

Memoria Técnica: Predicción del Éxito de Atracciones Turísticas con Deep Learning Híbrido

Título de la práctica: Predicción del Éxito de Atracciones Turísticas: Un Enfoque de Deep Learning

Autor: Ulises González

Email: gonzalez.ulises@gmail.com

Módulo: Práctica del Módulo de Deep Learning KeepCoding

Fecha: 02 de junio de 2025

Ciudad de Panamá

1. Introducción y Objetivo de la práctica	4
2. Descripción del Dataset	6
2.1. Imágenes	6
2.2. Metadatos	6
2.3. Métricas de Engagement Originales	7
2.4. Selección de Variables para el Modelo	7
3. Metodología	9
3.1. Preparación y Análisis de Datos	9
3.1.1. Carga Inicial y Primer Vistazo	9
3.1.2. Limpieza de Datos: IDs Duplicados y su Racional	9
3.1.3. Análisis Exploratorio de Datos (EDA) - Metadatos y sus Implicaciones (Post-Consolidación)	10
3.1.4. Análisis Exploratorio de Datos (EDA) - Imágenes y su Significado	11
3.1.5. Creación de la Métrica de Engagement (Variable Objetivo): Justificación y Proceso	11
3.1.6. Preprocesamiento Final de Metadatos para el Modelo: Decisiones y Racionales	12
3.1.7. División Estratificada del Dataset: Importancia y Método	13
3.2. Arquitectura del Modelo	13
3.2.1. Pipeline de Datos (Dataset y DataLoaders de PyTorch): Racional de Diseño	13
3.2.2. Diseño del Modelo Híbrido (HybridPoiModel): Fundamentos y Componentes	14
3.2.2.1. Componente Visual (CNN): Elección y Configuración	14

3.2.2.2. Componente Contextual (Metadatos): Estrategias de Procesamiento	14
3.2.2.3. Mecanismo de Fusión y Cabezal de Clasificación: Integración y Predicción	15
3.2.3. Justificación Detallada de las Decisiones Arquitectónicas (Integrada)	15
3.3. Entrenamiento y Optimización	16
3.3.1. Configuración del Entrenamiento: Elección de Pérdida y Optimizador	16
3.3.2. Proceso Experimental y Resultados del Entrenamiento	16
3.3.3. Técnicas Anti-overfitting Aplicadas y su Racional	17
3.4. Evaluación y Análisis	17
3.4.1. Evaluación en Conjunto de Test: Importancia	17
3.4.2. Interpretación de Resultados y Métricas de Rendimiento Detalladas	18
3.4.3. Análisis de Errores y Casos Específicos: Obtención de Insights	18
3.4.4. Interpretabilidad del Modelo y Visualización de Features Importantes: Comprendiendo las Decisiones del Modelo	20
4. Propuestas de Mejoras Futuras	21
5. Conclusiones Finales del Proyecto	24

1. Introducción y Objetivo de la práctica

El presente proyecto se enmarca en la práctica del Módulo de Deep Learning de KeepCoding y tiene como **objetivo principal** el desarrollo e implementación de un modelo avanzado de Deep Learning. Este modelo está diseñado para predecir con precisión el nivel de *engagement* (interacción) que generarán distintos Puntos de Interés (POIs) turísticos. La innovación clave de este proyecto reside en la integración de dos fuentes de información complementarias: las **características visuales** extraídas de las imágenes representativas de cada POI y los **metadatos estructurados** asociados a cada uno.

Los datos utilizados para este estudio provienen de la plataforma **Artgonuts**, lo que garantiza su relevancia y aplicabilidad en contextos reales del sector turístico. Es importante destacar que las imágenes empleadas han sido específicamente procesadas para los fines de esta práctica, siendo sus versiones originales en alta resolución procedentes de diversas fuentes, incluyendo el portal de datos abiertos de la Comunidad de Madrid.

El **desafío central** consiste en desarrollar un sistema capaz de anticipar con exactitud el nivel de interacción que cada POI generará, basándose tanto en sus atributos visuales como en sus metadatos descriptivos. Esta capacidad predictiva representa un activo estratégico significativo para plataformas como Artgonuts, ya que permite, entre otros:

- Optimizar la selección y priorización de contenido para maximizar la interacción del usuario.
- Identificar patrones visuales y características que correlacionan con un mayor engagement.
- Mejorar la experiencia del usuario al destacar contenido más relevante y atractivo.
- Proporcionar *insights data-driven* para la toma de decisiones sobre nuevos POIs.

El modelo desarrollado en este proyecto actuará como un **clasificador binario**, determinando si un punto de interés turístico generará un nivel de engagement **alto o bajo** entre los usuarios. Esta predicción se fundamenta en:

- **Características visuales:** Extraídas mediante Redes Neuronales Convolucionales (CNN) que analizan patrones, colores, composición y elementos distintivos en las imágenes.
- **Metadatos estructurados:** Información contextual como ubicación geográfica, categorización, etiquetas descriptivas y otras métricas asociadas.

Este enfoque híbrido multimodal busca capturar la complejidad del comportamiento del usuario, integrando tanto la respuesta estética inmediata a los elementos visuales como los factores prácticos y contextuales. Se espera que la combinación de ambas fuentes de información en un único modelo proporcione una capacidad predictiva superior a la que se obtendría utilizando cada tipo de datos por separado.

(Nota: Aunque el proyecto se basa en un caso realista con datos de Artgonuts, la empresa no requiere el desarrollo de estos modelos ni empleará sus resultados; su contribución se limita a la cesión de los datos).

2. Descripción del Dataset

El dataset utilizado en este proyecto contiene información detallada sobre Puntos de Interés (POIs) turísticos, proporcionada por la plataforma Artgonuts. Se estructura en dos componentes principales: imágenes representativas de cada POI y un conjunto de metadatos asociados.

2.1. Imágenes

Cada POI en el dataset incluye una imagen principal que lo representa visualmente. Según la documentación del proyecto, estas imágenes:

- Están almacenadas en carpetas individuales, donde cada carpeta está identificada por el ID único del POI correspondiente.
- La ruta específica a la imagen de cada POI se encuentra en la columna `main_image_path` del archivo de metadatos.
- Han sido "específicamente procesadas para los fines de esta práctica". Durante nuestro análisis exploratorio, confirmamos que las imágenes proporcionadas tienen dimensiones uniformes de **128x128 píxeles** y están en modo de color **RGB**. Se identificó que el nombre de archivo de imagen dentro de cada carpeta de ID es consistentemente `main.jpg`. El hecho de que las imágenes ya estuvieran estandarizadas en tamaño y modo de color simplificó significativamente el preprocesamiento visual, permitiendo enfocar los esfuerzos en la normalización de píxeles y la extracción de características.

2.2. Metadatos

La información contextual de cada POI se proporciona en un archivo CSV (`poi_dataset.csv`). Tras un proceso de limpieza y consolidación (detallado en la sección 3.1.2), se trabajó con un conjunto de **1492 POIs únicos**, cada uno con las siguientes características principales:

- **id**: Identificador único alfanumérico para cada punto de interés. Se aseguró su unicidad para garantizar que cada entrada del modelo correspondiera a un POI

distinto.

- **name:** Nombre descriptivo del POI.
- **shortDescription:** Breve descripción textual del POI.
- **categories (procesada como categories_parsed):** Lista de strings que indica el tipo de atracción o temática (ej. ['Patrimonio', 'Cultura']). Se identificaron 12 categorías únicas.
- **tier:** Clasificación numérica (valores de 1 a 4) que indica el nivel de relevancia o popularidad del POI.
- **locationLon y locationLat:** Coordenadas geográficas (longitud y latitud) del POI.
- **tags (procesada como tags_parsed):** Lista de strings con etiquetas descriptivas asociadas al POI (ej. ['Museos', 'Historia']). Se identificó una alta cardinalidad con 2935 tags únicos.
- **xps:** Valor numérico representando la "experiencia" obtenida por el usuario al visitar el POI.
- **main_image_path:** Ruta relativa a la imagen principal del POI, fundamental para enlazar los metadatos con su componente visual.

Nota: Las columnas distrito y barrio, aunque mencionadas en la descripción general del dataset en el enunciado de la práctica, no se encontraron en el archivo CSV proporcionado y, por lo tanto, no se utilizaron. Su ausencia implicó que el análisis de ubicación se basara principalmente en las coordenadas geográficas.

2.3. Métricas de Engagement Originales

Estas columnas del dataset original sirvieron de base para crear la variable objetivo:

- **Visits:** Número de visitas registradas.
- **Likes:** Cantidad de valoraciones positivas.
- **Dislikes:** Cantidad de valoraciones negativas.
- **Bookmarks:** Número de veces que el POI ha sido añadido a favoritos.

2.4. Selección de Variables para el Modelo

Como permite el enunciado, se seleccionó un subconjunto de estas características para el desarrollo del modelo. La **racional** detrás de esta selección fue enfocarse en aquellas variables que, tras el análisis exploratorio, se consideraron más informativas y manejables para predecir el engagement, evitando una complejidad innecesaria. Por ejemplo, name y shortDescription se excluyeron del modelo híbrido inicial para simplificar, aunque se reconocen como fuentes potenciales de información para futuras mejoras mediante NLP. La justificación del preprocesamiento específico de las variables seleccionadas se detalla en la sección de Metodología.

3. Metodología

En esta sección se detalla el proceso completo seguido para el desarrollo del modelo de predicción de engagement, desde la preparación inicial de los datos hasta la evaluación final del modelo entrenado, enfatizando los racionales detrás de las decisiones tomadas.

3.1. Preparación y Análisis de Datos

Esta fase inicial fue fundamental para comprender la naturaleza de los datos, limpiarlos y transformarlos en un formato adecuado para el entrenamiento del modelo de Deep Learning.

3.1.1. Carga Inicial y Primer Vistazo

El proyecto comenzó con la carga del archivo de metadatos `poi_dataset.csv`. El dataset inicial constaba de 1569 filas y 14 columnas. Una primera inspección (`df_meta.info()`) reveló los tipos de datos y sugirió una ausencia inicial de valores nulos directos, indicando una buena calidad de datos base.

3.1.2. Limpieza de Datos: IDs Duplicados y su Racional

Un hallazgo crítico fue la presencia de IDs duplicados: 1492 IDs únicos para 1569 filas, con 23 IDs apareciendo múltiples veces. Se observó que estas filas con el mismo id eran casi idénticas, diferenciándose principalmente en la columna `tags` y, en menor medida, en las métricas de engagement.

Racional de la Decisión: Para asegurar que cada POI físico/conceptual fuera representado por una única entrada y evitar que el modelo se confundiera o sesgara por múltiples entradas del "mismo" ítem con ligeras variaciones, se decidió **consolidar estas filas**. El objetivo era tener una representación única y agregada para cada POI.

- **Estrategia de Agregación:**

- Columnas consistentes (name, categories, etc.) tomaron el 'first' valor.
- xps y métricas de engagement (Visits, Likes, etc.) se promediaron ('mean') para obtener un valor representativo del engagement general del POI.
- tags se consolidaron combinando todos los tags únicos asociados a un id en una sola lista (usando ast.literal_eval para parsear los strings de listas y set para obtener la unicidad).
- **Resultado:** Un DataFrame limpio de **1492 POIs únicos**. Se confirmó que no había valores faltantes tras la consolidación.

3.1.3. Análisis Exploratorio de Datos (EDA) - Metadatos y sus Implicaciones (Post-Consolidación)

- **categories y tags:** Ambas fueron parseadas a listas de Python (categories_parsed, tags_parsed).
 - categories_parsed: Mostró **baja cardinalidad** (12 categorías únicas). *Implicación:* Esto permitió considerar técnicas de encoding simples como one-hot encoding, aunque se optó por embeddings de baja dimensión (dim=10) por consistencia y para capturar posibles relaciones, siguiendo la sugerencia del enunciado de experimentar.
 - tags_parsed: Reveló **alta cardinalidad** (2935 tags únicos). *Implicación:* Esto descartó el one-hot encoding, como advierte el enunciado, y apuntó directamente a la necesidad de usar capas de Embedding (dim=50, vocab_size=1000) para una representación eficiente.
- **Métricas de Engagement y Visits:**
 - Likes y Bookmarks mostraron buena dispersión, haciéndolas buenas candidatas para la métrica de engagement.
 - Visits mantuvo un rango extremadamente estrecho. *Implicación:* Se consideró poco informativa para diferenciar el engagement y se excluyó de la creación del score de engagement, simplificando el modelo y evitando introducir una característica con poca señal.
- **Ubicación Geográfica (locationLon, locationLat):**

- Se observó una concentración del ~70% en Madrid. Las columnas distrito y barrio no estaban presentes. *Implicación:* El análisis de ubicación se basó en coordenadas, y la ausencia de distrito/barrio se documentó como una limitación del dataset provisto frente a su descripción.
- **Conclusión del EDA de Metadatos:** Este análisis fue clave para identificar problemas (IDs duplicados, Visits), entender la naturaleza de las características (categories, tags) y tomar decisiones informadas para el preprocesamiento y la selección de características.

3.1.4. Análisis Exploratorio de Datos (EDA) - Imágenes y su Significado

- **Accesibilidad y Formato:** Se verificó que el 100% de los POIs tenían una ruta de imagen válida (data_main/POI_ID/main.jpg).
- **Propiedades:** Todas las imágenes de muestra eran de **128x128 píxeles** y modo **RGB**.
- *Significado:* El hecho de que las imágenes ya estuvieran estandarizadas (como indicaba el enunciado) fue una ventaja significativa, eliminando la necesidad de un complejo pipeline de redimensionamiento y conversión de modo de color, y permitiendo enfocar el preprocesamiento visual en la normalización.

3.1.5. Creación de la Métrica de Engagement (Variable Objetivo): Justificación y Proceso

El enunciado requería una variable objetivo binaria (alto/bajo engagement) sintetizando las métricas disponibles, con justificación.

1. **Selección de Métricas Base y Justificación:** Se seleccionaron Likes y Bookmarks. *Racional:* Son indicadores directos de la apreciación positiva (Likes) y el interés de visitar/guardar (Bookmarks), alineándose con el concepto de "engagement exitoso". Visits se excluyó por su bajo poder discriminatorio. Dislikes se excluyó por no representar necesariamente "éxito" y para simplificar la definición inicial del target.
2. **Transformación y Justificación:** Se aplicó np.log1p a Likes y Bookmarks.

Racional: Las métricas de conteo suelen tener distribuciones muy sesgadas; la transformación logarítmica ayuda a normalizar estas distribuciones, reduciendo el impacto de outliers y haciendo que las métricas sean más comparables.

3. **Escalado y Justificación:** Se escalaron al rango [0, 1] con MinMaxScaler.

Racional: Esto asegura que ambas métricas contribuyan de forma equitativa al score combinado, independientemente de sus magnitudes originales.

4. **Combinación (Score de Engagement) y Justificación:** Se creó un engagement_score promediando las dos métricas procesadas. *Racional:* Un promedio simple es un punto de partida transparente y fácil de interpretar para combinar las señales de engagement.

5. **Binarización y Justificación:** Se usó la mediana del engagement_score como umbral. *Racional:* Usar la mediana como umbral es una estrategia común para intentar obtener clases de tamaño similar (balanceadas), lo cual es beneficioso para el entrenamiento de modelos de clasificación.

Conclusión de la Creación de la Métrica: Se obtuvo una variable target_engagement binaria con clases perfectamente balanceadas (50%/50%), ideal para el entrenamiento. La metodología fue transparente y justificada en cada paso.

3.1.6. Preprocesamiento Final de Metadatos para el Modelo: Decisiones y Racionales

- **Características Numéricas Seleccionadas:** (tier, locationLon, locationLat, xps, num_categories_parsed, num_tags_parsed). Se escalaron con StandardScaler. *Racional:* El escalado estándar (media 0, std 1) es beneficioso para muchos algoritmos de ML, incluyendo redes neuronales, ya que ayuda a que el optimizador converja más eficientemente.
- **categories_parsed:** Se tokenizaron (vocabulario de 14) y padearon a longitud 5. *Racional:* Convertir a secuencias de enteros es necesario para las capas de Embedding. El padding asegura una longitud de entrada uniforme.
- **tags_parsed:** Se tokenizaron (vocabulario de los 1000 más frecuentes) y

padearon a longitud 15. Racional: Similar a las categorías, pero limitando el vocabulario para tags debido a su alta cardinalidad, como una forma de manejar la complejidad y el tamaño de la capa de Embedding, siguiendo la recomendación del enunciado.

Conclusión del Preprocesamiento: Los metadatos quedaron listos para ser consumidos por el modelo, con numéricos escalados y textuales/categoricos listos para embeddings.

3.1.7. División Estratificada del Dataset: Importancia y Método

Se dividió el dataset en entrenamiento (60%), validación (20%) y prueba (20%) usando `train_test_split` de forma estratificada respecto a `target_engagement`.

Importancia: La estratificación asegura que la proporción original de clases (50/50) se mantenga en cada conjunto. Racional: Esto es crucial para evitar sesgos en el entrenamiento y para que la evaluación en los conjuntos de validación y prueba sea representativa del problema original.

Conclusión de la División: Se obtuvieron particiones de datos adecuadas y balanceadas.

3.2. Arquitectura del Modelo

Se diseñó un modelo híbrido en PyTorch para procesar información visual y contextual.

3.2.1. Pipeline de Datos (Dataset y DataLoaders de PyTorch): Racional de Diseño

- **Transformaciones de Imágenes:** Se usó `transforms.ToTensor()` (para convertir a tensor y escalar a $[0,1]$) y `transforms.Normalize()` con medias/std de ImageNet.
Racional: ToTensor es estándar. La normalización de ImageNet es una práctica común y recomendada cuando se utiliza una CNN pre-entrenada en ImageNet, ya que el modelo espera datos distribuidos de manera similar a como fue entrenado.
- **Clase PoiDataset Personalizada:** Se implementó para cargar y transformar una muestra (imagen y metadatos) dado un índice. *Racional:* Permite un manejo flexible de datos multimodales y la aplicación de transformaciones específicas. Incluyó manejo de errores para imágenes corruptas (devolviendo un tensor de

ceros) para no interrumpir el entrenamiento.

- **DataLoaders:** Se usaron para el batching (BATCH_SIZE = 32), shuffling para entrenamiento, y carga multiproceso. *Racional:* El batching es esencial para el entrenamiento eficiente de redes neuronales. El shuffling ayuda al modelo a generalizar mejor. num_workers acelera la carga de datos.

Conclusión del Pipeline de Datos: Se estableció un sistema robusto para alimentar datos al modelo.

3.2.2. Diseño del Modelo Híbrido (HybridPoiModel): Fundamentos y Componentes

3.2.2.1. Componente Visual (CNN): Elección y Configuración

- **Base Convolutacional:** Se eligió ResNet18 con pesos pre-entrenados en ImageNet. *Racional:* ResNet18 ofrece un buen compromiso entre capacidad de representación y eficiencia computacional. El transfer learning (usar pesos pre-entrenados) es una técnica poderosa que permite aprovechar el conocimiento de características visuales genéricas aprendidas en un dataset masivo (ImageNet), lo cual es especialmente útil cuando el dataset específico del problema (como el de POIs) no es gigantesco.
- **Congelación de Pesos:** Inicialmente, se congelaron todos los parámetros de ResNet18. *Racional:* Para preservar los pesos pre-entrenados y entrenar solo las capas añadidas, acelerando el entrenamiento inicial y reduciendo el riesgo de sobreajuste al principio.
- **Procesador de Características Visuales:** La salida de avgpool de ResNet18 (512 características) se procesó con una capa Lineal, ReLU y Dropout para obtener 256 características visuales. *Racional:* Esta capa adicional permite adaptar y refinar las características extraídas por ResNet18 a la tarea específica y a la dimensionalidad deseada para la fusión.

3.2.2.2. Componente Contextual (Metadatos): Estrategias de Procesamiento

- **Características Numéricas:** Procesadas por una MLP (Lineal, ReLU, Dropout) para extraer patrones. *Racional:* Una MLP simple puede capturar relaciones no

lineales entre las características numéricas.

- **categories_parsed:** Se usó nn.Embedding (vocab=14, dim=10) seguido de mean(dim=1) y una MLP. *Racional:* Aunque la cardinalidad es baja, el embedding permite aprender una representación densa para cada categoría. El promedio es una forma simple de obtener un vector de tamaño fijo a partir de múltiples categorías por POI.
- **tags_parsed:** Se usó nn.Embedding (vocab=1000, dim=50) seguido de mean(dim=1) y una MLP. *Racional:* Con alta cardinalidad, el embedding es esencial. Se limitó el vocabulario a los 1000 tags más frecuentes para manejar la complejidad. Una dimensión de embedding de 50 se consideró un buen punto de partida.
- **Combinación y MLP Final de Metadatos:** Las salidas de las sub-ramas se concatenaron y pasaron por otra MLP. *Racional:* Esta MLP final permite que el modelo aprenda interacciones entre los diferentes tipos de metadatos procesados antes de la fusión con las características visuales.

3.2.2.3. Mecanismo de Fusión y Cabezal de Clasificación: Integración y Predicción

- **Fusión:** Las características visuales (256) y de metadatos (128) se **concatenaron**. *Racional:* La concatenación es el método de fusión más directo y común, permitiendo que las capas subsiguientes aprendan de la combinación de ambas modalidades.
- **Cabezal de Clasificación:** Una MLP (dos capas densas con ReLU y Dropout) tomó el vector fusionado (384 dims) y produjo un único logit. *Racional:* Esta MLP final aprende a mapear las características fusionadas de alto nivel a la predicción de engagement. La salida de un solo logit es estándar para la clasificación binaria con BCEWithLogitsLoss.

3.2.3. Justificación Detallada de las Decisiones Arquitectónicas (Integrada)

Las decisiones sobre el número de capas, neuronas, dimensiones de embedding y tasas de dropout se tomaron buscando un equilibrio entre la capacidad del modelo

para aprender patrones complejos y el riesgo de sobreajuste, comenzando con arquitecturas relativamente estándar y efectivas. El uso de ReLU como activación y Dropout (tasa 0.3) para regularización son prácticas comunes y recomendadas. El diseño modular con ramas separadas para cada tipo de input y luego su fusión permite un procesamiento especializado y una integración flexible.

Conclusión de la Arquitectura: Se diseñó e implementó un modelo híbrido completo, con cada componente y decisión justificados en función de las características de los datos, las directrices del enunciado y las buenas prácticas en Deep Learning.

3.3. Entrenamiento y Optimización

3.3.1. Configuración del Entrenamiento: Elección de Pérdida y Optimizador

- **Función de Pérdida:** Se eligió `nn.BCEWithLogitsLoss`. *Racional:* Es la función de pérdida estándar y numéricamente estable para problemas de clasificación binaria cuando la salida del modelo es un logit (antes de la sigmoide).
- **Optimizador:** Se seleccionó Adam. *Racional:* Adam es un optimizador adaptativo que suele funcionar bien en una amplia gama de problemas con poca necesidad de ajuste de hiperparámetros iniciales, y a menudo converge más rápido que SGD tradicional. Se usó una tasa de aprendizaje inicial de 0.001.
- **Dispositivo:** El entrenamiento se realizó en cuda (GPU) para eficiencia.

3.3.2. Proceso Experimental y Resultados del Entrenamiento

Se entrenó el modelo durante 10 épocas, monitorizando pérdida y precisión en entrenamiento y validación. Se implementó "model checkpointing" para guardar el modelo con la mejor `val_loss`.

- **Resultados:** La pérdida de entrenamiento disminuyó de 0.6224 a 0.1251, y la precisión aumentó de 65.21% a 95.08%. En validación, la pérdida mínima fue 0.2497 y la precisión máxima 91.64% (ambas en la época 10). Se manejó un error de carga de una imagen corrupta haciendo que el `PoiDataset` devolviera un tensor de ceros.

Conclusión del Entrenamiento: El modelo aprendió efectivamente, mejorando consistentemente en ambos conjuntos. El rendimiento superó al de un clasificador aleatorio.

3.3.3. Técnicas Anti-overfitting Aplicadas y su Racional

1. **Dropout (tasa 0.3):** Se usó en varias capas. *Racional:* Previene la co-adaptación excesiva de neuronas, forzando a la red a aprender representaciones más robustas.
2. **Monitorización del Rendimiento en Validación y Model Checkpointing:** *Racional:* Evaluar en un conjunto de validación no visto durante el ajuste de gradientes permite detectar el inicio del sobreajuste. Guardar el modelo que mejor generaliza en validación es una forma efectiva de obtener el mejor modelo posible antes de que el sobreajuste degrade el rendimiento en datos nuevos.
3. **Uso de CNN Pre-entrenada con Capas Congeladas:** *Racional:* Reduce el número de parámetros a entrenar desde cero, aprovechando el conocimiento previo y limitando la capacidad del modelo de sobreajustarse al dataset específico, especialmente en las primeras etapas del entrenamiento.

Conclusión de Técnicas Anti-overfitting: Estas estrategias contribuyeron a un aprendizaje generalizable.

3.4. Evaluación y Análisis

3.4.1. Evaluación en Conjunto de Test: Importancia

Se cargó el mejor modelo (basado en val_loss) y se evaluó en el test_loader.

- **Resultados:** Pérdida en Test: 0.1770, Precisión en Test: 93.31% (o 93.65% según classification_report).

Importancia: La evaluación en el conjunto de prueba proporciona la medida más imparcial del rendimiento del modelo en datos completamente nuevos, siendo el indicador final de su capacidad de generalización.

Conclusión de la Evaluación en Test: El modelo generalizó excelentemente, con un

rendimiento en test incluso superior al de validación, indicando robustez.

3.4.2. Interpretación de Resultados y Métricas de Rendimiento Detalladas

- **Matriz de Confusión:** [[145 TN, 5 FP], [14 FN, 135 TP]]. Mostró pocos Falsos Positivos y un número ligeramente mayor pero aún bajo de Falsos Negativos.
- **Reporte de Clasificación:** Precision, Recall y F1-score altos para ambas clases (ej. F1-score promedio ~0.9364). Precision para "Alto Engagement" fue del 96.43%.

- **Curva ROC y AUC: AUC de 0.9755.**

Interpretación: El modelo es altamente efectivo y fiable. La alta precisión para "Alto Engagement" es valiosa para destacar contenido. El AUC excepcional confirma un gran poder discriminatorio. Los Falsos Negativos representan el principal "coste" en términos de oportunidades perdidas.

3.4.3. Análisis de Errores y Casos Específicos: Obtención de Insights

Se realizó una inspección cualitativa de una muestra de los **5 Falsos Positivos** y los **14 Falsos Negativos** identificados en el conjunto de prueba. El objetivo de este análisis es ir más allá de las métricas agregadas y entender cualitativamente por qué el modelo falla en ciertos casos, lo que puede revelar limitaciones o áreas de mejora no evidentes en los números.

- **Para Falsos Positivos (Modelo predijo "Alto Engagement", pero el real era "Bajo Engagement"):**
 - Una observación recurrente fue que algunos de estos POIs presentaban **imágenes visualmente muy atractivas o profesionales**, con buena iluminación, composición o sujetos interesantes. Esto podría haber llevado al componente visual (CNN) del modelo a asignarles un peso positivo fuerte.
 - En otros casos, los POIs tenían **categorías (categories_parsed) o tags (tags_parsed) que comúnmente se asocian con alto**

engagement en el resto del dataset (ej. "Patrimonio", "Cultura", o tags populares como "Madrid"). El componente contextual podría haber sido influenciado por estas asociaciones.

- La discrepancia podría surgir porque, a pesar de estas características aparentemente positivas, los **valores reales de Likes y Bookmarks (que usamos para crear el target_engagement) eran bajos**. Esto podría deberse a factores no capturados por el modelo, como:
 - Ser un POI muy nuevo con pocas interacciones acumuladas aún.
 - Una descripción textual (`shortDescription`, que no usamos directamente) poco atractiva o informativa.
 - Competencia muy alta en su nicho específico.
 - Que el `engagement_score` calculado para estos POIs estuviera muy cerca del umbral de decisión, y pequeñas variaciones en las características lo empujaron al lado incorrecto de la predicción.
- **Para Falsos Negativos (Modelo predijo "Bajo Engagement", pero el real era "Alto Engagement"):**
- Algunos de estos POIs presentaban **imágenes que no eran particularmente espectaculares visualmente** (ej. fotos más sencillas, menos artísticas, o de objetos/lugares que no son universalmente llamativos a primera vista). Esto podría haber llevado a una contribución neutral o incluso ligeramente negativa del componente visual.
- En ciertos casos, las **categorías o tags asociados no eran los más populares** o los que el modelo había aprendido a asociar fuertemente con alto engagement.
- El alto engagement real de estos POIs podría deberse a:
 - Un **nicho de interés muy específico** pero con una comunidad muy activa que el modelo no capturó con las características disponibles.
 - Una **excelente reputación local, valor práctico, o "joyas ocultas"** cuyo atractivo no se refleja directamente en la imagen o en los

metadatos más genéricos.

- Que su `engagement_score` calculado estuviera justo por debajo del umbral, siendo casos límite.
- Contenido textual en `shortDescription` (no usado) que fuera excepcionalmente bueno o que generara mucha curiosidad.

Conclusión del Análisis de Errores: Este análisis cualitativo, aunque basado en una muestra, sugiere que los errores del modelo no son completamente aleatorios. Los Falsos Positivos a menudo parecen tener características superficialmente atractivas (visual o categóricamente) que no se traducen en un alto engagement medido. Los Falsos Negativos, por otro lado, podrían ser POIs cuyo atractivo es menos obvio para el modelo basado en las features actuales, o cuyo engagement es impulsado por factores más sutiles o de nicho. Este entendimiento ayuda a formular hipótesis sobre las debilidades del modelo y a guiar futuras iteraciones, como la incorporación de más características (ej. análisis de texto de descripciones) o el ajuste del umbral de decisión si se quisiera priorizar la reducción de un tipo específico de error.

3.4.4. Interpretabilidad del Modelo y Visualización de Features Importantes:

Comprendiendo las Decisiones del Modelo

- **Importancia de Metadatos (Permutation Importance):**
 - `category_features` fueron las más influyentes (caída AUC de 0.2346).
 - `numerical_features` tuvieron impacto moderado (caída AUC de 0.0032).
 - `tag_features` tuvieron el menor impacto (caída AUC de 0.0020).
 - *Racional de usar Permutation Importance:* Es una técnica agnóstica al modelo que mide la importancia de una característica observando cuánto se degrada el rendimiento si se rompe la relación entre esa característica y el target. Es útil para obtener una visión general de la relevancia de diferentes grupos de entrada.
 - *Conclusión:* La categorización principal del POI es el factor de metadatos más determinante.

4. Propuestas de Mejoras Futuras

Basándome en el desarrollo y los resultados de este proyecto, identifico las siguientes áreas como posibles mejoras futuras para optimizar aún más el modelo y su comprensión:

1. **Fine-Tuning Avanzado del Componente Visual (CNN):**

- Descongelar progresivamente más capas de la ResNet18 (o experimentar con arquitecturas CNN base más grandes como ResNet34 o EfficientNet) y continuar el entrenamiento con una tasa de aprendizaje muy baja (ej., $1e-5$ a $1e-4$). Esto podría permitir que el modelo adapte más finamente las características visuales al dominio específico de las imágenes de POIs turísticos.

2. **Optimización Exhaustiva de Hiperparámetros:**

- Emplear técnicas de búsqueda de hiperparámetros más sistemáticas (ej., Grid Search, Random Search, o herramientas como Optuna) para explorar un rango más amplio de configuraciones para la tasa de aprendizaje, schedulers, batch size, dimensiones de embedding para categorías y tags, arquitectura de las MLPs (número de capas, neuronas por capa), y tasas de Dropout.

3. **Incorporación de Características Textuales de name y shortDescription:**

- Actualmente, el contenido textual de name y shortDescription no se utiliza directamente como característica de entrada. Se podrían explorar técnicas de Procesamiento de Lenguaje Natural (NLP) para extraer información semántica de estos textos. Esto podría incluir:
 - Embeddings de palabras promediados (ej., Word2Vec, GloVe pre-entrenados).
 - Redes Recurrentes (LSTM/GRU) sobre secuencias de embeddings de palabras.
 - Modelos basados en Transformers pre-entrenados (ej., Sentence-BERT o

similares) para obtener embeddings de las descripciones completas, que luego se podrían integrar en la rama contextual.

4. **Revisión de la Métrica de Engagement y la Columna Visits:**

- Investigar más a fondo el origen y la naturaleza de la columna Visits debido a su rango de valores extremadamente estrecho. Si se pueden obtener datos de visitas más representativos o si se puede derivar una característica más útil de ella, podría reconsiderarse su inclusión.
- Experimentar con diferentes formas de combinar Likes, Bookmarks (y potencialmente Visits si se resuelve su problema) o diferentes umbrales para la binarización de target_engagement.

5. **Manejo Avanzado de Características de Ubicación:**

- Dado que no se encontraron las columnas distrito y barrio, y que existe una mezcla de POIs de Madrid y globales, se podría:
 - Intentar obtener datos de distrito/barrio para los POIs de Madrid a partir de sus coordenadas (locationLon, locationLat) utilizando servicios de geocodificación inversa (si las políticas de uso de datos lo permiten).
 - Crear características de ingeniería como la distancia a puntos de referencia clave en Madrid (para los POIs de esa zona).
 - Considerar una característica binaria "es_madrid" o incluso explorar modelos separados o adaptados para los POIs de Madrid versus los "globales", si se sospecha que sus patrones de engagement difieren significativamente.

6. **Data Augmentation para Imágenes:**

- Aunque las imágenes ya estaban preprocesadas a un tamaño uniforme, aplicar técnicas de aumento de datos ligeros durante el entrenamiento (ej., flips horizontales aleatorios, pequeñas rotaciones, ligeros ajustes de brillo/contraste) podría ayudar a que el componente visual del modelo sea más robusto a variaciones menores y a reducir el riesgo de sobreajuste, especialmente si se realiza un fine-tuning más extenso de la CNN.

7. Experimentación con Arquitecturas de Fusión y Mecanismos de Atención:

- Explorar mecanismos de fusión más sofisticados que la simple concatenación. Por ejemplo, se podrían implementar capas de atención que permitan al modelo ponderar dinámicamente la importancia relativa de las características visuales frente a las de metadatos (o incluso entre diferentes tipos de metadatos) para cada predicción individual.

8. Análisis de Interpretabilidad Más Profundo y Continuo:

- Además de Grad-CAM y Permutation Importance, se podrían explorar otras técnicas como LIME o SHAP (aunque SHAP puede ser computacionalmente costoso para modelos de Deep Learning) para obtener una comprensión más granular de la importancia de las características individuales, tanto visuales como de metadatos.

Estas propuestas ofrecen un camino para continuar mejorando el rendimiento, la robustez y la comprensión del modelo de predicción de engagement.

5. Conclusiones Finales del Proyecto

El desarrollo de este proyecto ha culminado con la creación exitosa de un modelo híbrido de Deep Learning capaz de predecir el nivel de engagement de Puntos de Interés Turístico con un alto grado de precisión y fiabilidad. Los resultados obtenidos en el conjunto de prueba, con una **precisión superior al 93%** y un **AUC de 0.9755**, demuestran la efectividad del enfoque multimodal que integra información visual de imágenes y datos contextuales de metadatos.

A lo largo del proyecto, se ha seguido una metodología rigurosa que abarcó desde un análisis exploratorio exhaustivo y un preprocesamiento cuidadoso de los datos, hasta el diseño e implementación de una arquitectura de red neuronal compleja, su entrenamiento y una evaluación detallada. Los principales hitos y aprendizajes incluyen:

- La importancia de la **limpieza de datos inicial**, como la consolidación de IDs duplicados, para asegurar la calidad del dataset.
- La confirmación de que las **imágenes estaban preprocesadas** a un tamaño y modo de color uniformes (128x128, RGB), lo que simplificó el pipeline visual.
- La necesidad de **estrategias de encoding diferenciadas** para metadatos categóricos, utilizando embeddings para características de alta cardinalidad como tags_parsed (2935 únicos) y considerando opciones más simples para baja cardinalidad como categories_parsed (12 únicas).
- El desarrollo de una **métrica de engagement binaria y balanceada** (target_engagement) a partir de Likes y Bookmarks, lo cual fue crucial para entrenar un clasificador robusto.
- La implementación de un **modelo híbrido en PyTorch** que combina una CNN pre-entrenada (ResNet18) para la extracción de características visuales con múltiples ramas de MLP y embeddings para procesar los metadatos, fusionando ambas fuentes de información para la predicción final.
- La obtención de un **rendimiento que supera claramente a un clasificador**

aleatorio y que generaliza bien a datos no vistos, como lo demuestran las métricas en el conjunto de prueba.

- La aplicación de técnicas de **interpretabilidad** como Permutation Importance, que reveló que las `categories_parsed` son el grupo de metadatos más influyente para el modelo actual.

Este proyecto no solo cumple con los objetivos técnicos planteados en el enunciado, como el desarrollo de un modelo funcional y la justificación de las decisiones, sino que también subraya el potencial de los enfoques híbridos de Deep Learning para resolver problemas complejos en dominios como el turismo. La capacidad de predecir el engagement puede ser una herramienta estratégica valiosa para plataformas que buscan optimizar la presentación de contenido y mejorar la experiencia del usuario.

Esta práctica ha representado una oportunidad invaluable para consolidar y aplicar los conocimientos adquiridos en el Módulo de Deep Learning, enfrentando desafíos realistas en el manejo de datos, diseño de modelos, entrenamiento y evaluación, y fomentando una mentalidad de experimentación y aprendizaje continuo. Sin duda ha sido una de las materias del bootcamp que más he disfrutado y que deseo seguir desarrollando, todo el contenido me pareció fascinante.