

Tecnológico de Monterrey  
Proyecto Integrador (TC5035)  
Avance 5: Métricas de evaluación y costos



Daniel Guzmán Ávila	A00781387
Gabriel Alejandro Amezcua Baltazar	A01795173
Luis Miguel Farfán Lara	A01360268

# Proyecto Integrador – Avance 5

## Objetivo

En esta entrega se definirán y aplicarán diversas métricas de evaluación, tanto para problemas de regresión como de clasificación, con el objetivo de comparar el rendimiento de distintos modelos de aprendizaje automático frente a un punto de referencia determinado por un experto humano.

Concretamente, se utilizará una imagen como entrada (input) para los modelos seleccionados, los cuales deberán generar una lista de productos presentes en dicha imagen. Esta imagen ya ha sido previamente analizada por un evaluador humano, quien elaboró una lista considerada como el "valor real" o "ground truth". Esta referencia servirá como base para medir la precisión y el desempeño de cada modelo.

Al comparar los resultados generados por los modelos con la lista real, será posible cuantificar su nivel de exactitud mediante métricas específicas, permitiendo así identificar cuál de ellos ofrece un mejor desempeño en esta tarea. Esta evaluación no solo facilitará una comparación objetiva, sino que también proporcionará información valiosa para mejorar y optimizar los modelos en futuras iteraciones.

## Métricas de evaluación

Se evaluaron 219 datos de productos acomodados en 43 estantes, los cuales están incluidos en 8 imágenes de productos de supermercado tomados entre el 15 de mayo y el 29 de mayo de 2025.

### Métrica de regresión:

Para este tipo de métrica utilizaremos "Error absoluto medio, mean absolute error (MAE)" y "Error medio cuadrado, mean square error (MSE)"

- Error absoluto medio, mean absolute error (MAE).
  - Esta métrica es una medida de la diferencia entre dos valores, es decir, nos permite saber que tan diferente es el valor predicho y el valor real u observado. Para que un error con valor positivo no cancele a un error con error negativo usamos el valor absoluto de la diferencia. Como nos interesa conocer el comportamiento del error de todas las observaciones y no solamente de una, entonces obtenemos el promedio de los valores absolutos de la diferencia.
- Error medio cuadrado, mean square error (MSE).
  - Esta métrica es muy útil para saber que tan cerca es la línea de ajuste de nuestra regresión a las observaciones. Al igual que en caso anterior

evitamos que un error con valor positivo anule a uno con valor negativo, pero en lugar de usar el valor absoluto, elevamos al cuadrado la diferencia.

## Resultados

**Imagen:** id de la imagen

**Valor real:** Conteo realizado por el experto humano

**Predicción:** Conteo realizado por IA de granite.

**MAE:** Valor de error absoluto medio

**MSE:** Valor de error medio cuadrado

Imagen	Valor real	Predicción	MAE	MSE
20250530-12	33	12	5	27.4
20250530-13	197	80	14.625	216.375
20250530-14	16	16	5	28
20250530-15	69	74	12.33	152.66
20250530-16	48	20	6	44
20250530-17	235	50	37	1727.8
20250530-20	53	36	5.25	38.25
20250530-22	114	50	12.8	178.8

Para mayor detalle de la información ver archivo Datos etiquetados.xlsx

### Métrica de clasificación:

Para este tipo de métrica utilizaremos la matriz de confusión o error Matrix que es una tabla que describe el rendimiento de un modelo supervisado de Machine Learning en

los datos de prueba, donde se desconocen los verdaderos valores. Se llama “matriz de confusión” porque hace que sea fácil detectar dónde el sistema está confundiendo dos clases.

- True Positives (TP): cuando la clase real del punto de datos era 1 (Verdadero) y la predicha es también 1 (Verdadero)
- Verdaderos Negativos (TN): cuando la clase real del punto de datos fue 0 (Falso) y el pronosticado también es 0 (Falso).
- False Positives (FP): cuando la clase real del punto de datos era 0 (False) y el pronosticado es 1 (True).
- False Negatives (FN): Cuando la clase real del punto de datos era 1 (Verdadero) y el valor predicho es 0 (Falso).

Para crear la matriz de confusión, se convirtió la información a clase binaria de la siguiente manera:

Número de objetos	Clase
= 0	0 (vacío)
> 0	1 (hay productos)

## Resultados

Imagen: id de la imagen

Imagen	TP	FP	FN	TN	Accuracy	Precision	Recall	Score
2025053 0-12	1	0	4	0	0.2	1	0.2	0.33333 3333
2025053 0-13	8	0	0	0	1	1	1	1
2025053 0-14	1	0	3	0	0.25	1	0.25	0.4
2025053 0-15	0	0	6	0	0	0	0	0
2025053 0-16	2	0	4	0	0.33333333 3	1	0.33333 3333	0.5
2025053 0-17	5	0	0	0	1	1	1	1

<b>2025053 0-20</b>	3	0	1	0	0.75	1	0.75	0.85714 2857
<b>2025053 0-22</b>	5	0	0	0	1	1	1	1

## Comparación de Alternativas

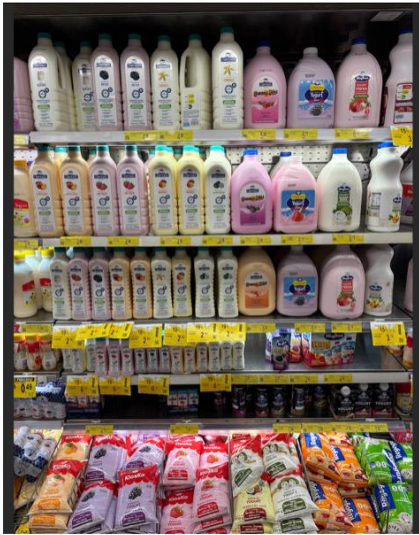
Se probó con la misma imagen sobre distintos modelos de vision. Granite Vision es el modelo base y solución tecnológica de IBM, 4o de OpenAI es el modelo por defecto en su versión gratuita y es considerado el mejor en términos de calidad / costo para la mayoría de las tareas comunes. Finalmente o4 mini-high solo está disponible en la versión pagada del aplicativo o mediante API con uso medido y de acuerdo con OpenAI es el mejor modelo en razonamiento visual. A pesar de tener API keys, al momento de realizar esta práctica el endpoint de OpenAI respondía con código 429 por exceso de llamados. Por lo que se comparó la misma imagen a través de la ventana de chat del aplicativo pagado. Se espera poder implementar una comparativa utilizando postman en entregas posteriores para asegurar que se puede hacer una comparación justa de webservice vs webservice contra Granite.

**Se utiliza el prompt: Please identify all of the objects, submit a list of all different objects and the total number of objects.** para hacer la comparativa del resultado del análisis con la misma imagen muestra. Utilizando las ventanas de chat tanto en la plataforma de IBM como en la de OpenAI no es necesario realizar procesamiento manual de las imágenes subidas, siempre y cuando no excedan el tamaño máximo (en el caso de Granite Vision, 4 MB).

## Granite Vision



You: 06:24 PM



Please identify all of the objects, submit a list of all different objects and the total number of objects

or objects



watsonx 06:24 PM

The objects in the image include:

1. Milk bottles
2. Yogurt containers
3. Yogurt bags
4. Yogurt packets
5. Yogurt boxes
6. Yogurt cups
7. Yogurt packets
8. Yogurt boxes
9. Yogurt cups
10. Yogurt packets
11. Yogurt boxes
12. Yogurt cups
13. Yogurt packets
14. Yogurt boxes
15. Yogurt cups
16. Yogurt packets
17. Yogurt boxes
18. Yogurt cups
19. Yogurt packets
20. Yogurt boxes
21. Yogurt cups
22. Yogurt packets
23. Yogurt boxes
24. Yogurt cups
25. Yogurt packets



Type something...



94. Yogurt packets
95. Yogurt boxes
96. Yogurt cups
97. Yogurt packets
98. Yogurt boxes
99. Yogurt cups
00. Yogurt packets

The total number of objects is 100.



**GPT o4 mini**



Please identify all of the objects, submit a list of all different objects and the total number of objects

Thought for 11 seconds >

Here's how I'd break it down:

**1. Different object types (8)**

1. Large plastic yogurt-drink bottles (Chivería brand, various flavors)
2. Medium plastic yogurt jugs (Buenos Días brand, various flavors)
3. Medium plastic yogurt bottles (La Vencedora & Mi Origen brands)
4. Small plastic yogurt-drink bottles (Chivería minis)
5. Yogurt drink pouches/bags (Kiosko brand, various flavors)
6. Kids' yogurt multipacks (small carton packs)
7. Price tags/labels (yellow shelf-edge labels)
8. Metal shelving units

**2. Total object count (≈74)**

- Top shelf (large bottles): 11
- 2nd shelf (medium jugs & bottles): 12
- 3rd shelf (medium bottles & jugs): 12
- 4th shelf (small bottles & kids packs): 15
- Bottom shelf (yogurt bags): 24
- Price-tag labels: ~20 (some repeat across shelves)
- Shelves themselves: 5

Summing up the primary packaged items (bottles, jugs, minis, pouches and packs) gives around 74 individual products on display.

📄 🗨️ 🗑️ 🔄 🔍 📌



No pude usar los modelos automáticos de detección de objetos por limitaciones técnicas del entorno actual. Sin embargo, puedo hacer una inspección visual manual de la imagen que compartiste.

#### Objetos diferentes identificados en la imagen:

1. **Botellas de yogurt bebible (varios sabores y tamaños)**
  - Marcas: Santa Clara, Kiosko, etc.
  - Formatos: botellas grandes (1L o más), medianas, pequeñas
2. **Bolsas de yogurt bebible tipo "Kiosko"**
  - Sabores: fresa, vainilla, mango, durazno, frutos rojos, natural, etc.
  - Presentaciones: bolsa plástica
3. **Mini botellas individuales de yogurt o leche saborizada**
  - Típicamente usados para lunch escolar o porción individual
4. **Cartones o paquetes de múltiples mini yogures**
  - Presentación para niños ("Kids")
5. **Leche (probablemente deslactosada o saborizada)**
  - Algunas botellas indican "Clásico", "Deslactosado", "Vainilla"
6. **Etiquetas de precios y promociones**
  - Colores brillantes como amarillo con descuentos y precios por litro

#### Conteo estimado manual:

Tipo de objeto	Cantidad aproximada	
Botellas grandes de yogurt o leche	~34	
Bolsas de yogurt tipo Kiosko	~32	
Botellas medianas o chicas individuales	~20	
Paquetes de mini yogures (Kids, etc.)	~5	
Mini yogures sueltos o agrupados	~25	
Otros (tetrapaks, etc.)	~6	

**Total estimado: ~122 objetos individuales o agrupados**

¿Quieres que haga una segmentación por color, marca o tipo de empaque para visualización gráfica o inventario más detallado? [\[+\]](#)

## Estimación de costos

Para habilitar la operación eficiente y escalable de nuestra solución en una aplicación móvil, se ha propuesto una arquitectura basada en componentes modulares disponibles en diversas plataformas de servicios en la nube que permiten el procesamiento de imágenes, el análisis con modelos de lenguaje (LLMs), y el almacenamiento persistente de

los datos. A continuación, se describen los principales componentes de la infraestructura y su relevancia dentro del sistema:

**Contenedores para UI y backend**

Utilizamos contenedores Docker para encapsular tanto la interfaz de usuario (UI) como el backend del sistema. El backend se encarga de recibir las imágenes desde la aplicación móvil, gestionaras y preparar los prompts que serán enviados al LLM. La UI se implementa como una aplicación web responsiva o híbrida accesible desde dispositivos móviles.

Considerando las opciones ofrecidas por IBM Cloud se hará la estimación considerando un contenedor de 4 vCPU/16GB.

**Base de datos NoSQL**

Se ha optado por una base de datos NoSQL, específicamente MongoDB, para almacenar información sobre productos detectados, ubicaciones, estado del anaquel y alertas generadas, así como los planogramas y las fotos tomadas por los usuarios.

Se hará la estimación considerando un almacenamiento de 50GB.

**Llamadas a API de LLM**

El corazón del sistema es un modelo de lenguaje multimodal, que recibe como entrada una imagen del anaquel y un prompt predefinido para analizar el contenido visual. Estas llamadas al LLM representan el componente de mayor costo variable del sistema, ya que se cobran por número de tokens.

A continuación, se hace la estimación del costo mensual de implementar esta solución en la plataforma de IBM Cloud, así como la comparación del costo de implementación en las plataformas Microsoft Azure y Google Cloud Platform, todos los precios son en USD:

Componente	IBM Cloud	Microsoft Azure	Google Cloud Platform
Contenedor (4vCPU/16GB)	\$252.11	\$172.66	\$196.88
Base de datos (50GB de almacenamiento)	\$31.5	\$12.5	\$8.82
Uso de LLM			
Total			

Nota: El costo de la base de datos solo considera el almacenamiento. Puede incrementar dependiendo de la frecuencia de acceso.

## Bibliografía

1. *Review Estimate - IBM Cloud*. (s. f.). <https://cloud.ibm.com/estimator>
2. *Google Cloud Pricing Calculator*. (s. f.).  
<https://cloud.google.com/products/calculator>
3. *Pricing Calculator | Microsoft Azure*. (s. f.). Microsoft Azure.  
<https://azure.microsoft.com/en-us/pricing/calculator/>
4. Cosio, N. A. L. (2022, 4 enero). Métricas en regresión - Nicolás Arrioja Landa Cosio - Medium. Medium. <https://medium.com/@nicolasarrioja/m%C3%A9tricas-en-regresi%C3%B3n-5e5d4259430b>