

Trabajo Final de LLM:

# Análisis de sentimientos de reseñas de películas

Alumno: Ing. Darío Navarro

Profesor: Ing. Ezequiel Espósito



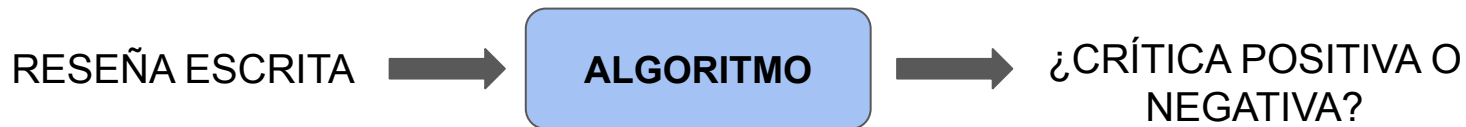
# Índice

1.	Introducción.....	3
2.	Desarrollo.....	4
3.	Resultado.....	6
4.	Conclusiones.....	8

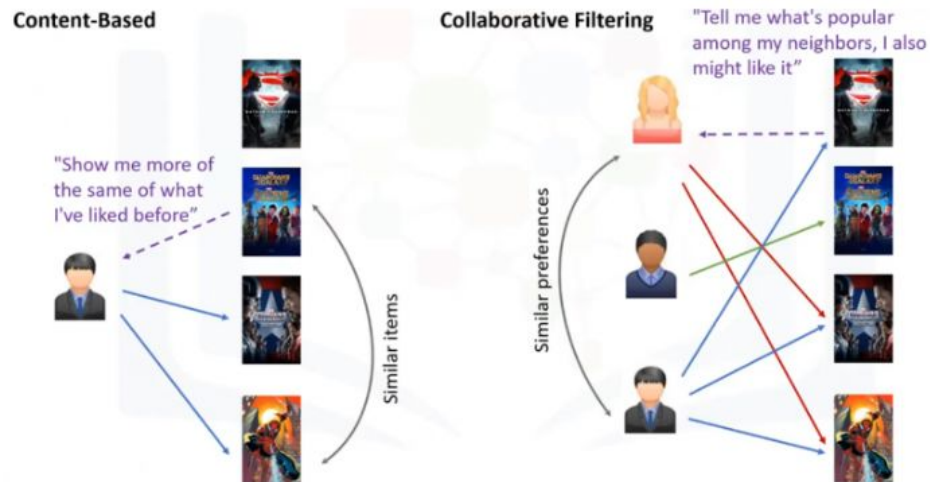


# 1. Introducción

**Desafío:** ¿cómo traducir una reseña escrita de una película en un sentimiento como “positivo” o “negativo”?



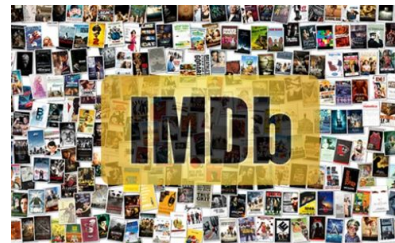
**Utilidad:** el sistema podría usarse como base para alimentar otro algoritmo de recomendación de tipo “basado en contenido” o de “filtrado colaborativo”.



# 2. Desarrollo

## Dataset

- ❑ Se usará un dataset con reseñas en inglés de películas de la página web IMDB.

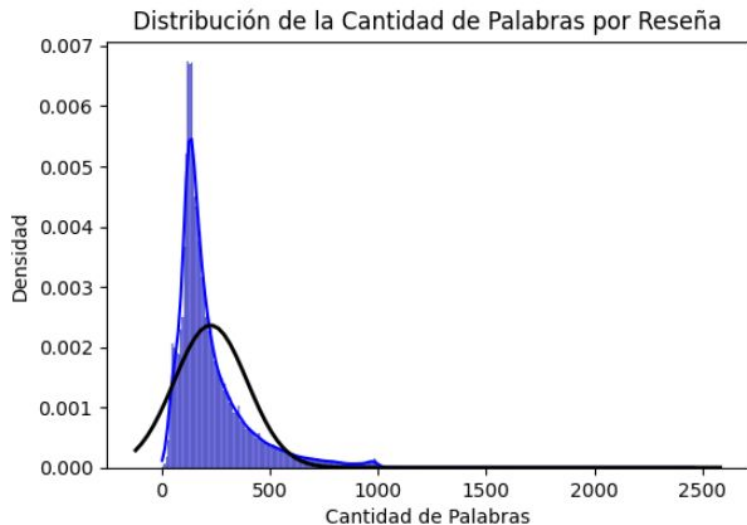


## Arquitectura

- ❑ Se usará BERT junto con su tokenizer específico. Se agregarán algunas capas adicionales a BERT para poder realizar el transfer learning y ajuste fino.

## Análisis exploratorio de datos

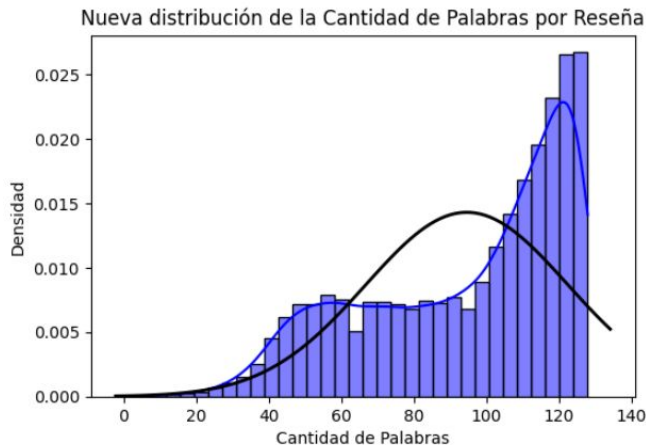
- ❑ **Longitud del dataset:** 50.000 filas
- ❑ **Clases:** 2
- ❑ **Distribución del dataset:** 50/50
- ❑ **Máxima longitud de reseña:** 2459 caracteres



## 2. Desarrollo

### Preprocesamiento

- ❑ Limpieza de texto con regex.
- ❑ Extracción de un subset reducido con reviews que no excedan una longitud máxima de secuencia de 128 palabras (para agilizar el cómputo en Colab).
- ❑ Extracción de un segundo subset que además de la condición anterior tenga 2500 reseñas de cada clase (para agilizar el cómputo en Colab).
- ❑ Codificación: se mapea el sentimiento “positivo” a 1 y el “negativo” a 0.
- ❑ Tokenización usando el tokenizador de BERT (lower case).
- ❑ División del dataset en train/test 80/20.
- ❑ Creación del data loader para cargar los batch de entrenamiento.



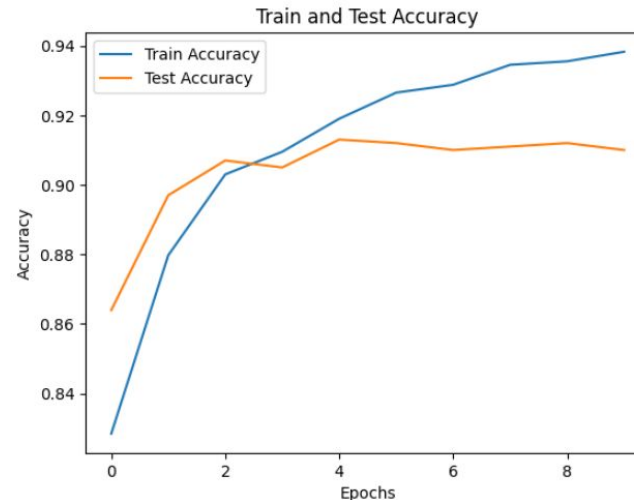
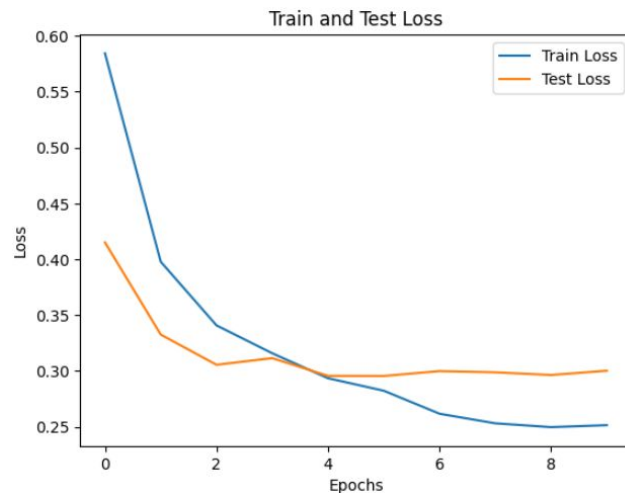
### Arquitectura del modelo

**BERT** + 1 capa de 8 neuronas (con activación Tanh) + 1 capa de 2 neuronas

### Entrenamiento del modelo

- ❑ **Epochs:** 10
- ❑ **Learning rate:** 0.000002
- ❑ **Optimizer:** Adam
- ❑ **Entorno de cálculo:** Google Colab usando T4(GPU)
- ❑ **Métrica seleccionada:** Accuracy (se elige esta métrica por tratarse de un dataset balanceado).

# 3. Resultados









## Classification Report:

	precision	recall	f1-score	support
0	0.91	0.90	0.91	479
1	0.91	0.92	0.91	521
accuracy			0.91	1000
macro avg	0.91	0.91	0.91	1000
weighted avg	0.91	0.91	0.91	1000

### 3. Resultados

A continuación se muestran algunos resultados con reseñas que no estaban en el dataset original:

Review		Sentimiento	Correcto?
<i>"Great movie. I would say it could be a great Oscar winner for sure this year"</i>	→		
<i>"A total disaster. Don't ever watch it because it's a waste of time"</i>	→		
<i>"The movie extols moral values from another era. It's the type of film that normalizes the moral decay of the world."</i>	→		

## 4. Conclusiones

- El algoritmo desarrollado en este trabajo alcanza una precisión del orden del 90 % por lo que el modelo se considera satisfactorio.
- Se evidencia que la predicción es buena cuando la reseña es categórica y usa palabras clave como “good”, “bad”, “recommend”, “great” o “disaster”. Es decir, el algoritmo es sensible al uso de adjetivos calificativos.
- La precisión ya no es muy buena cuando la reseña es ambigua y la crítica a la película debe interpretarse en clave de juicios morales o estéticos subyacentes (ver ejemplo 3 en la slide anterior).
- El punto anterior nos da un indicio de que para mejorar la precisión del algoritmo en reseñas difíciles de interpretar sería necesario volver a hacer un ajuste fino del algoritmo entrenándolo con reseñas engañosas donde el sentimiento no está dado tanto por los adjetivos usados sino más bien por el metamensaje.
- Como idea para seguir profundizando en el tema, sería bueno probar los resultados al entrenar el algoritmo con el dataset entero de 50.000 reviews en vez de usar un subset de 5.000 reviews como el utilizado en este trabajo por las limitaciones en el poder de cómputo disponible.