Análisis de Sentimiento con Reseñas de IMDB

Dora Alicia Guevara Villalpando
Matrícula: 1551003

Universidad Autónoma de Nuevo León)
Facultad de Ciencias Físico Matemáticas
Maestría en Ciencia de Datos
Procesamiento y Clasificación de Datos

dora.guevaravll@uanl.edu.mx

I. Introducción

El análisis de sentimiento es una tarea fundamental en el campo de la minería de texto y el aprendizaje automático. Este tipo de análisis permite identificar las emociones o actitudes expresadas en textos, como positivas, negativas o neutrales. En el contexto de las reseñas de usuarios, es especialmente útil para comprender la percepción del público hacia productos, servicios o contenidos.

En este proyecto, se empleó el conjunto de datos *IMDB Movie Reviews*, que contiene 50,000 reseñas de películas etiquetadas como positivas o negativas. Este dataset es ampliamente utilizado en la comunidad de aprendizaje automático debido a su balance entre clases y la complejidad del lenguaje natural empleado en las reseñas.

El objetivo principal fue desarrollar un modelo capaz de predecir el sentimiento de las reseñas con un alto grado de precisión. Para lograr esto, se exploraron técnicas clásicas de representación de texto y clasificación, evaluando tres enfoques: *Logistic Regression*, *Support Vector Machines (SVM)* y *Random Forest*. Este trabajo también buscó comparar el rendimiento de los modelos para recomendar el más adecuado según las características del dataset y las necesidades del análisis.

II. Metodología

II-A. Selección del Dataset

El dataset utilizado, *ÏMDB Movie Reviews"*, fue descargado desde el repositorio de Stanford. Este conjunto de datos está dividido equitativamente en dos subconjuntos: 25,000 reseñas para entrenamiento y 25,000 reseñas para prueba. Cada reseña tiene una etiqueta binaria que indica si el sentimiento es positivo o negativo. La estructura del dataset permitió realizar una división adicional del conjunto de entrenamiento en datos de validación para ajustar los modelos.

II-B. Preprocesamiento

El preprocesamiento de los datos incluyó:

1. Limpieza del texto: Se eliminaron etiquetas HTML, caracteres no alfanuméricos y espacios extra.
2. Normalización: Las reseñas se convirtieron a minúsculas para garantizar consistencia.
3. Eliminación de stop words: Durante la vectorización, se excluyeron palabras comunes que no aportan información significativa al análisis (como "the", .ªnd", etc.).

II-C. Vectorización

Para transformar las reseñas en un formato numérico adecuado para los modelos de clasificación, se utilizó el método **TF-IDF (Term Frequency-Inverse Document Frequency)**. Esta técnica mide la importancia de una palabra dentro de un documento en relación con el corpus completo. Se seleccionó un máximo de 5000 características, lo que permitió capturar las palabras más relevantes sin sobrecargar el modelo con dimensionalidad excesiva.

II-D. Modelos Evaluados

- **Logistic Regression:** Este modelo lineal es eficiente y altamente interpretable. Es adecuado para tareas donde las características tienen una relación lineal con las etiquetas de salida.
- **Support Vector Machines (SVM):** Utilizando un kernel lineal, este modelo busca maximizar los márgenes entre clases en espacios de alta dimensión, como los generados por TF-IDF.
- **Random Forest:** Este enfoque basado en conjuntos utiliza múltiples árboles de decisión para capturar patrones no lineales y manejar datos complejos.

Cada modelo fue entrenado utilizando el conjunto de datos de entrenamiento y evaluado en el conjunto de validación. Las métricas clave, como precisión, recall, F1-score y exactitud, fueron utilizadas para comparar su desempeño.

III. Resultados

Logistic Regression

- **Precisión**: 0.89 (clase negativa), 0.86 (clase positiva).
- **Recall**: 0.86 (clase negativa), 0.89 (clase positiva).
- **F1-Score**: 0.87-0.88.
- **Exactitud Global**: 0.87.

Support Vector Machines (SVM)

- **Precisión**: 0.88 (clase negativa), 0.86 (clase positiva).
- **Recall**: 0.86 (clase negativa), 0.89 (clase positiva).
- **F1-Score**: 0.87.
- **Exactitud Global**: 0.87.

Random Forest

- **Precisión**: 0.83 (clase negativa), 0.85 (clase positiva).
- **Recall**: 0.85 (clase negativa), 0.83 (clase positiva).
- **F1-Score**: 0.84.
- **Exactitud Global**: 0.84.

IV. Análisis comparativo

Acorde a lo que se observa en la tabla :

Modelo	Exactitud	F1 Score Promedio
Logistic Regression	0.87	0.87
SVM	0.87	0.87
Random Forest	0.84	0.84

Tabla I
Análisis comparativo

1. **Logistic Regression** y **SVM** mostraron desempeños casi idénticos, con una exactitud del 87 % y F1-Score promedio de 0.87.
2. **Random Forest** tuvo un desempeño inferior con una exactitud del 84 %, lo que sugiere que no es tan efectivo en este tipo de representación vectorial (TF-IDF).

Para el primer libro se obtuvo la siguiente nube de palabras:

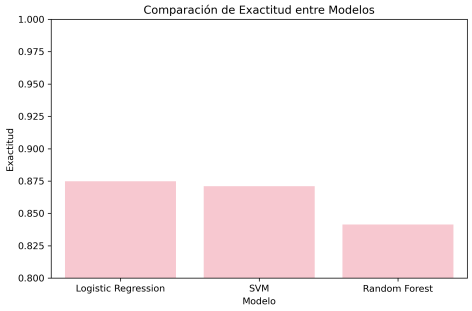

Figura 1. Comparación.

V. Conclusiones

1. **Modelo Recomendado**:
   - **Logistic Regression** es la mejor opción si se prioriza simplicidad y velocidad.
   - **SVM** es una alternativa igualmente válida si se busca robustez en otras configuraciones.
2. **Modelo Descartado**:
   - **Random Forest**, aunque flexible, no fue competitivo en este escenario.
3. **Futuras Mejoras**:
   - Probar modelos basados en *embeddings* preentrenados como BERT para mejorar el desempeño.
   - Incrementar el conjunto de datos para explorar el impacto en modelos más complejos.

Referencias

- Dataset: IMDB Movie Reviews (https://ai.stanford.edu/~amaas/data/sentiment/)
- Métodos: Scikit-learn Library (TF-IDF, Logistic Regression, SVM, Random Forest)
- Análisis de texto (text mining) con Python. (https://cienciadedatos.net/documentos/py25-text-mining-python)