

## **Laboratorio #5**

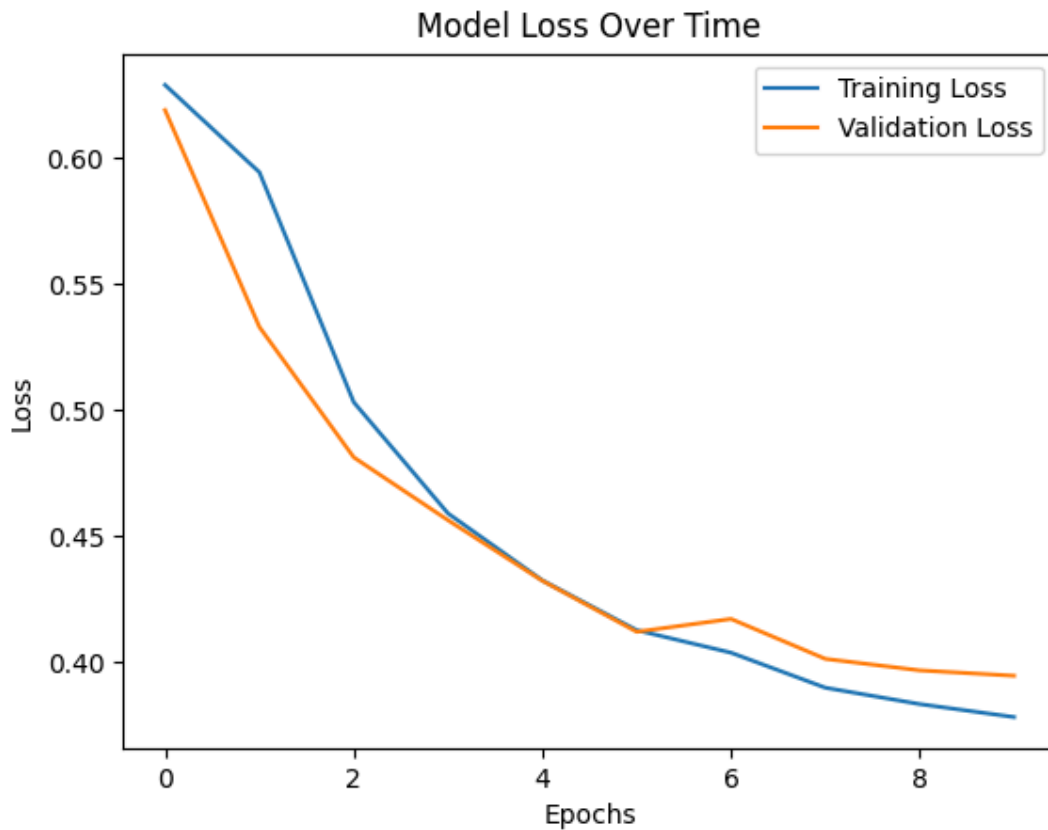
### LSTM

21016 Javier Chavez  
21085 Andres Quezada  
21631 Mario Cristalitos

## Resultados:

### *Modelo Base:*

Pérdida a través del entrenamiento:



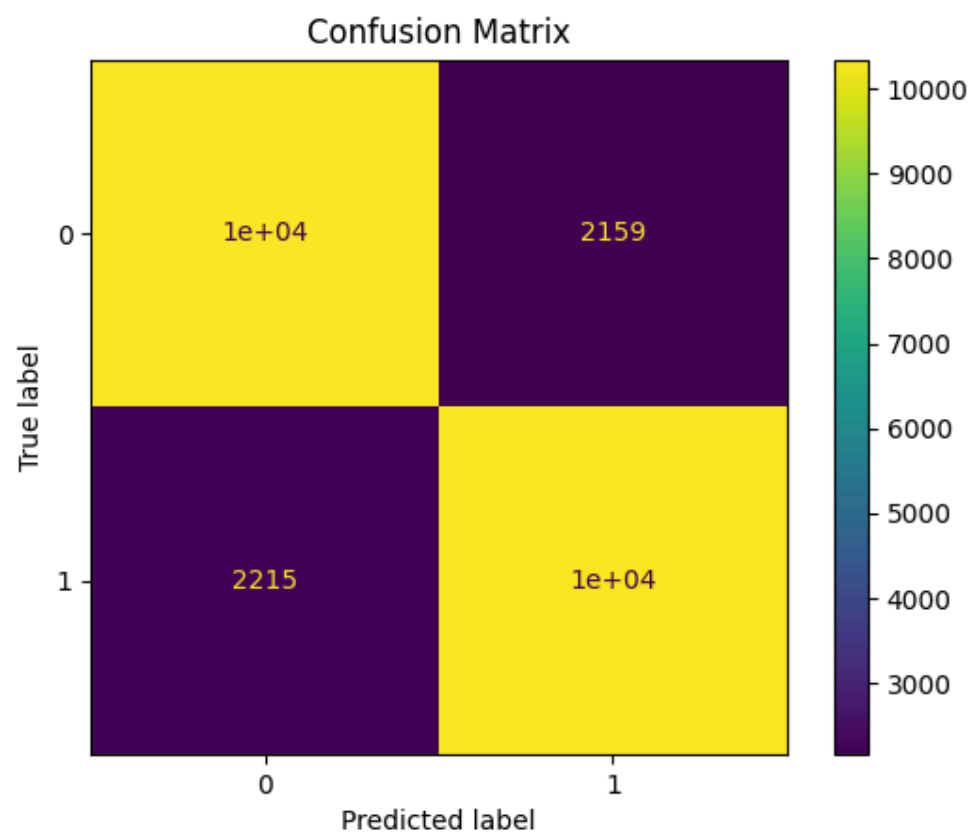
Pérdida y Accuracy final:

```
702/702 1293 165ms/step accuracy: 0.8262  
Loss: 0.39362633228302, Accuracy: 0.8261600136756897  
702/702 1293 165ms/step
```

Pérdida (Loss): El modelo alcanzó una pérdida de 0.3936 en el conjunto de prueba, lo que indica que el modelo ha mejorado/reducido la diferencia entre las predicciones y las etiquetas reales.

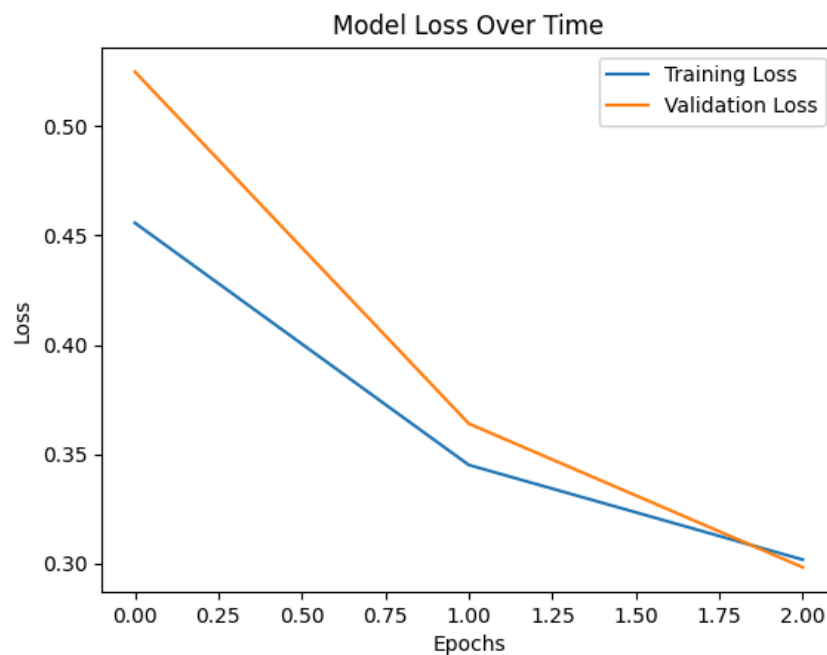
Exactitud (Accuracy): El modelo logró una exactitud de 82.62% en el conjunto de prueba.

Matriz de confusión:



### Modelo Mejorado:

Pérdida a través del entrenamiento:



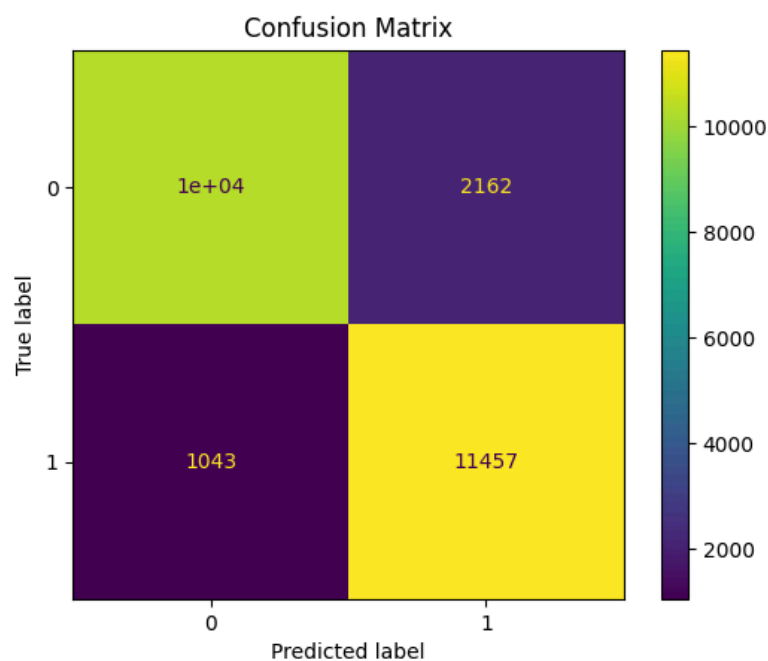
Pérdida y Accuracy final:

Loss: 0.3116371364593506, Accuracy: 0.8802000284194946

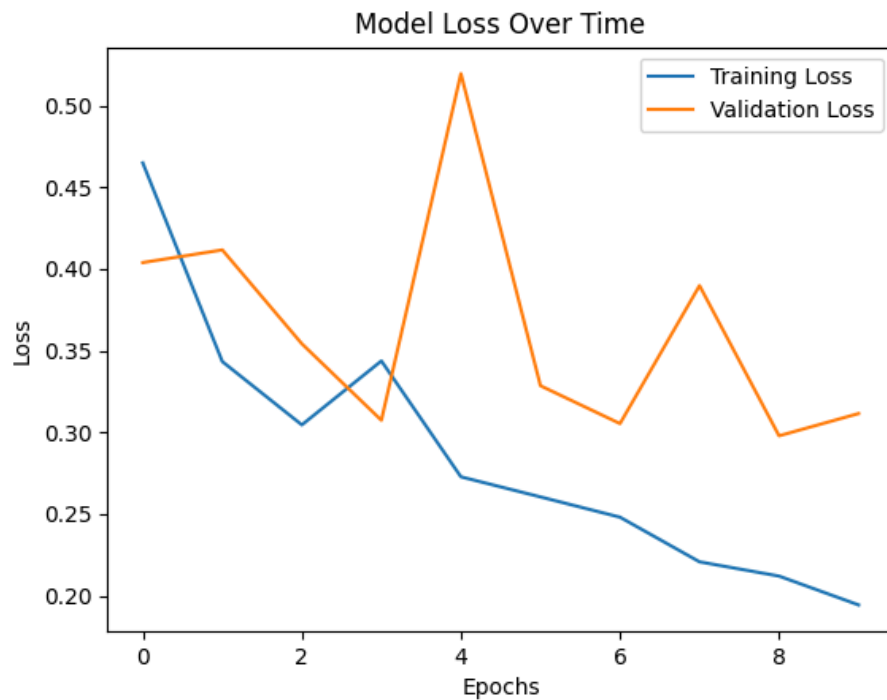
Pérdida (Loss): El modelo alcanzó una pérdida de 0.311.

Exactitud (Accuracy): El modelo logró una exactitud de 88.02%.

Matriz de confusión:



Como se puede observar, el modelo mejorado incrementa la accuracy del modelo por aproximadamente 6% y disminuye la pérdida de 0.394 a 0.311. Esto se ve reflejado en la matriz de confusión también, porque vemos mayor cantidad de aciertos tanto negativos como positivos. Y cabe mencionar que para el modelo mejorado se acortó el tiempo de entrenamiento a 3 épocas para evitar el sobre ajuste, ya que luego de experimentar con este modelo, se podía observar que luego de la tercera época empezaba a oscilar y sobre ajustarse sin mejora:



### Mejoras con respecto al modelo base

En comparación con el modelo base, el modelo modificado ha demostrado (una mejora en la exactitud, una reducción en la pérdida, una mejor capacidad para clasificar correctamente las reseñas, etc.).

Estas mejoras son el resultado de los siguientes cambios:

- Aumento de max\_features: Al incrementar el número de palabras consideradas en el dataset de 400 a 2000, el modelo pudo captar un mayor rango de palabras, mejorando su comprensión de las reseñas.
- Incremento de maxlen: Al extender el largo máximo de las reseñas de 300 a 500, se permitió que el modelo tomará en cuenta un contexto más amplio de las reseñas, lo que resultó en una mejor captura de las dependencias a largo plazo.
- Se añaden dos capas LSTM en lugar de una:
  - Primera LSTM: Con 64 unidades y return\_sequences=True, lo que permite que la salida de cada paso de tiempo sea utilizada por la siguiente capa LSTM.
  - Segunda LSTM: Con 32 unidades, que recibe la secuencia completa de la capa LSTM anterior, mejorando la capacidad del modelo para capturar dependencias a largo plazo en las secuencias.

### Justificación de los cambios

Los cambios realizados en el modelo fueron motivados por la necesidad de mejorar su capacidad de generalización y su rendimiento en la tarea de clasificación de sentimientos. A continuación, se explica el razonamiento detrás de cada cambio:

**Aumento de max\_features:** Incrementar el número de palabras consideradas permite al modelo trabajar con un vocabulario más rico y variado. Esto es especialmente útil en tareas de procesamiento de lenguaje natural, donde la comprensión de un mayor número de palabras puede mejorar significativamente la precisión del modelo.

**Incremento de maxlen:** Al permitir que el modelo procese secuencias más largas, se aumenta su capacidad para comprender el contexto completo de una reseña. Esto es crucial para capturar matices y relaciones a largo plazo entre las palabras, lo que puede ser clave en la clasificación de sentimientos.

**Adición de capas LSTM:** Las redes LSTM son potentes para capturar dependencias a largo plazo en datos secuenciales. Al agregar más capas LSTM, se espera que el modelo capte patrones más complejos en los datos, lo que debería traducirse en una mejor capacidad de predicción. Este enfoque, sin embargo, también puede aumentar el riesgo de sobreajuste, por lo que se debe monitorear cuidadosamente el rendimiento en el conjunto de validación.

Repositorio:

<https://github.com/JaniMariQuesiRami/DeLe-Lab5>

Referencias:

Brownlee, J. (2020) How to Tune LSTM Hyperparameters with Keras. Recuperado de:

<https://machinelearningmastery.com/tune-lstm-hyperparameters-keras-time-series-forecasting/>