**Procesamiento de lenguaje natural para críticas de películas en ImDB aplicando modelos de regresión logística y SVM**

**Huamani Loredo, Alvaro**

20151966@aloe.ulima.edu.pe

Universidad de Lima

# INTRODUCCIÓN

El presente documento analizará la aplicación de dos modelos, regresión logística y SVM, junto con el procesamiento de lenguaje natural, buscando predecir si una crítica en ImDB realizada a una película es positiva o negativa, para esto se usará un dataset que contiene 25000 críticas con sus respectivas valoraciones (“ImDb Movie Reviews Dataset | Kaggle,” n.d.). Debido a la cantidad de vectores que se tendrán despues de procesar las críticas, en lugar de validar el mejor modelo usando cross validation se usará el AUC de la curva ROC.

# BASES TEÓRICAS

## Máquinas de vectores de soporte (SVM)

Las máquinas de vectores de soporte (SVM) son clasificadores lineales particulares que se basan en el principio de maximización de márgenes. Realizan minimización del riesgo estructural, lo que mejora la complejidad del clasificador con el objetivo de lograr un excelente desempeño de generalización. El SVM realiza la tarea de clasificación mediante la construcción del hiperplano, en un espacio dimensional muy alto, que separa de manera óptima los datos en dos categorías. (Adankon & Cheriet, 2007)

## Procesamiento de lenguajes naturales (NLP)

El procesamiento del lenguaje natural es el análisis de datos lingüísticos, más comúnmente en forma de datos textuales como documentos o publicaciones, utilizando métodos computacionales. El objetivo del procesamiento del lenguaje natural es generalmente construir una representación del texto que agregue estructura al lenguaje natural no estructurado, aprovechando los conocimientos de la lingüística. Esta estructura puede ser de naturaleza sintáctica, capturando las relaciones gramaticales entre los componentes del texto, o más semántica, capturando el significado que transmite el texto. (Verspoor & Cohen, 2013)

## Técnicas de normalización para NLP

Para llevar a cabo el procesamiento de lenguaje natural, necesitamos realizar una normalización, para el análisis en este informe se usará:

**Eliminar signos de puntuación**

A veces, mientras se tokeniza, es conveniente eliminar la puntuación. La eliminación de la puntuación se considera una de las tareas principales al realizar la normalización en NLTK.

**Conversión a minúsculas o mayúsculas**

Un texto dado se puede convertir en texto en minúsculas o mayúsculas usando las funciones lower() y upper(). La tarea de convertir texto en mayúsculas o minúsculas se incluye en la categoría de normalización.

**Palabras vacías (stopwords)**

Las palabras vacías son palabras que deben filtrarse durante la tarea de recuperación de información u otras tareas del lenguaje natural, ya que estas palabras no contribuyen mucho al significado general de la oración. Hay muchos motores de búsqueda que funcionan eliminando palabras vacías para reducir el espacio de búsqueda. La eliminación de palabras vacías se considera una de las tareas de normalización que es crucial en la PNL. (Chopra, Mathur, & Joshi, n.d.)

## Convertir texto a matrices de tokens

En este análisis se usará la función CountVectorizer de la librería scikit-learn (“sklearn.feature\_extraction.text.CountVectorizer — scikit-learn 0.23.2 documentation,” n.d.) para convertir los textos en matrices de recuentos de tokens. Un problema con los recuentos simples es que algunas palabras como "el" aparecerán muchas veces y sus recuentos grandes no serán muy significativos en los vectores codificados. Una alternativa es calcular la frecuencia de las palabras y, con mucho, el método más popular se llama TF-IDF. Este es un acrónimo que significa “Frecuencia de términos - Frecuencia de documento inverso”, que son los componentes de las puntuaciones resultantes asignadas a cada palabra; para aplicar este método se usará la función TfidfTransformer de la librería scikit-learn (“sklearn.feature\_extraction.text.TfidfTransformer — scikit-learn 0.23.2 documentation,” n.d.).

## Curva ROC y el AUC

El análisis de la curva de característica operativa del receptor, abreviado como, ROC emplea la relación entre la sensibilidad y la especificidad de un clasificador binario. La sensibilidad o tasa de verdaderos positivos mide la proporción de positivos clasificados correctamente; La especificidad o tasa de verdaderos negativos mide la proporción de negativos clasificados correctamente. Convencionalmente, la tasa de verdaderos positivos (tpr) se traza frente a la tasa de falsos positivos (fpr), que es uno menos la tasa de verdaderos negativos.(Buhmann et al., 2011)

El área bajo una curva de característica operativa del receptor (ROC), abreviada como AUC, es un valor escalar único que mide el desempeño general de un clasificador binario (Hanley y McNeil 1982). El valor de AUC está dentro del rango [0.5–1.0], donde el valor mínimo representa el rendimiento de un clasificador aleatorio y el valor máximo correspondería a un clasificador perfecto, por ejemplo, con una tasa de error de clasificación equivalente a cero. (Melo, 2013)

# METODOLOGÍA Y EXPERIMENTACIÓN

Para la experimentación se hizo uso de la librería scikit learn de Python.

## Análisis de datos

El dataset de críticas a películas en la plataforma ImDb cuenta con 25000 registros, siendo la primera columna el texto de la crítica y la segunda la clasificación de esta, siendo 1 una valoración positiva y 0 una negativa. Aplicando un conteo general se puede observar que no tiene registros con valores nulos.

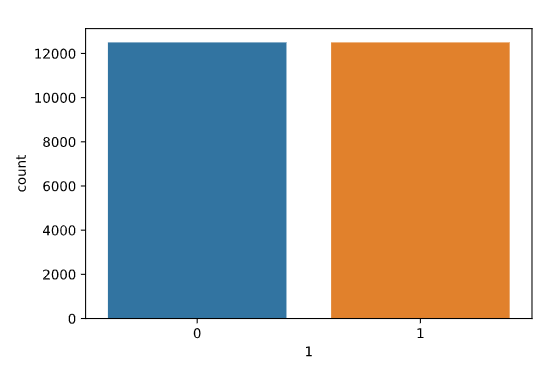


Fig. 1. Conteo de valoraciones negativas (0) y positivas (1)

Se puede observar en la figura 1 que la cantidad de críticas con valoración negativa es igual a las positivas.

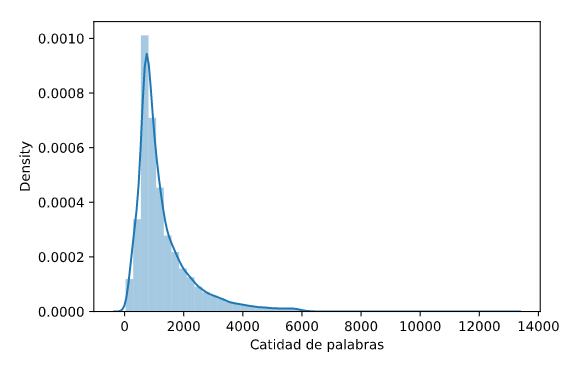


Fig. 2. Gráfico de distribución de la cantidad de palabras en cada crítica

Se puede observar en la figura 2 que la mayor cantidad de críticas cuentan con alrededor de 1000 palabras; también se observa que existen críticas con más de 12000 palabras; se puede concluir que se tiene un dataset extenso, con una considerable cantidad de palabras por crítica.

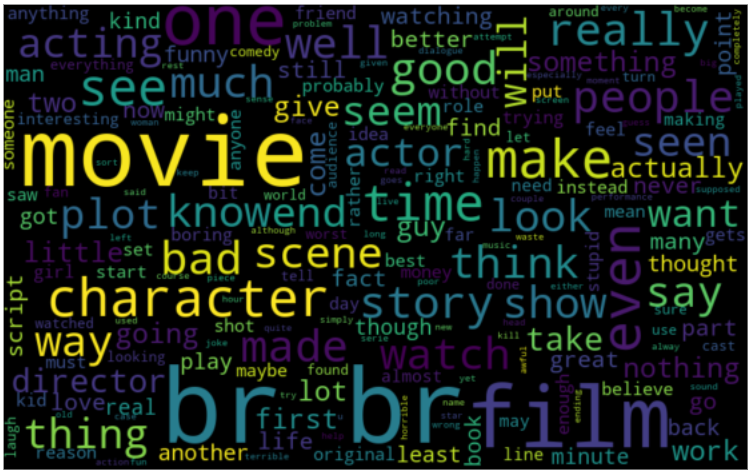


Fig. 3. Nube de palabras que más se repiten de las críticas negativas



Fig. 4. Nube de palabras que más se repiten de las críticas positivas

Observando la figura 3 y 4 se puede concluir que tanto para las críticas negativas y como para las positivas se tienen una gran variedad de palabras, donde en cada clasificación no se tiene una excesiva cantidad de palabras que se repiten en ambas; teniendo en cuenta que el dataset cuenta con una gran cantidad de palabras, se puede decir que el modelo tiene una buena cantidad de datos para trabajar.

## Aplicación de modelos con una configuración básica

Para poder convertir las palabras en vectores se hará uso de la función CountVectorizer, aplicando el modelo de regresión logística y SVM se observa que debido a que el dataset cuanta con una gran variedad de datos, el proceso de entrenamiento tiende a ser lento, por esta razón se decide usar como número máximo de iteraciones 1000 en ambos casos; también se usará el AUC del ROC para medir y comparar ambos modelos, esto en lugar de usar cross validation debido también a que en las primeras pruebas, el score obtenido del cross validation es similar al accuracy de una primera predicción del modelo.

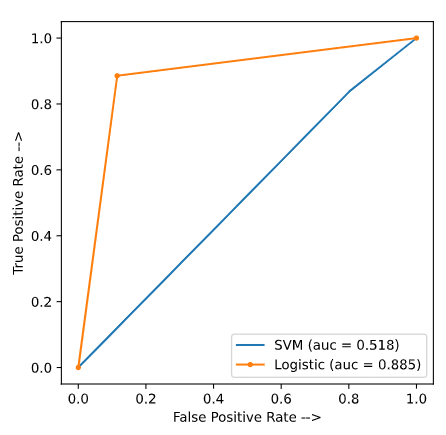


Fig. 5. Curva ROC de los modelos con una configuración inicial

Como resultado de los primeros modelos, se puede observar en la figura 5 que el modelo de regresión logística tiene un AUC que tiene a acercarse a 1, esto es un reflejo de que tiende a tener una buena cantidad de aciertos y pocos falsos positivos, el AUC del modelo SVM se acerca a 0.5, esto quiere decir que se tiene una similar cantidad de aciertos y falsos positivos, esto nos dice que el modelo no está aprendiendo correctamente.

Dados los resultados, se necesita de técnicas para mejorar la precisión del modelo, en especial para mejorar el modelo SVM.

## Usando técnicas de procesamiento de texto

Para lograr una mejor precisión, se convierten todas las palabras a minúsculas, se elimina los signos de puntuación de las críticas y también las palabras “vacías”, es decir aquellas palabras que no ayudan en la clasificación.

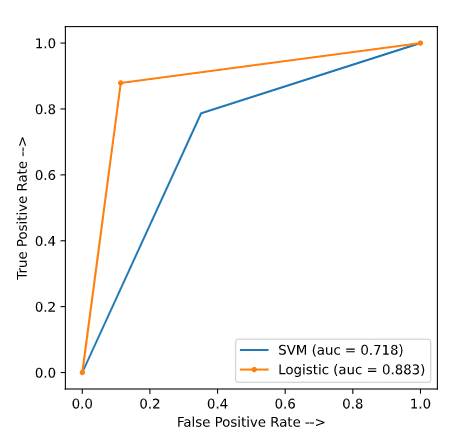


Fig. 6. Curva ROC de los modelos después del procesamiento de texto

Usando estas técnicas, aplicando los modelos con la misma configuración inicial se puede observar en la figura 6 que el modelo SVM mejoró en su puntuación, se tiene un incremento de 0.518 a 0.718 en el AUC. Esto es un buen indicio para seguir probando más técnicas para mejorar el modelo SVM.

## Transformando la matriz de recuento (CountVectorizer) en una representación tf o tf-idf normalizada (TfidfVectorizer)

Como alternativa a la función CountVectorizer se hace uso de la función TfidfVectorizer, con el fin de normalizar el conteo de vectores.

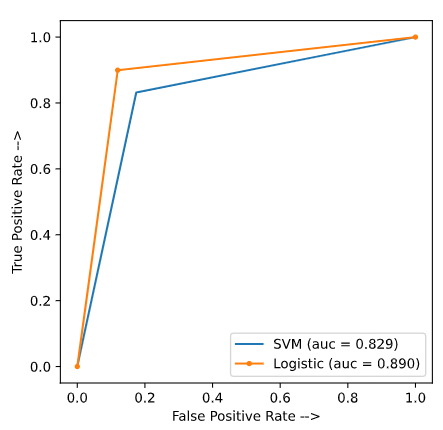


Fig. 7. Curva ROC de los modelos después normalizar el conteo de vectores

Luego de aplicar esta función para vectorizar las palabras procesadas y luego normalizarlas, se tiene como un pequeño incremento en el AUC del modelo de regresión logística (de 0.883 a 0.90) y un considerable incremento para el modelo SVM (de 0.718 a 0.829). Se puede concluir que el AUC de ambos modelos están por igualarse, a diferencia de la configuración inicial, donde la diferencia era mucho mayor.

## Buscando mejorar el modelo SVC implementando Grid Search

Buscando que la precisión de el modelo SVM iguale o supere al de regresión logística, se implementa el Grid Search, tienen como parámetros para el Kernel, linear y rbf, para la regularización (C), 1, 10, 100, para el coeficiente gamma, 1, 0.1, 0.01, 0.001. Debido a la larga duración de este proceso, no se llegaron a incluir más parámetros. Luego del entrenamiento del modelo con Grid Search, se obtuvo como mejores parámetros: kernel con rbf, regularización (C) con 1 y el parámetro gamma como 1.

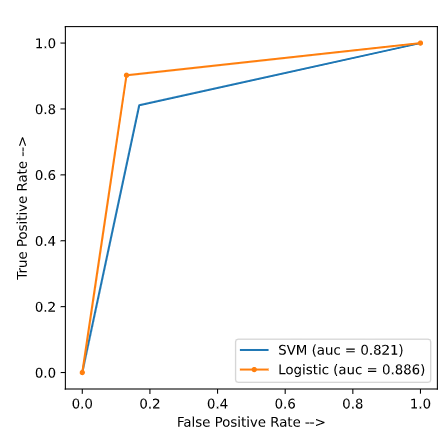


Fig. 8. Curva ROC de los modelos después de aplicar Grid Search

Como resultados de esta aplicación, observando la figura 8 se puede concluir que el AUC del modelo de SVM es similar al anterior, en parte porque los valores obtenidos del entrenamiento del grid search también son similares.

# CONCLUSIONES

Luego del análisis aplicando los modelos de regresión logística y SVM se llega a la conclusión de que el primero tiene una mejor precisión para clasificar estos datos, esto siguiendo el alcance de las configuraciones que se aplicaron a los modelos; la diferencia de puntuación entre ambos modelos no es muy grande, por lo que se puede intuir que aplicando más parámetros en el proceso de Grid Search, el modelo SVM puede llegar a superar al de regresión logística. También se puede concluir que las técnicas de procesamiento de texto usadas, tienen un gran impacto al momento de buscar mejorar la precisión del modelo, en especial para el modelo SVM.

**Link del repositorio:**

<https://github.com/alvaroenrique/ML_course-project2>

**REFERENCIAS**

Adankon, M. M., & Cheriet, M. (2007). Support Vector Machine. *Power Systems*, *28*, 161–226. https://doi.org/10.4018/978-1-60960-557-5.ch007

Buhmann, M. D., Melville, P., Sindhwani, V., Quadrianto, N., Buntine, W. L., Torgo, L., … Fürnkranz, J. (2011). ROC Analysis. In *Encyclopedia of Machine Learning* (pp. 869–875). https://doi.org/10.1007/978-0-387-30164-8\_733

Chopra, D., Mathur, I., & Joshi, N. (n.d.). *Mastering natural language processing with Python : maximize your NLP capabilities while creating amazing NLP projects in Python*.

ImDb Movie Reviews Dataset | Kaggle. (n.d.). Retrieved November 22, 2020, from https://www.kaggle.com/mantri7/imdb-movie-reviews-dataset

Melo, F. (2013). Area under the ROC Curve. In *Encyclopedia of Systems Biology* (pp. 38–39). https://doi.org/10.1007/978-1-4419-9863-7\_209

sklearn.feature\_extraction.text.CountVectorizer — scikit-learn 0.23.2 documentation. (n.d.). Retrieved November 22, 2020, from https://scikit-learn.org/stable/modules/generated/sklearn.feature\_extraction.text.CountVectorizer.html

sklearn.feature\_extraction.text.TfidfTransformer — scikit-learn 0.23.2 documentation. (n.d.). Retrieved November 22, 2020, from https://scikit-learn.org/stable/modules/generated/sklearn.feature\_extraction.text.TfidfTransformer.html

Verspoor, K., & Cohen, K. B. (2013). Natural Language Processing. In *Encyclopedia of Systems Biology* (pp. 1495–1498). https://doi.org/10.1007/978-1-4419-9863-7\_158