

# Tarea 3. Análisis de Modelos de Clasificación para Predicción de Sentimiento en Reseñas de Películas

Ivan Gabriel Salinas Castillo

30 de enero de 2025

## 1. Introducción

El objetivo de este proyecto es comparar distintos modelos de clasificación para predecir el sentimiento (positivo o negativo) de reseñas de películas. Utilizando un conjunto de datos con reseñas de películas etiquetadas, se entrenaron y evaluaron varios modelos: Support Vector Machine (SVM) [1], Naive Bayes [2], XGBoost [3] y una Red Neuronal con Keras [4].

## 2. Metodología

Para este análisis, se realizó un preprocesamiento del texto que incluyó la limpieza de las reseñas, eliminación de stopwords, y tokenización [5, 6]. Posteriormente, se aplicaron técnicas de vectorización, como TF-IDF [2], para transformar las reseñas de texto en características numéricas utilizables para los modelos de clasificación. Además se utilizó el *IMDB Dataset of 50K Movie Reviews (Spanish)* [7].

Se entrenaron y evaluaron los siguientes modelos:

- **Support Vector Machine (SVM):** Un clasificador basado en márgenes máximos que busca separar las clases de manera óptima.
- **Naive Bayes:** Un modelo probabilístico basado en la independencia condicional de las características.
- **XGBoost:** Un algoritmo de boosting que combina múltiples modelos débiles para obtener un modelo fuerte. Entrenado de forma separada, utilizando las representaciones TF-IDF ya calculadas.

- **Red Neuronal con Keras:** Modelo de red neuronal con capas densas y activaciones ReLU, entrenado utilizando los vectores TF-IDF como entrada

## 2.1. Vectorización de Texto

Para representar el texto en un formato que pueda ser utilizado por los modelos de aprendizaje automático, se empleó el método de Term Frequency-Inverse Document Frequency (TF-IDF). Este método asigna un peso a cada palabra en función de su frecuencia en un documento y su relevancia en comparación con el resto del corpus. De esta forma, se minimiza el impacto de palabras comunes y se resaltan aquellas más representativas del contenido de la reseña. El TF-IDF se calcula con la siguiente fórmula:

$$TF - IDF(t, d) = TF(t, d) \times IDF(t)$$

Donde:

- **TF(t, d):** Frecuencia de un término en un documento.
- **IDF(t):** Logaritmo inverso de la frecuencia del término en todos los documentos.

Se empleó la implementación de TF-IDF de Scikit-learn para transformar las reseñas en matrices de características antes de entrenar los modelos de clasificación.

Para la evaluación, se utilizó el conjunto de prueba separado previamente y se calcularon las métricas de precisión, recall, F1-score y accuracy [8]. Además, se generaron matrices de confusión para visualizar el desempeño de cada modelo.

## 3. Visualización de Datos

Para comprender mejor la distribución del texto en el conjunto de datos, se generaron las siguientes gráficas:

- Un histograma que muestra la distribución del número de palabras en las reseñas positivas y negativas (Figura 1).
- Un gráfico de barras comparando el porcentaje de reseñas positivas y negativas en el dataset (Figura 2). Donde podemos observar que son la mitad para cada sentimiento hay 25,000 reviews, en total 50,000.

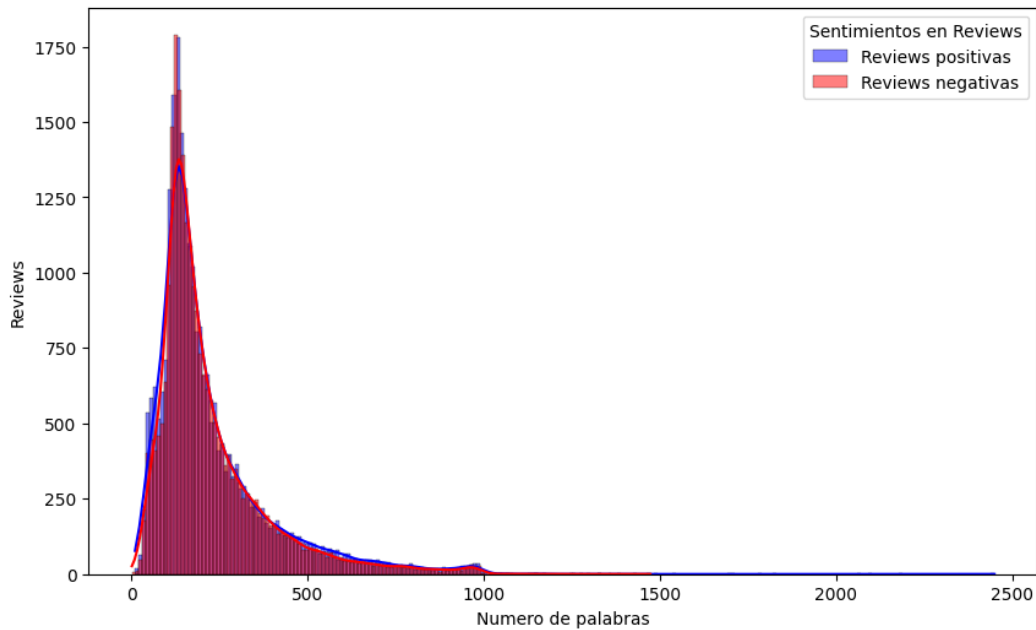


Figura 1: Distribución de numero de palabras.

## 4. Resultados

A continuación, se muestran los resultados de evaluación para cada uno de los modelos:

### 4.1. Support Vector Machine (SVM)

El modelo SVM logró una precisión del 86 % y un recall del 85 % para la clase 0, y una precisión del 85 % y un recall del 88 % para la clase 1. La matriz de confusión (Figura 3) muestra que el modelo tiene un desempeño equilibrado en ambas clases.

Evaluación para SVM:

	precision	recall	f1-score	support
0	0.87	0.85	0.86	4961
1	0.85	0.88	0.87	5039
accuracy			0.86	10000
macro avg	0.86	0.86	0.86	10000
weighted avg	0.86	0.86	0.86	10000

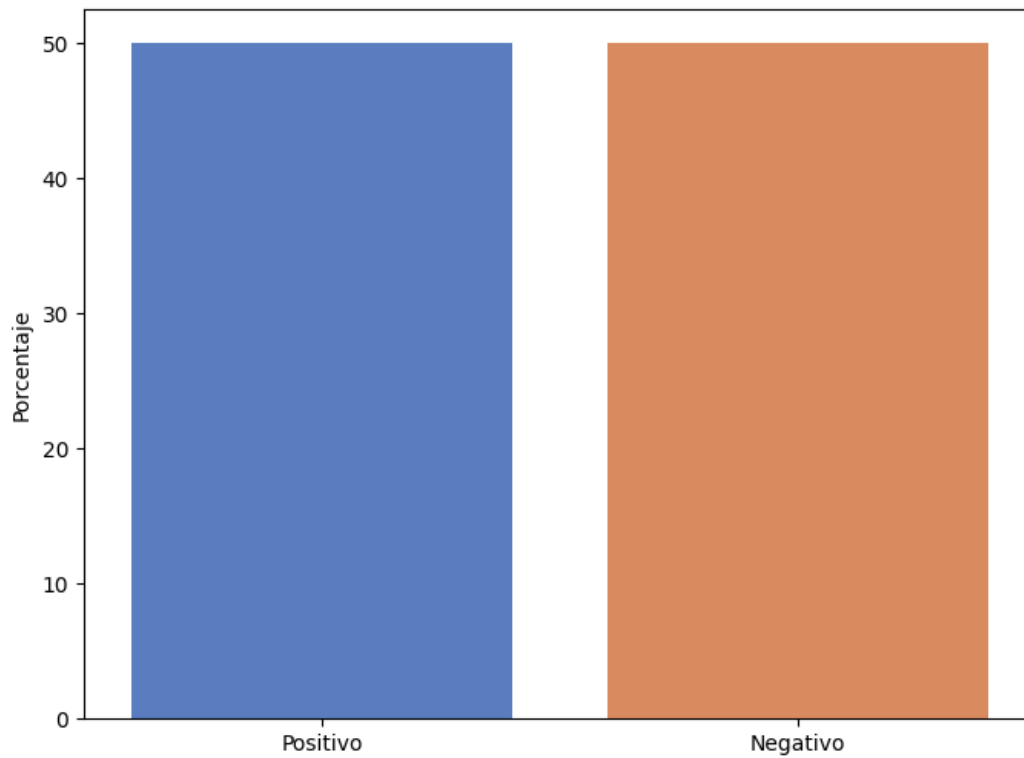


Figura 2: Proporción de reviews.

## 4.2. Naive Bayes

El modelo Naive Bayes alcanzó una precisión del 84 % y un recall del 82 % para la clase 0, y una precisión del 82 % y un recall del 85 % para la clase 1. La matriz de confusión (Figura 4) refleja un buen rendimiento general del modelo.

Evaluación para Naive Bayes:

	precision	recall	f1-score	support
0	0.84	0.82	0.83	4961
1	0.82	0.85	0.83	5039
accuracy			0.83	10000
macro avg	0.83	0.83	0.83	10000
weighted avg	0.83	0.83	0.83	10000

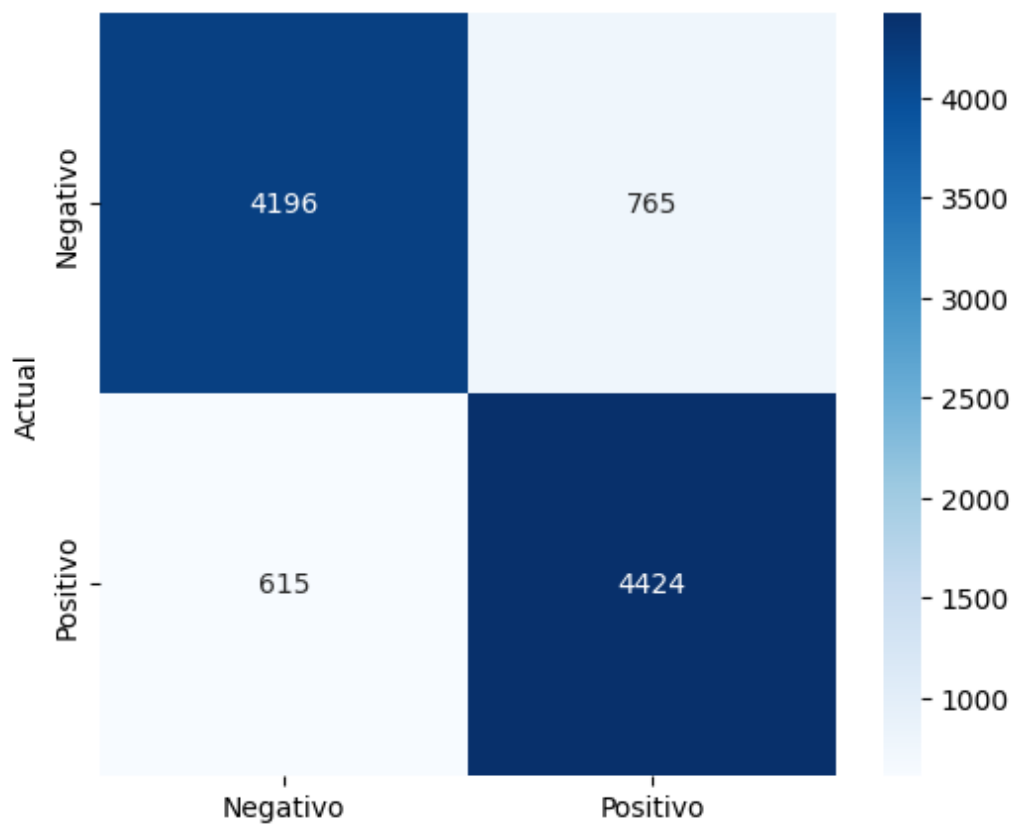


Figura 3: Matriz de confusión SVM.

### 4.3. XGBoost

El modelo XGBoost obtuvo una precisión del 86 % y un recall del 82 % para la clase 0, y una precisión del 83 % y un recall del 86 % para la clase 1. La matriz de confusión (Figura 5) muestra que el modelo tiene un rendimiento sólido, aunque ligeramente inferior al SVM y la red neuronal.

Evaluación para XGBoost:

	precision	recall	f1-score	support
0	0.86	0.82	0.84	4961
1	0.83	0.86	0.85	5039
accuracy			0.84	10000
macro avg	0.84	0.84	0.84	10000
weighted avg	0.84	0.84	0.84	10000

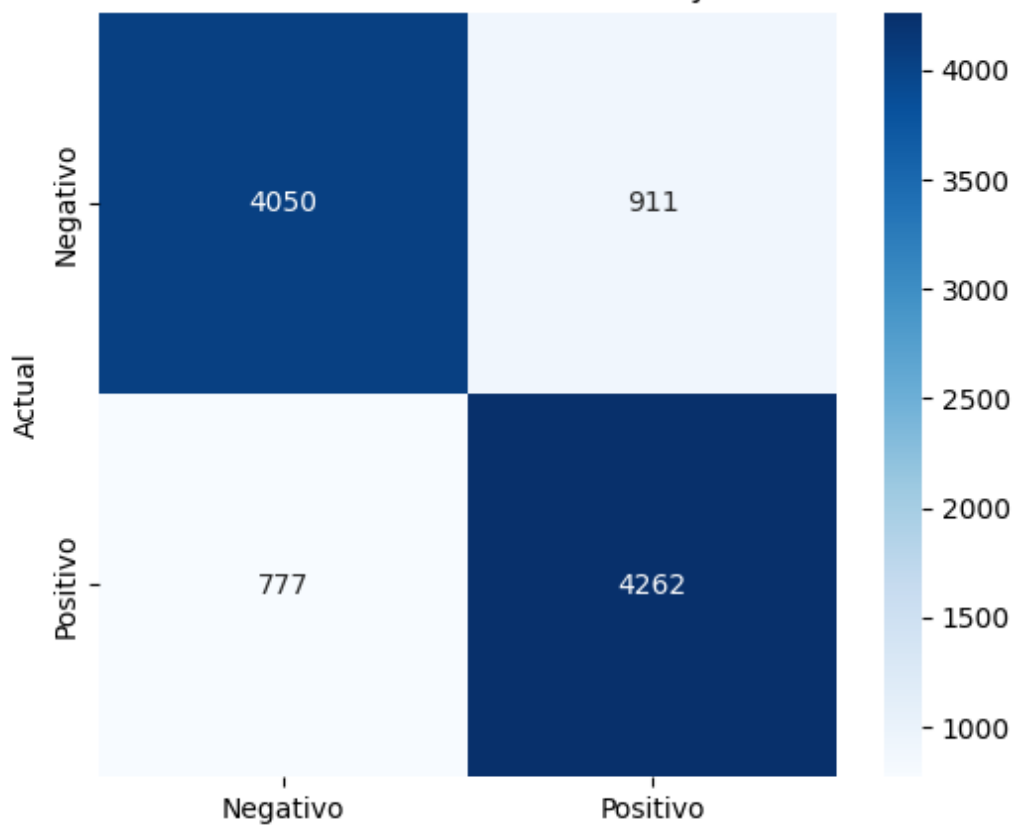


Figura 4: Matriz de confusión Naive Bayes.

#### 4.4. Red Neuronal con Keras

La red neuronal con Keras logró una precisión del 85 % y un recall del 86 % para la clase 0, y una precisión del 86 % y un recall del 85 % para la clase 1. La matriz de confusión (Figura 6) ilustra un rendimiento equilibrado y robusto del modelo.

Evaluación para Neural Network:

	precision	recall	f1-score	support
0	0.85	0.86	0.85	4961
1	0.86	0.85	0.85	5039
accuracy			0.85	10000
macro avg	0.85	0.85	0.85	10000
weighted avg	0.85	0.85	0.85	10000

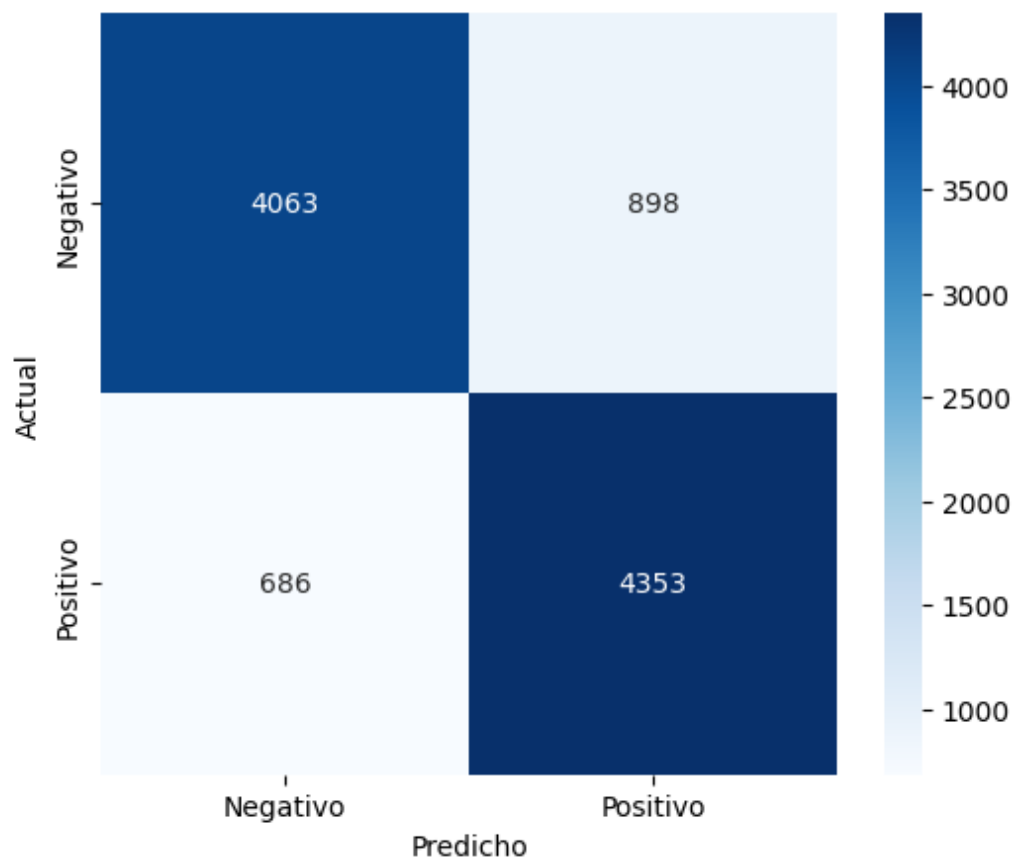


Figura 5: Matriz de confusión XGBoost.

## 5. Discusión

Los resultados muestran que el modelo SVM obtuvo el mejor desempeño con un accuracy de 0.86 y un f1-score promedio de 0.86, superando ligeramente a la red neuronal con Keras, que alcanzó un accuracy de 0.85. XGBoost y Naive Bayes obtuvieron un rendimiento inferior con accuracies de 0.84 y 0.83, respectivamente.

El modelo SVM logró una mayor precisión y recall en ambas clases, lo que sugiere que es una opción robusta para la clasificación de sentimiento en este conjunto de datos. La red neuronal, aunque tuvo un desempeño muy cercano, podría beneficiarse de una mayor optimización de hiperparámetros [9] y técnicas como embeddings de palabras en lugar de TF-IDF.

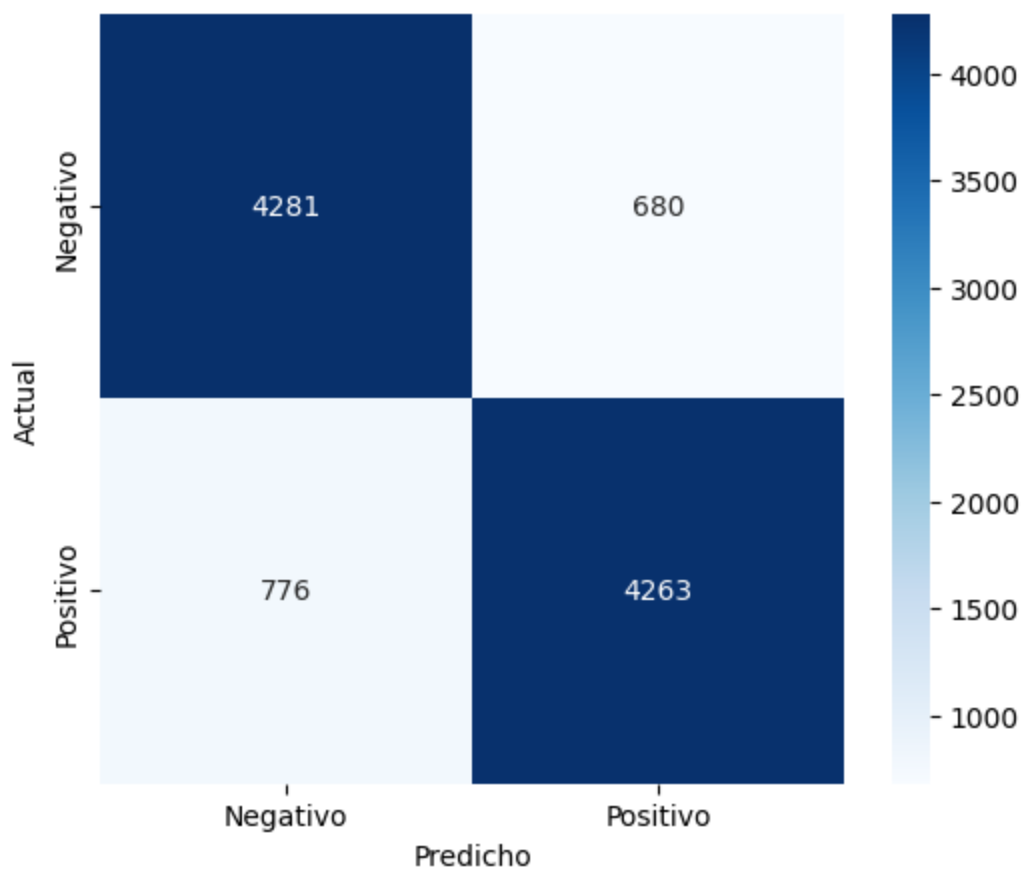


Figura 6: Matriz de confusión Red Neuronal.

## 6. Conclusiones

En este estudio se compararon cuatro modelos de clasificación para la tarea de análisis de sentimiento en reseñas de películas. Se encontró que el modelo SVM es la mejor opción en términos de precisión y recall, seguido muy de cerca por la red neuronal con Keras. Mientras que Naive Bayes y XGBoost también demostraron ser competitivos, no lograron superar el rendimiento de SVM.

## Referencias

- [1] Thorsten Joachims. “Text categorization with Support Vector Machines: Learning with many relevant features”. En: *Machine Learning* (1998), págs. 137-142.



- [2] Christopher D Manning, Prabhakar Raghavan e Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [3] Tianqi Chen y Carlos Guestrin. “XGBoost: A scalable tree boosting system”. En: (2016), págs. 785-794.
- [4] Yann LeCun, Yoshua Bengio y Geoffrey Hinton. “Deep learning”. En: *Nature* 521.7553 (2015), págs. 436-444.
- [5] Steven Bird, Ewan Klein y Edward Loper. *Natural Language Processing with Python*. 2009.
- [6] Explosion AI. *spaCy: Industrial-strength Natural Language Processing (NLP)*. <https://spacy.io>. 2023.
- [7] Luis Diego FV. *IMDB Dataset of 50K Movie Reviews (Spanish)*. Accedido el 29 de enero de 2025. 2021. URL: <https://www.kaggle.com/datasets/luisdiegofv97/imdb-dataset-of-50k-movie-reviews-spanish>.
- [8] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [9] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.