

Objetivo general:

El modelo desarrollado corresponde a una arquitectura multimodal de regresión diseñada para predecir métricas de engagement turístico. Los datos originales del proyecto corresponden a dos fuentes de información, características visuales extraídas de imágenes y metadatos estructurados asociados a cada POI.

Preparación y análisis de datos.

Esta parte buscaba analizar los atributos de los metadatos, que inicialmente tiene 14 columnas y 1.569 registros y la fuente de imágenes 1.492 elementos. Precisamente ese fue uno de los desafíos posteriores al integrar las dos fuentes de información ya que un mismo ID se repetía varias veces de acuerdo con la cantidad de tags que tuviera. Sobre el análisis de los metadatos realicé algunas transformaciones para que el modelo entendiera mejor los valores de los datos:

Variables originales vs. normalizadas

Durante el proceso de preparación del dataset se trabajó con variables numéricas como **Visits, Likes, Dislikes, Bookmarks y Xps**. Estas variables reflejan valores absolutos de interacción de los usuarios con cada POI. Sin embargo, para fines de modelado predictivo y comparabilidad entre registros, fue necesario aplicar procesos de **normalización** que transforman dichas variables a una escala común.

Justificación

- **Variables originales:** permiten interpretar los datos en su magnitud real. Ejemplo: “este POI recibió 10,000 visitas”.
- **Variables normalizadas:** eliminan diferencias de escala entre variables, lo que evita que el modelo favorezca aquellas con valores más grandes.
- **Decisión metodológica:** se conservaron ambas versiones durante la fase exploratoria, pero en la etapa final de modelado se trabajó únicamente con las normalizadas para reducir redundancia y colinealidad.

Proceso de normalización

Se aplicó la normalización min-max, que transforma cada valor x a un rango entre 0 y 1:

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

Ejemplo:

- Likes originales: 0 a 12,000
- Likes normalizados: 0.0 a 1.0

De esta forma, un POI con 6,000 likes se representa como 0.5 en la escala normalizada.

Dichas transformaciones las incluí como columnas adicionales al dataset inicial.

Información geográfica

El dataset contiene información de latitud y longitud de cada punto de interés, dicha información me pareció relevante para obtener las ubicaciones en términos de ciudades y países donde se encuentra el sitio. Para este caso intenté primero procesar todas las ubicaciones geográficas con una geocodificación inversa, pero el procesamiento era muy alto y demorado, así que decidí agrupar los puntos en clusters por cercanía y con las coordenadas del centroide identificar la ciudad y país. De tal manera, obtuve dos columnas adicionales 'city' y 'country' que se agregaron al dataset.

Creación de métricas de engagement

Además de las variables originales de interacción (**Likes, Dislikes, Bookmarks, Visits, Xps**), se definieron métricas derivadas denominadas **ratios de engagement**. Estas métricas permiten evaluar la calidad relativa de la interacción de los usuarios con cada POI, eliminando sesgos de escala y facilitando comparaciones entre puntos de interés con distinta popularidad.

- Los valores absolutos (ej. número de visitas) tienden a favorecer a los POIs más grandes, ocultando patrones en los más pequeños.
- Los ratios transforman esas cifras en proporciones comparables, lo que permite identificar POIs con **alta calidad relativa** aunque tengan menos volumen de interacción. Se buscó capturar tres dimensiones del engagement: **aceptación o rechazo, intención de retorno** y **profundidad de interacción**.

Definición de métricas

- **Like Ratio**

$$\{like_ratio\} = \{Likes\} / \{\{Likes\} + \{Dislikes\} + \epsilon\}$$

- Descripción: Proporción de interacciones positivas respecto al total de interacciones (likes + dislikes).
- Interpretación: Valores cercanos a 1 indican alta aceptación del POI; valores bajos reflejan mayor proporción de rechazo.
- Uso: Métrica principal de satisfacción/engagement.

- **Dislike Ratio**

$$\{dislike_ratio\} = \{Dislikes\} / \{\{Likes\} + \{Dislikes\} + \epsilon\}$$

- Descripción: Proporción de interacciones negativas respecto al total.
- Interpretación: Valores altos reflejan rechazo significativo.
- Uso: Métrica complementaria a like_ratio para balancear percepción.

- **Bookmark Rate**

$$\{\text{bookmark_rate}\} = \{\text{Bookmarks}\} / \{\text{Visits}\} + \epsilon$$

- Descripción: Porcentaje de visitantes que guardan el POI en sus favoritos.
- Interpretación: Indica el nivel de interés sostenido o intención de volver.
- Uso: Métrica de fidelización o valor percibido

- **XP por visita**

$$\{\text{xp_per_visit}\} = \{\text{Xps}\} / \{\text{Visits}\} + \epsilon$$

- Descripción: Experiencia acumulada por cada visita al POI.
- Interpretación: Cuantifica la intensidad de interacción por visita.
- Uso: Métrica de profundidad de engagement.

- **dislikes_per_visit**

$$\{\text{dislikes_per_visit}\} = \{\text{Dislikes}\} / \{\text{Visits}\} + \epsilon$$

- Descripción: Tasa de rechazo relativa al tráfico recibido.
- Interpretación: Indica qué tan frecuente es la reacción negativa por cada visita.
- Uso: Métrica de riesgo o insatisfacción.

ϵ : Se añade este término de estabilidad = 1e-5 para evitar divisiones por cero.

Conclusión

Los ratios de engagement constituyen un aporte fundamental al dataset, ya que permiten evaluar la interacción de los usuarios desde una perspectiva relativa y comparativa. Su inclusión en el pipeline asegura que los modelos predictivos y los análisis descriptivos capturen no solo la cantidad de interacción, sino también su calidad y profundidad.

Embeddings o vectores de las imagenes

El tratamiento de las imágenes se realizó con modelo pre entrenado de ResNet18, se almacenaron los ebeddings indexados por id para la integración posterior con los metadatos en el modelo multimodal y se verificó que cada POI tuviera su embedding asociado de 512 dimensiones.

Procesamiento de variables categóricas.

Las columnas categories y tags son de gran importancia para ayudar en las predicciones deseadas, sin embargo, su manejo no es sencillo, utilice one-hot encoder inicialmente pero

cada categoría se convertía en una columna lo cual hacía que el dataset creciera en dimensiones demasiado y al final la información así tabulada es más difícil de computar y de entender para el modelo, por eso la decisión fue transformar columnas categóricas complejas (categories, tags) en representaciones numéricas densas y semánticas mediante embeddings. Al final se selecciona un vector de 32 dimensiones para categories y tags, luego se suman y se procesa al final un vector de 64 dimensiones. *Se definieron 32 dimensiones porque es un tamaño intermedio, eficiente y suficiente para capturar relaciones semánticas en un vocabulario moderado de categorías/tags, sin sobrecargar el modelo ni perder capacidad de representación. Además, facilita la integración multimodal al producir un vector final de 64 dimensiones.*

Arquitectura del modelo.

Luego de la explicación sobre las variables tabulares normalizadas, embeddings visuales de 512 dimensiones y representaciones de categorías/tags de 64 dimensiones. Cada modalidad se procesa mediante capas lineales con activación ReLU y regularización por Dropout, y posteriormente se fusionan en un bloque denso para generar la predicción final.

Entrenamiento y Optimización.

Se seleccionó un tamaño oculto de 256 unidades como compromiso entre capacidad de representación y eficiencia, y un único valor de salida para la tarea de regresión (like_ratio).

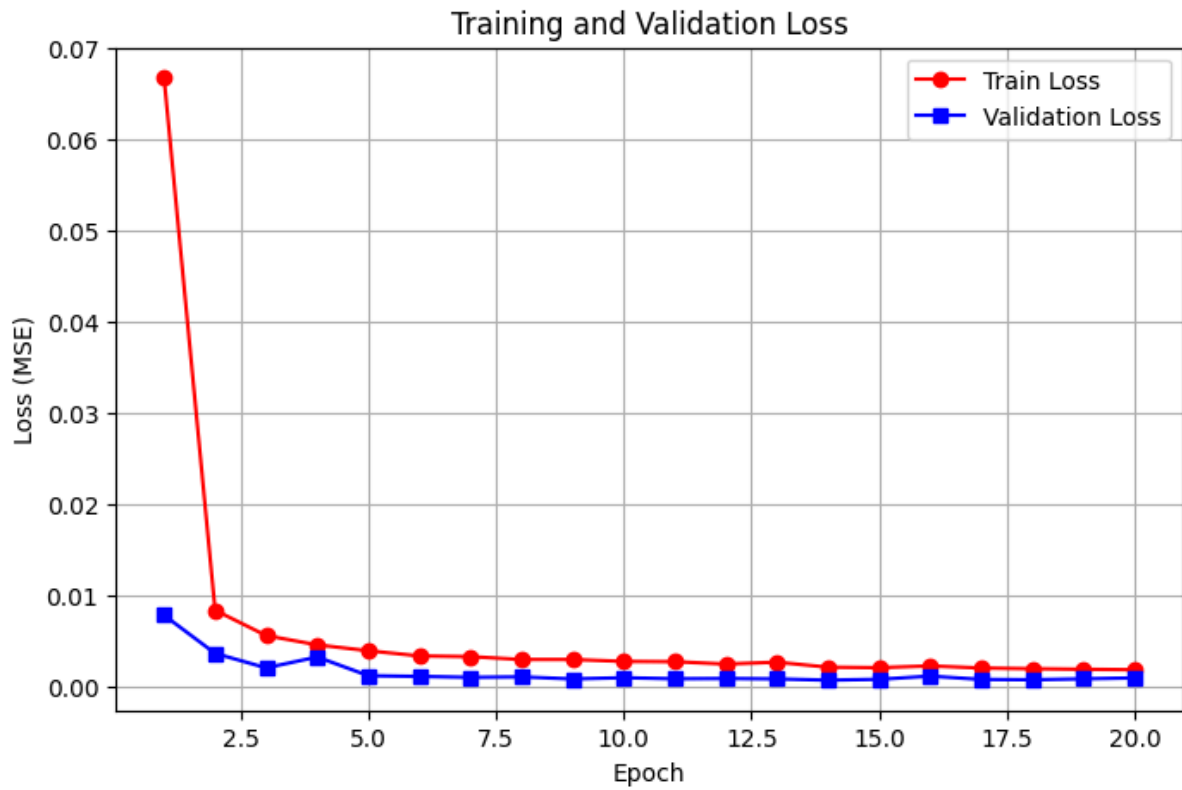
Se dividió el dataset en train (70%), validation (10%) y test (20%). El entrenamiento se realizó con el optimizador Adam ($\text{lr}=1\text{e-}3$) y función de pérdida MSE, durante 20 épocas, número elegido como punto de equilibrio para observar convergencia sin sobreentrenamiento, complementado con validación en cada ciclo. El batch size se fijó en 32 para garantizar estabilidad en el gradiente y eficiencia computacional.

Finalmente, la evaluación en el conjunto de prueba se realizó con métricas interpretables (RMSE, MAE y R^2), confirmando la capacidad del modelo para generalizar y explicar la variabilidad de la variable objetivo con alta precisión.

Evaluación y análisis.

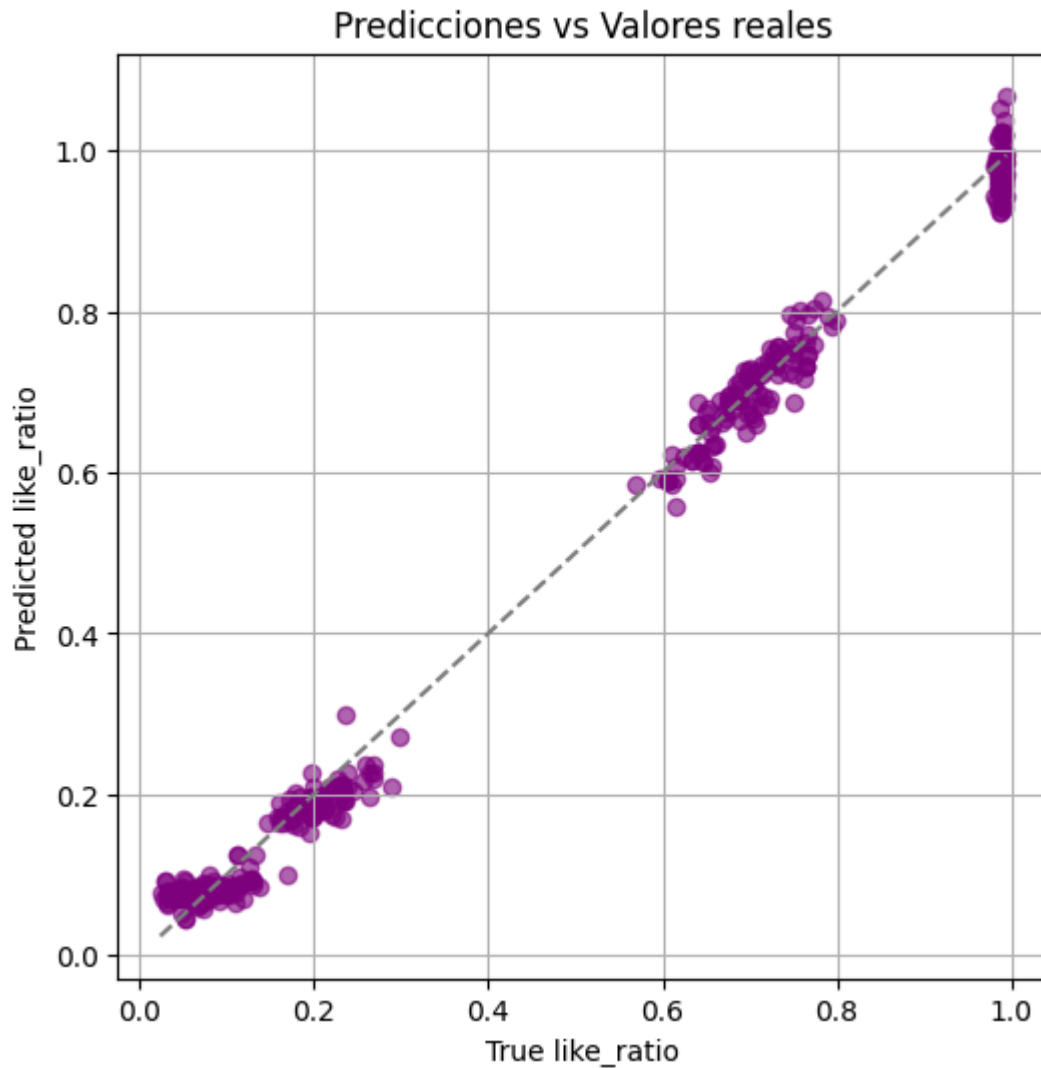
La evaluación final se realizó sobre el conjunto de test, se llegaron a los siguientes resultados.

- RMSE = 0.0270 → error promedio bajo en la escala de like_ratio.
- MAE = 0.0217 → errores absolutos pequeños y consistentes.
- $R^2 = 0.9945$ → el modelo explica más del 99% de la variabilidad de la variable objetivo.



Los resultados confirman que el modelo generaliza correctamente y logra predicciones precisas.

El siguiente scatter plot muestra que las predicciones se alinean a la diagonal ideal, lo que valida la calidad del ajuste.



Conclusiones y otros pasos.

En general es un ejercicio interesante, un poco diferente a los vistos en el módulo pues no se trataba de una decisión binaria si la imagen corresponde o no a una categoría, en donde el accuracy cobra relevancia, sino que es un ejercicio más de Machine Learning de predecir una(s) variable(s) pero incluyendo otra fuente de datos como las imágenes y luego integrando datos y fotos, lo cual realmente lo hace mucho más complejo.

Realmente al revisar los resultados no vi la necesidad de realizar un data augmentation sobre el conjunto de imágenes, si bien las fotos no eran de la mejor calidad, tal parece que fueron suficientes para el modelo.

Por tiempo no alcancé a realizar las salidas sobre más variables, me parece muy pertinente hacerlo y ver como predice otras variables como bookmark_rate o dislike_ratio. Igual ver como pueden influir la calidad de las imágenes en la predicción con un data augmentation.

Por último dejo un Pipeline general de todo lo realizado:

Pipeline Multimodal

1. Preparación de datos

- Cargar df_poi y verificar estructura de columnas.
- Confirmar rutas de imágenes (main_image_path) y existencia en data/data_main/.
- Reportar faltantes para depuración y consistencia.

2. Procesamiento geográfico

- **Clustering geográfico:** aplicar KMeans sobre lat/lon → generar geo_cluster.
- **Centroides y enriquecimiento:** calcular centroid_lat, centroid_lon; añadir city y country vía geocodificación inversa.
- **Persistencia:** guardar cluster_centroids.csv en data/ para reutilizar en reinicios.
- **Merge:** integrar con df_poi para que cada POI tenga city/country asignados.

3. Procesamiento de imágenes

- **Extracción de embeddings visuales:** usar modelo preentrenado (ResNet18, EfficientNet o CLIP).
- **Almacenamiento:** guardar embeddings en data/image_embeddings.npy indexados por id.
- **Verificación:** confirmar que cada POI tiene su embedding asociado (512 dimensiones).

4. Procesamiento tabular

- **Selección de features relevantes:**
 - Engagement: Likes, Dislikes, Bookmarks, ratios (like_ratio, bookmark_rate, xp_per_visit, dislike_ratio, dislikes_per_visit).
 - Contexto: city, country, geo_cluster.
 - Metadatos: tier, categories, visits, xp.
- **Normalización y escalado:** usar columnas *_norm ya calculadas.
- **Consistencia:** verificar reproducibilidad y ausencia de valores nulos

5. Integración multimodal y modelado

- **Unificación de fuentes:** concatenar embeddings visuales (512) + features tabulares (25) + embeddings de categorías/tags (64).
- **Arquitectura del modelo:**
 - Bloques lineales independientes por modalidad.
 - Fusión por concatenación → capas densas con ReLU y Dropout.
 - Salida: regresión (like_ratio) o multitarea (like_ratio, bookmark_rate, xp_per_visit).

- **Entrenamiento y validación:**
 - División en train (70%), validation (10%), test (20%).
 - Optimización: Adam ($lr=1e-3$), MSELoss.
 - Configuración: 20 épocas, batch size = 32.
- **Métricas:** RMSE, MAE, R^2 para regresión.