Instituto Tecnológico de Estudios Superiores de Monterrey

Evidencia 1. Artículo de investigación PBL1

Análisis de métodos de razonamiento e incertidumbre (Gpo 201)

Docente: Dr. Daniel Otero Fadul

Diego Armando Mijares Ledezma **- A01722421**

Pedro Soto Juárez - **A00837560**

Alexei Carrillo Acosta **- A01285424**

Marcos Renato Aquino Garcia - **A00835576**

Mauricio Octavio Valencia Gonzalez - **A01234397**

30 de septiembre de 2024

Problematización

El spam en correos electrónicos es un problema que afecta tanto a usuarios como a organizaciones. Hace referencia al envío masivo de correos no solicitados, usualmente con fines comerciales. No obstante, recientemente ha habido un incremento del spam con intenciones maliciosas como el phishing o la propagación de malware. El spam congestiona los buzones de entrada y provoca la pérdida de información importante. Así mismo, compromete la seguridad de los usuarios. En el 2022, el 45.37% del tráfico global de *emails* fue spam (Statista, 2023), lo que resalta la magnitud del problema. A pesar de la implementación de filtros en los sistemas de correo, las técnicas utilizadas por *spammers* evolucionan rápidamente, lo que dificulta detectar y prevenir el spam eficientemente (Gupta et al., 2022).

Enfoque

Para abordar el problema del spam, se implementó un modelo de clasificación basado en el algoritmo Naive Bayes, el cual ya es ampliamente utilizado en aplicaciones de filtrado de spam por su simpleza y eficiencia (Puthran et al., 2020). Este enfoque estadístico clasifica los correos electrónicos en "spam" y "no spam" utilizando el Teorema de Bayes. A través del procesamiento de lenguaje natural (NLP), se normaliza el texto para que sea interpretado por el algoritmo, eliminando palabras irrelevantes y aplicando técnicas como la tokenización y lematización (Jurafsky & Martin, 2021). Aunque este método es efectivo, la suposición de independencia condicional entre las palabras puede limitar la precisión del modelo. Esto se puede abordar mediante técnicas más avanzadas, como el uso de n-gramas o TF-IDF.

Propósito

El propósito es construir un clasificador de Naive Bayes para identificar correos electrónicos spam. Este se entrena utilizando conjunto de correos etiquetados, aprendiendo de probabilidades de que ciertas palabras sean spam o no. Cada palabra del correo electrónico es independiente de las demás, lo que simplifica cálculos y hace que el algoritmo sea eficiente, aplicando el teorema de Bayes (*Academia Carta Blanca*, 2023).

Para que el clasificador Naive Bayes funcione correctamente, es fundamental transformar los correos en datos para que el modelo los procese, utilizando así las técnicas ya mencionadas de tokenización (dividir texto en palabras) y stemming (reducir palabras a raíces). Se exploran conceptos como el aprendizaje probabilístico,la independencia de características, técnicas de procesamiento de lenguaje natural, y se busca comprender el rendimiento del modelo con diferentes métricas.

Información

El código utilizado usa el clasificador Naive Bayes ya mencionado, clasificando los correos electrónicos en sus respectivas categorías. Este enfoque es el más adecuado para problemas de clasificación de texto ya que la frecuencia de aparición de palabras en un documento ayuda a determinar su pertenencia en cierta categoría. La suposición de que todas las palabras en un correo son independientes entre sí, simplifica el cálculo de probabilidades.

El proceso general comienza con el preprocesamiento de correos, reduciendo las palabras a su raíz y pasandolos a una representación numérica con el enfoque de un *bag of words*. Después, el modelo usa las frecuencias de las palabras para estimar la probabilidad de que un correo sea spam o no. Con estas probabilidades se puede clasificar un nuevo correo usando la probabilidad conjunta de que sus palabras correspondan a una de las dos clases.

Una de las funciones importantes creadas fue el método de *bag of words,* que como se explicó anteriormente, pasa los correos en una lista que cuenta la frecuencia de aparición de cada palabra. Esto es de suma importancia para que el modelo logre aprender los datos, ya que a partir de estas frecuencias se obtienen las probabilidades que el modelo utiliza para hacer predicciones.

Otra función importante que se usó fue la de clasificar correos electrónicos. Esta función se encarga de predecir si un correo es o no spam en base de las probabilidades calculadas. Para esto, evalúa cada palabra del correo acumulando las probabilidades logarítmicas de que esa palabra pertenezca a un correo que sea spam o no. Usar los logaritmos en vez de multiplicar las probabilidades es importante para evitar problemas numéricos, dado que las probabilidades en Naive Bayes son por lo general muy pequeñas. La función nos devuelve una clasificación final en función de cuál de las dos probabilidades es mayor.

Otra función importante es la *probability words*, que calcula la probabilidad de que cada palabra del vocabulario aparezca en correos spam o no spam basándose en la frecuencia durante el entrenamiento. Usamos una técnica conocida como suavizado de laplace, que asegura que todas las palabras tengan una probabilidad distinta a cero, aunque no estén en el conjunto de entrenamiento. Esto es esencial para asegurarnos que el clasificador funcione adecuadamente con correos nuevos que puedan tener palabras no vistas anteriormente.

Por último, se incluyeron dos funciones que generan correos ficticios, uno que genera un correo spam y otro que genera uno no spam, basándose en las probabilidades calculadas para cada palabra. Estos correos, aunque no hagan sentido gramaticalmente, nos enseñan el tipo de palabras que se espera encontrar en cada una de las dos clasificaciones.

Resultados

A continuación, se presentan los resultados obtenidos mediante la aplicación del algoritmo Naive Bayes. Los resultados de la matriz de confusión resultante son:

- Correos clasificados correctamente como spam (True positives): 131
- Correos clasificados incorrectamente como spam (False positives): 5
- Correos clasificados correctamente como no spam (True negatives): 965
- Correos clasificados incorrectamente como no spam (False negatives): 13

Con los resultados obtenidos de la matriz de confusión se pueden calcular las métricas de rendimiento correspondiente, dando los siguientes resultados:

- Accuracy: 0.983842
- Precision: 0.963235
- Recall: 0.909722
- F1 Score: 0.935714

La *accuracy* obtenida es del 98.38%, lo que indica que el modelo clasifica correctamente el 98.38% del total de correos, ya sean spam o no spam. El modelo tuvo un desempeño general excelente. La *precisión* es del 96.23%, lo que indica que el modelo tiene una baja tasa de falsos positivos. En este contexto, significa que de todos los correos clasificados como spam, el 96.23% efectivamente lo eran, reduciendo la probabilidad de que correos no spam sean erróneamente etiquetados como spam. Por otro lado, el *recall* es del 90.97%, lo que indica que el modelo fue

capaz de identificar correctamente el 90.97% de los correos que realmente eran spam. Esto implica que, aunque el modelo es preciso, aún hay un pequeño porcentaje de correos spam que pueden no haber sido detectados (falsos negativos). Finalmente, el *F1 Score* es del 93.57%, una métrica que combina la *precisión* y el *recall*; es especialmente útil cuando es importante equilibrar los falsos positivos (correos no spam marcados como spam) y los falsos negativos (correos spam no detectados). En escenarios donde ambos tipos de error son críticos, el F1 Score proporciona una visión más completa del desempeño del modelo.

Conclusiones

Dadas las características de la base de datos, la cual está sumamente desequilibrada hacia la cantidad de correos de no spam que contiene, la medida de "accuracy" no sirve de mucho, pues una de sus características es que para que sirva, la base de datos tiene que estar equilibrada. En su lugar, se puede usar el F1-score, cuyo propósito es muy similar, pero si toma en cuenta el desequilibrio que hay en la base de datos; así brindando el rendimiento general del modelo.

En cada proyecto en el que se entrena un modelo, se tiene que pensar en qué métricas serán las que más importaran para medir su desempeño. En este caso, la métrica que más importa es el *recall*, ya que esta mide los falsos negativos específicamente. Esto debido a que se busca asegurar que todos los emails importantes lleguen, incluso si eso significa dejar pasar pasar un poco de spam.

El *recall* fue la métrica que más baja salió, pero su resultado aun así fue excelente con un valor de 0.909. Es decir, menos de 1 de cada 10 mails importantes es clasificado como spam. Es difícil mejorar esta taza más de a lo que ya se llegó, dado el desbalance de la base de datos. Pero incluso si estuviera balanceada, los resultados presentados ya están en un nivel excelente, porque las métricas son bastante altas y cualquier esfuerzo futuro por mejorar las puntuaciones puede resultar en overfitting.

Referencias

Academia Carta Blanca (2023). Explicación del Teorema de Bayes: fórmula y ejemplos. Retrieved from https://academiacartablanca.es/blog/teorema-de-bayes-probabilidad/

Gupta, K., Tewari, A., & Rathore, R. (2022). **A Survey on Spam Detection Techniques in Emails**. *International Journal of Information and Communication Technology*.

Jurafsky, D., & Martin, J. H. (2021). **Speech and Language Processing** (3rd ed.). Pearson.

Puthran, S., Rathod, V., & Sahu, S. (2020). **A Study on Spam Detection Using Naive Bayes Algorithm**. *International Journal of Computer Science Trends and Technology*.

Statista. (2023). **Global Email Spam Traffic Share 2007-2022**. Retrieved from https://www.statista.com.