de GPT-4o y su habilidad para identificar patrones detallados sin dividir los textos, proporcionando una visión integral de cada entrada.

La técnica de **Retrieval-Augmented Generation (RAG)** optimiza la eficiencia de consultas al emplear bases de datos de dominio específico, como los códigos HS, evitando costos de entrenamiento adicionales y manteniendo precisión sin introducir complejidades innecesarias. Esto hace de GPT-4o una solución robusta y rentable para clasificar códigos HS, combinando precisión y simplicidad operativa de forma óptima.

Desde una perspectiva empresarial, se deben de considerar factores más alla de la simple precision de una predicción, **GPT-4o es altamente competitivo para la clasificación de códigos HS**, ofreciendo un rendimiento superior con un 73% de precisión en la clasificación a 2 dígitos y un 53% a 4 dígitos, superando notablemente a los otros modelos probados. Estas métricas son especialmente relevantes, dada la ambigüedad y variabilidad de términos en las descripciones de los agentes aduanales, donde el desempeño de GPT marca una ventaja importante en precisión.

Al elegir un modelo de lenguaje para clasificar códigos HS, un negocio debe considerar riesgos legales, informáticos y tecnológicos, optando por una solución que equilibre estos aspectos con costos razonables y aumente la productividad de los empleados.

1. **Riesgos Legales**: Modelos que requieren transferencia de datos a terceros pueden enfrentar problemas de cumplimiento con regulaciones de privacidad como GDPR o CCPA. Un modelo que se aloje en servidores internos, como Mistral7b, reduce este riesgo al mantener el control de los datos.
2. **Riesgos Informáticos**: Modelos dependientes de servidores externos pueden sufrir interrupciones si hay problemas de conectividad. Implementaciones locales o en nubes privadas aseguran la continuidad operativa.
3. **Riesgos Tecnológicos**: Modelos avanzados como GPT-4o demandan infraestructura costosa y conocimientos especializados. Mistral7b, en cambio, requiere menor infraestructura, integrándose de forma más económica y ágil en sistemas existentes.
4. **Productividad y Costos Balanceados**: Un modelo eficiente y sencillo de usar facilita su adopción, aumentando la productividad de los empleados al reducir tareas repetitivas. Elegir un modelo como Mistral7b permite liberar recursos para otras áreas estratégicas sin sacrificar la precisión en exceso.

Optar por un modelo de bajo riesgo, costo balanceado y fácil de integrar maximiza los beneficios, reduciendo problemas legales y permitiendo que el equipo se enfoque en actividades de mayor valor, mejorando así la competitividad del negocio.

## 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

Lewis, Patrick & Perez, Ethan & Piktus, Aleksandara & Petroni, Fabio & Karpukhin, Vladimir & Goyal, Naman & Küttler, Heinrich & Lewis, Mike & Yih, Wen-tau & Rocktäschel, Tim & Riedel, Sebastian & Kiela, Douwe. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. 10.48550/arXiv.2005.11401.

Afolabi, Olusekemi. (2024). RAG (RETRIEVAL-AUGMENTED GENERATION) AGENTS: REVOLUTIONIZING DECISION-MAKING IN SUPPLY CHAIN MANAGEMENT.

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