Proyecto: "Movie Genre Prediction"

David Hernando Avila Jimenez [a,c] , Erica Marcela Martínez Silva [a,c] , Lina Marcela Rivas[a,c],
Raúl Eduardo Vásquez Duarte [a,c], Sebastián Amaya Porras[a,c]

Sergio Alberto Mora Pardo[b,c]

[a]*Estudiantes de la Maestria en Analitica para Inteligencia de negocios*
[b]*Profesor, Departamento de Ingeniería Industrial*
[c]*Pontificia Universidad Javeriana, Bogotá, Colombia*

1. ENTENDIMIENTO DE NEGOCIO

La industria cinematográfica en las últimas décadas ha experimentado diversos cambios, generados principalmente por la digitalización y la diversificación de las plataformas de consumo. Actualmente, el consumo de películas no se limita únicamente a las salas de cine tradicionales, sino que su distribución y consumo se ha extendido a plataformas de *streaming,* televisión por cable y servicios de video bajo demanda (VOD). Este cambio, llevó a transformar los patrones de consumo, abriendo camino al acceso a una amplia gama de contenidos desde cualquier dispositivo y en cualquier momento.

La segmentación de películas por géneros es clave para la industria, dado que ayuda a identificar audiencias específicas y facilita la efectividad en la creación y despliegue de campañas de marketing. Los géneros tradicionales, como la comedia, el drama, el suspenso o la ciencia ficción siguen siendo fundamentales, pero también han surgido subgéneros híbridos que mezclan elementos de diferentes categorías para atraer a públicos más diversos. La competencia en el sector ha llevado a una creciente demanda de contenido original y personalizado, llevando a los estudios y plataformas a invertir en producciones variadas que aborden nichos de mercado específicos.

El auge de plataformas de streaming como Netflix, Amazon Prime y Disney+ ha impulsado un modelo de negocio basado en la personalización, en el que los algoritmos juegan un papel central al recomendar contenido que se ajuste a los gustos y hábitos de los usuarios. Estos nuevos medios de consumo cambiaron la manera como gestionan los catálogos de contenido, donde la categorización precisa y la rápida identificación del género de una película o serie juegan un papel esencial en la experiencia de los usuarios. Así mismo, estas nuevas formas de consumo impactaron a los estudios cinematográficos y las distribuidoras tradicionales, los cuales deben adaptarse a las nuevas formas de consumo y distribución digital para seguir siendo competitivos.

Por otro lado, el marketing cinematográfico se ha vuelto cada vez más visual y digital, con el póster de una película desempeñando un papel fundamental en la comunicación de la identidad y el género de una producción. Los *pósters* se han adaptado a los nuevos formatos digitales y a las plataformas de redes sociales, donde son utilizados para captar la atención del espectador en un entorno saturado de imágenes. Estos elementos visuales no solo deben atraer a los consumidores, sino que también deben transmitir rápidamente la esencia y el género de la película, influyendo en la decisión de ver el contenido.

En este contexto, la automatización en el análisis de pósteres a través de técnicas de inteligencia artificial representa una oportunidad importante para la industria del cine. Facilita la creación de sistemas más

eficientes para la clasificación y recomendación de contenido, permitiendo que las plataformas y estudios optimicen sus estrategias de marketing y mejoren la experiencia del usuario en un mercado cada vez más competitivo y saturado. Para el entendimiento a nivel interno y externo del sector se llevó el estudio de 2 metodologías diferentes PESTEL y DOFA.

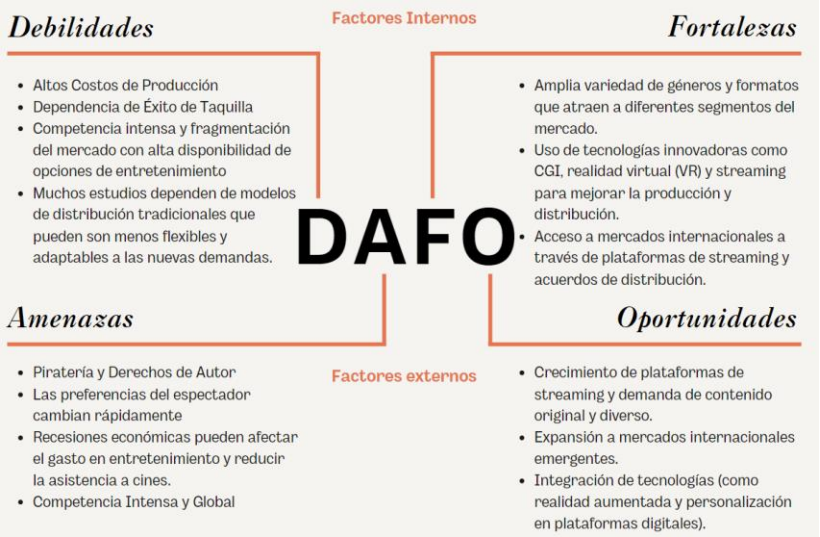


Ilustración 1. Matriz DOFA. Elaboración propia

Ilustración 2. Análisis PESTEL. Elaboración propia

1.1 Contexto de negocio.

1.1.1. Objetivo de negocio:

- Ofrecer a las plataformas de streaming de películas, la oportunidad de fortalecer sus mecanismos de categorización de contenido, a través de una herramienta que precise correctamente las clasificaciones y mejore así la experiencia al cliente.

- Brindar una herramienta que sea un puente entre el marketing y publicidad digital cinematográfico y las plataformas de streaming de películas, al permitir la categorización de estas a partir de una pieza digital.

1.1.2. Criterio de éxito:

- Realizar la categorización correcta de al menos el **70%** de un catálogo que contiene 3.383 películas, brindado como muestra para el presente estudio.

1.2. Determinación de objetivos de minería de datos

1.2.1. Objetivo minería de datos

- Desarrollar un modelo basado en el procesamiento de lenguaje natural, que prediga el género de las películas a partir del contenido que hay en sus posters.

1.2.2. Criterio de éxito

- Se espera obtener un AUC mayor o igual a **0.89**, a través del uso de un modelo que utilice técnicas de vectorización y representaciones distribuidas (*embbedings*), para la predicción del género de las películas.

2. ENTENDIMIENTO DE LOS DATOS

El presente proyecto se desarrolla a partir de 2 *datasets* suministrados por el profesor Fabio González, Ph.D. y su estudiante John Arévalo, quienes a partir de estos datos llevaron a cabo la construcción del artículo titulado "*Gated Multimodal Units for Information Fusion*". Las bases suministradas se distribuyen de la siguiente manera:

Set de entrenamiento:

Dataset en formato .csv, el cual contiene una lista de 7.895 películas diferentes, cada cuenta con características como un código ID único, el año de su estreno, título, resumen, los géneros asociados a esta y el rating.

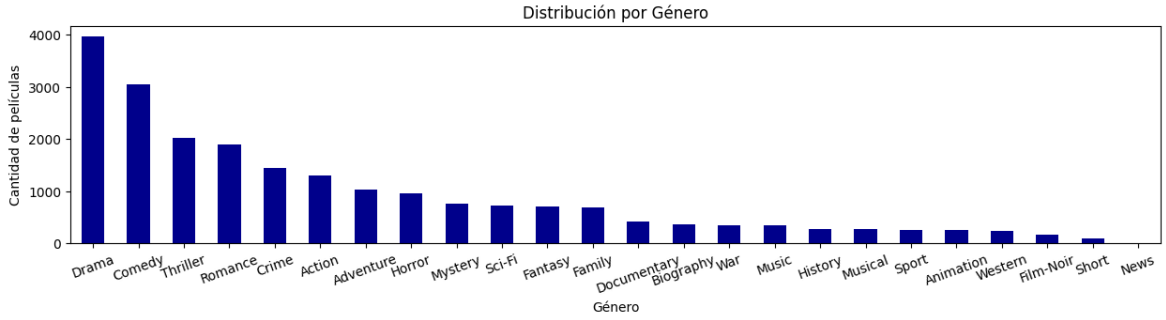


Ilustración 3. Gráfico de barras para demostrar la cantidad de películas por género, en el Dataset.

La distribución por género nos deja ver se cuenta con 24 categorías, lideradas por Drama con 3.965 títulos asociados, seguida por Comedia con 3.046 títulos. Es importante resaltar, que cada película puede estar clasificada en más de un género. La distribución de la variable año de lanzamiento deja ver que el *dataset* contiene películas lanzadas entre los años 1894 y 2015, presentando una tendencia creciente a partir del año 1980, lo cual guarda relación con el avance tecnológico y la disposición de recursos para la producción cinematográfica.

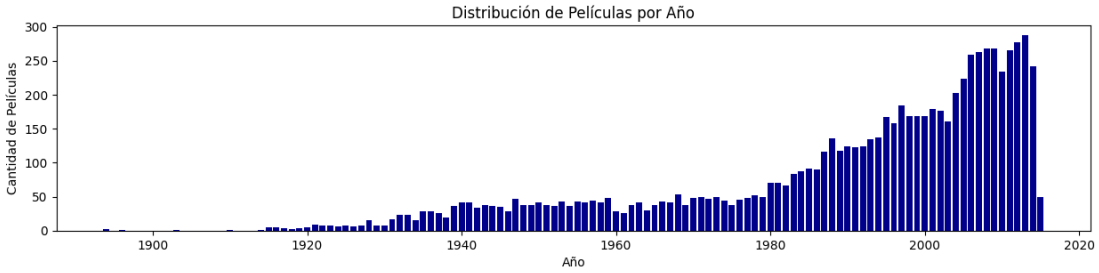


Ilustración 4. Distribución de las películas a lo largo del tiempo, según su año de lanzamiento.

La variable plot contiene la descripción o resumen de la trama de la película, componente fundamental dado que proporciona una idea general de la historia, su género y eventos que se desarrollaran en esta.

Set de testing:

De igual manera, el formato del set de testing es tipo .csv, este contiene una lista de **3.383** películas, cada una de ellas con información referente a ID único, el año de su estreno, título, y su correspondiente resumen. Para este caso, la distribución de años de estreno se encuentra entre 1983 y 2015, sin embargo, presenta el mismo comportamiento creciente evidenciado en la base de train.

3. PREPARACIÓN DE LOS DATOS

3.1 Limpieza de datos

Para la limpieza de datos se realizaron los siguientes pasos:

a. Se lleva a cabo la limpieza de símbolos mediante la función *re.sub(r"[.,;:!-]"*.
b. Se convierten el texto de la columna *plot* en minúsculas mediante la función *doc.lower()* y se eliminan espacios adicionales usando *split()* y posteriormente *join()* para retornar el texto sin espacios.
c. Se aplica un proceso de *lemmatización*, algoritmo creado para reducir las palabras a su raíz/base y de esta manera facilitarle al modelo una información más normalizada, mejorando su aprendizaje.
d. Se descargan las stopwords en inglés usando el módulo de stopwords de NLTK, las cuales serán eliminadas del texto.
e. Para algunos modelos, se decidió hacer Stemming como preprocesamiento.
f. Se aplica un proceso de *Tokenización* sencilla, es decir, por espacios como delimitador.

4. MODELING & EVALUATION

A continuación, se muestran 9 de los modelos realizados, con sus respectivos AUC.

MODELO	PARÁMETROS	AUC
TF-IDF + Word2Vec & Random Forest	**Preprocesamiento:** Lematizado **TfidfVectorizer(max_features=3000)** **Modelo embeding Word2Vec:** **vector_size**=100, **window**=10, **min_count**=1**, workers**=4 **Clasificador:** (n_estimators=100, max_depth=10)	0.502
Word2Vec & Random Forest	**Preprocesamiento:** Lematizado **Modelo embeding Word2Vec:** **vector_size**=100, **window**=5, **min_count**=1**, workers**=4 **Clasificador:** (n_estimators=100, max_depth=10)	0.588
Word2Vec	**Preprocesamiento:** Lematizado **Modelo embeding Word2Vec:** **vector_size**=1000, **window**=3, **min_count**=1**, workers**=4	0.619
TF-IDF & Random Forest	**Preprocesamiento:** Lematizado **TfidfVectorizer(max_features=3000)** **Clasificador:** (n_estimators=100, max_depth=10)	0.809
N-grams (1- 4) & Random Forest	**Preprocesamiento:** Lematizado **ngram_range= (1, 4)** **Clasificador:** (n_estimators=200, max_depth=20, 'min_samples_split': 5, 'min_samples_leaf': 1)	0.843
N-Grams (1-3) con ajuste de min_df & Random Forest	**Preprocesamiento:** Lematizado **ngram_range= (1, 4), min_df = 2** **Clasificador:** (n_estimators=200, max_depth=20, 'min_samples_split': 5, 'min_samples_leaf': 1)	0.866
BOW & Red Neuronal Sencilla	**Preprocesamiento:** Lematizado **Vectorizador:** Bag of Words **Red Neuronal:** Secuencial, Flatten, Dense('relu'),428 neuronas, Dropout(0.29), Dense activación: sigmoid, #salidas = #etiquetas), **Optimizador:** Adam**, loss:** binary_crossentropy.	0.872
N-Grams (1-3) con ajuste de max feautres & Random Forest	**Preprocesamiento:** Lematizado **ngram_range =** (1, 3), **max_features =** 900,000 **Clasificador:** (n_estimators=100, max_depth=10)	0.877
Regresión Logística / Random Forest	Unión 'plot' + 'Title' como data de entrenamiento. Usar la variable de 'Year' en el vector de x **ngram_range =** (1, 3) **Preprocesamiento:** Lematizado **Regresión logística:** C=1.0, Solver= 'lbfgs', max_iter=2000 **Random Forest:** ('n_estimators': 500, 'min_samples_split': 2, 'min_samples_leaf': 4, 'max_depth': 30, 'bootstrap': False)	0.879
BOW + GLOVE & Red Neuronal Sencilla	Unión 'plot' + 'Title' como data de entrenamiento. **Preprocesamiento:** Lematizado **Vectorización:** Bag of Words **Red Neuronal:** Secuencial, Flatten, Dense('relu'), Dropout(0.29), Dense(activación: sigmoid, #salidas = #etiquetas), **Optimizador:** Adam**, loss:** binary_crossentropy.	0.9134

4.1.1 Asses Model

De acuerdo con la tabla presentada, se detallaron la construcción de los modelos, destacando su comportamiento y capacidad de clasificación de los géneros para cada película, evaluados mediante el AUC (métrica clave utilizada para medir el éxito de los modelos desarrollados). A continuación, se describirán los 4 modelos con mejores resultados:

4.1.2 BOW & Red Neuronal Sencilla

En este modelo, se utilizó un preprocesamiento de lematización con el fin de mantener las palabras en su base o raíz. Posterior a ello, para la vectorización de las palabras se utiliza la técnica de Bag of Words, a través de la cual se genera una matriz en la que se hace conteo de términos, de acuerdo con el número de veces que aparece la palabra en el documento, en este caso la columna de 'plot'.

Una vez generado el modelo de Bag of Words, se desarrolló un modelo de predicción de red neuronal sencilla, usando Keras, para realizar la clasificación *multilabel* que se necesita en el presente proyecto, generando que la capa de salida de la red fuese igual a la cantidad de labels por predecir. De esta manera se construyó un modelo iniciando con una entrada de Flatten, con el fin de estructurar los datos de entrada para que quedasen totalmente conectados; para la capa oculta, se le asignaron **428** neuronas y una función de activación **ReLU**.

Para la capa de salida, al ser un problema de multi etiqueta, se estableció que el número de neuronas en esta capa era igual, al número de etiquetas de salida, es decir ***y.shape = 24*** y una función de activación sigmoide dado que, de igual manera se requería clasificar si pertenece o no a la etiqueta (género de película) correspondiente.

Para este modelo se utilizó el optimizador Adam, con una tasa de aprendizaje de **0,00004**; debido a la complejidad de la red y la cantidad de datos que se estaban manejando, puesto que el rango común para este optimizador es entre 0.001 – 0.01.

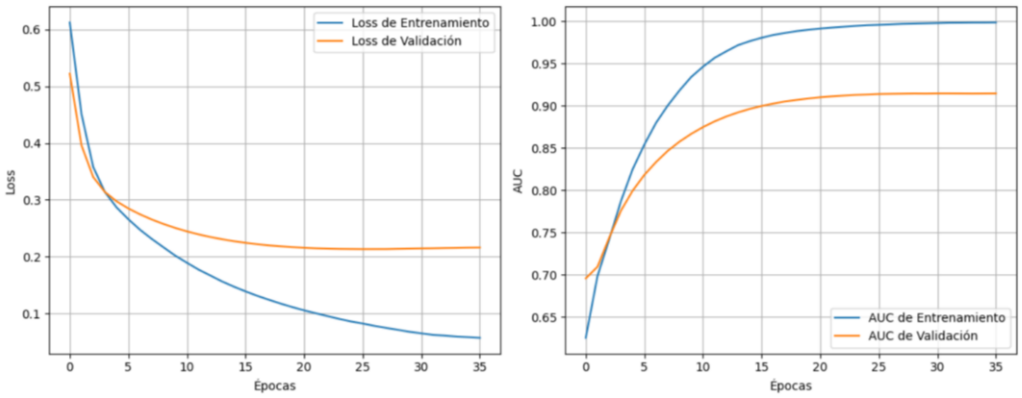


Ilustración 5. Gráfica de Loss y AUC generadas en el modelo BOW & Red Neuronal Sencilla

El modelo arrojó un **AUC de 0.872**, 0.04 puntos por debajo del mejor modelo obtenido. Este AUC, al revisar su comportamiento (*Ilustración 5*), se puede notar un poco de sobre ajuste entre entrenamiento y validación,

probablemente por la cantidad de neuronas asignadas en la capa oculta. Sin embargo, se acerca al AUC objetivo.

4.1.3 N-Grams (1-3) con ajuste de max feautres & Random Forest

En este modelo, se mantuvo como preprocesamiento la lematización y posteriormente, se decidió ajustar la vectorización *N-grams* entre *1 y 3*; con el fin de generar la matriz de palabras que estuviesen desde una sola palabra hasta una combinación de 3, lo cual ayuda al modelo a capturar mayor contexto y relación entre las palabras de un texto. Así mismo, se le agregó el ajuste de *max features = 900K*, con el objetivo que el modelo analice los Ngramas generados y sólo tome esta cantidad de características más frecuentes en el texto analizado, reduciendo el ruido que pueda haberse creado previamente.

Luego del preprocesamiento desarrollado previamente, se definió el modelo de *Random Forest*, con sus hiperparámetros optimizados a través de un método de *RandomizedSearchCV* para luego seleccionar la mejor combinación de hiperparámetros y aplicarlos a los conjuntos de entrenamiento y validación creados. Los mejores hiperparámetros, con los cuales se desenvolvió el modelo Random Forest, fueron los siguientes:

```
print("Mejores hiperparámetros:", random_search.best_params_)
```
✓ 0.0s

```
Mejores hiperparámetros: {'n_estimators': 200, 'min_samples_split': 10, 'min_samples_leaf': 2, 'max_depth': None, 'bootstrap': False}
```

Finalmente, al obtener la métrica de AUC se obtuvo un **0.874,** acercándose por 0.02 puntos al objetivo planteado de 0.89.

4.1.4 Regresión Logística / Random Forest

En los siguientes modelos, se usaron una regresión logística, así como la optimización de un modelo de Random Forest mediante la búsqueda de hiperparámetros. Ambos modelos fueron diseñados con el fin de predecir géneros de películas, utilizando como datos las descripciones, títulos y el año de lanzamiento de estas. La métrica de evaluación empleada fue el AUC macro, dado que proporciona una medida equilibrada del rendimiento en problemas de clasificación multiclase.

En cuanto al preprocesamiento, se inició con la limpieza y preparación del texto. Las descripciones de las películas se combinaron con sus respectivos títulos y, posteriormente, fueron lematizadas y tokenizadas. Utilizando el vectorizador CountVectorizer, se transformaron los textos en una matriz de n-gramas con un rango de (1, 3). Además, la característica 'year' (año de lanzamiento) fue normalizada empleando un escalador MinMaxScaler, que transformó sus valores entre 0 y 1. A continuación, las matrices de texto y la característica del año se combinaron para formar el conjunto de características final que se utilizó tanto en el entrenamiento como en la evaluación del modelo.

Para el modelo de clasificación, se utilizó una regresión logística multiclase mediante el clasificador OneVsRestClassifier, configurado con los parámetros de regularización (C=1.0), el solver lbfgs y un límite de iteraciones de 2000, utilizando los datos combinados de texto y año.

Al evaluar el rendimiento, se calculó el AUC macro tanto en los datos de entrenamiento como en los de prueba. En el conjunto de entrenamiento, el modelo alcanzó un AUC macro de 1.00, lo que indica un ajuste perfecto a estos datos. Sin embargo, en el conjunto de prueba, el AUC macro obtenido fue de 0.87, lo cual,

aunque muestra una buena capacidad de generalización, evidencia una ligera disminución del rendimiento en comparación con el conjunto de entrenamiento, sugiriendo la posibilidad de un leve sobreajuste.

En el escenario de Random Forest mediante se usó una búsqueda aleatoria de hiperparámetros, utilizando RandomizedSearchCV. Entre los parámetros evaluados se incluyeron el número de árboles (n_estimators), la profundidad máxima (max_depth), el número mínimo de muestras para dividir un nodo (min_samples_split) y el número mínimo de muestras por hoja (min_samples_leaf). La búsqueda de hiperparámetros se realizó utilizando validación cruzada de tres pliegues, y el mejor modelo resultante fue entrenado con 500 árboles, una profundidad máxima de 30, y un mínimo de cuatro muestras por hoja. Tras el ajuste, el modelo de Random Forest optimizado alcanzó un AUC macro de 0.879 en el conjunto de prueba, superando ligeramente al modelo de regresión logística en términos de capacidad predictiva. Se propone desarrollar un tipo de modelos diferentes, más robustos, pero sin incurrir en complejidad y un vectorizador con una capacidad de generar embeddings para lograr un AUC más alto y cumplir los objetivos.

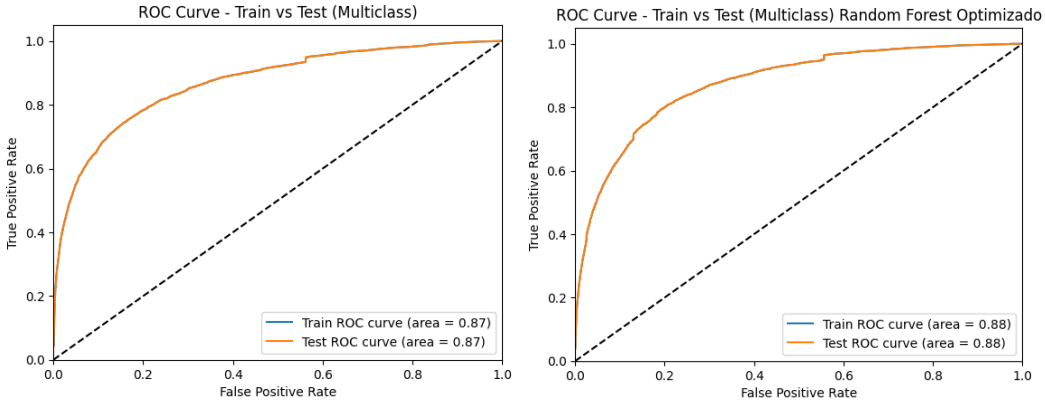


Ilustración 6. Gráfica curva ROC generadas en el modelo Regresión Logística / Random Forest

4.1.5 BOW + GloVe & Red Neuronal Sencilla

Finalmente, el modelo que obtuvo el mejor AUC entre las pruebas realizadas para este proyecto fue uno en el que se unificó el tipo de vectorización Bag Of Words junto con el tipo de *embedding* denominado GloVe y el modelo de predicción Red neuronal Sencilla. Dicho modelo se construyó de la siguiente manera:

Inicialmente se realizó un ajuste en los datos de entrenamiento, para los cuales se unieron las columnas de la descripción de la película y el título de esta, con el fin de brindarle más información al modelo para su entrenamiento. Estos datos tuvieron también un preprocesamiento de lematización y limpieza de stopwords y de símbolos.

Posteriormente, se utilizó el modelo pre entrenado de embedings GloVe (Global Vectors for Word Representation) diseñado para capturar información estadística global sobre patrones de coocurrencia de palabras en un corpus, para aprender incrustaciones que capturen eficazmente las relaciones semánticas entre las palabras (IBM, s.f.). Se requería unir la representación obtenida a partir de BOW junto con el modelo GloVe, en una misma matriz densa concatenándolas donde cada fila tenía la representación BoW y su correspondiente embedding de GloVe.

Luego de la unificación en una misma matriz, se dividieron los datos en entrenamiento y validación respectivamente; para ingresarlos a una red neuronal sencilla. Esta red estaba estructurada de la misma manera que la descrita en el modelo **4.1.1 BOW & Red Neuronal Sencilla,** cambiando únicamente la cantidad de neuronas en la capa oculta a **400**. Finalmente, este modelo dio como resultado un **AUC de 0.91**, siendo este el más alto entre los modelos desarrollados en el presente proyecto; con un comportamiento de Loss y AUC como se muestra a continuación:

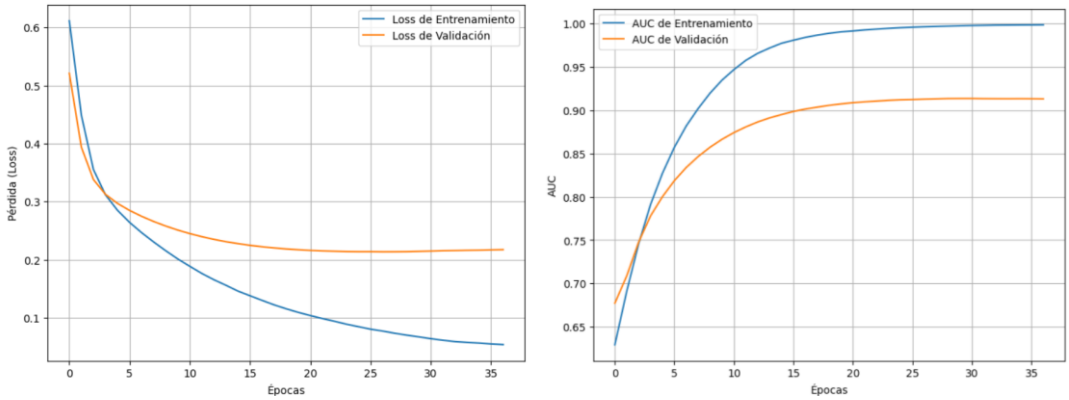


Ilustración 7. Gráficas de Loss y AUC para el modelo con el mayor AUC.

5. DEPLOYMENT

Enlazados los objetivos planteados con los modelos construidos y los resultados obtenidos, donde se evidencia un AUC de 0.913, se plasma un despliegue a partir de 3 visuales: 1. Un escenario de prueba, 2. Estrategias enfocada en el despliegue y 3. Pasos siguientes; con el fin de disponer a los medios cinematográficos y de streaming una herramienta solida que fortalezca el negocio.

Pruebas efectuadas

Una vez seleccionado el mejor modelo a nivel de métricas de rendimiento, se llevaron a cabo pruebas para validar el género sobre unas películas específicas. Para ello, se seleccionaron 2 películas del *dataset* de entrenamiento y 2 películas externas. De las 4 películas, para el caso de BABAR el modelo predijo erróneamente el género Comedia; para el caso de Toy Story el modelo no predijo el género de Animación.

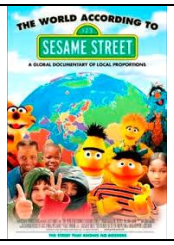
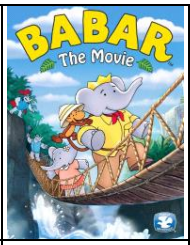
Pelicula				
Género Correcto	Documentary	Adventure, Fantasy, Family, Animation, Musical	Animation, Adventure, Comedy, Family, Fantasy	Action, Adventure
Género Predicción	Documentary	Adventure, Comedy	Adventure, Comedy, Family, Fantasy	Action, Adventure

Ilustración 8. Resultado Predicción Género Película

A partir de los resultados, el modelo demostró un desempeño sólido, prediciendo correctamente el género de las películas en la mayoría de los casos, lo cual refuerza su capacidad de adaptación y precisión (75%) para nuevos títulos.

Pasos Siguientes

- **Optimización de modelos:** Refinar el modelo para mejorar su eficiencia computacional, asegurando que pueda manejar catálogos de películas más grandes sin presentar perdidas en el rendimiento; incluyendo mejoras en los tiempos de ejecución y en la optimización de los recursos.
- **Escalabilidad:** Es importante desarrollar una infraestructura escalable que permita llevar a cabo el despliegue del modelo en cada una de las plataformas de streaming, al tiempo que soporte la integración de nuevo catálogo, incrementos en el volumen de películas y llevar a cabo una mejora continua.
- **Actualización del catálogo:** Incluir en el modelo nuevas fuentes de información como las bases de datos proporcionadas por IMDB, para actualizar constantemente las clasificaciones a medida que se lanzan nuevas películas o se identifican tendencias emergentes.
- **Pruebas Piloto:** Previo a la implementación a escala, se debe realizar pruebas con un subconjunto del catálogo de películas para verificar la correcta clasificación de al menos el 70% de las películas.
- **Monitoreo y mantenimiento:** Una vez efectuado el despliegue del modelo, es necesario implementar un sistema de monitoreo del rendimiento del modelo, lo cual permitirá evidenciar variaciones en la precisión del modelo o posibles problemas de clasificación, permitiendo actuar inmediatamente en las correcciones pertinentes.

Estrategias

- Crear una API que permita a otras plataformas o desarrolladores integrar el sistema de clasificación de géneros en sus propios servicios, generando nuevas fuentes de ingresos.
- Establecer colaboraciones con proveedores de tecnologías para mejorar la capacidad de procesamiento, garantizar la sostenibilidad a largo plazo del modelo y proporcionar acceso a infraestructuras avanzadas.
- Expandir el modelo para clasificar contenidos en otros formatos, como series de televisión, documentales o incluso contenido generado por usuarios en plataformas como YouTube, abriendo nuevas oportunidades de negocio.

Link del Modelo en Git: