NLP CLICKBAIT

25 DE ABRIL DE 2022

Grupo: Manuel Tendero, Weronica Jaroszek, Isaac Nicolás, José Picón

1. **El problema: Describir el 'problema' que desea abordar. Especialmente, detallad el dominio. Explicitando toda la complejidad que podría llegar a ofrecer (independientemente de cómo lo abordéis en este proyecto en concreto).**

El *clickbait* que ha sido traducido al español como “ciberanzuelo”, se usa de forma peyorativa para describir a los contenidos en internet que persiguen generar ingresos publicitarios usando titulares de maneras sensacionalistas y engañosas para atraer la mayor proporción de clics posibles.

Nuestro grupo pretende abordar un reto de clasificación de un conjunto de titulares de noticias en clickbait o no clickbait.Los datos se recopilan de varios sitios de noticias.
Los titulares de clickbait se recopilan de sitios como 'BuzzFeed', 'Upworthy', 'ViralNova', 'Thatscoop', 'Scoopwhoop' y 'ViralStories'. Por otro lado, los titulares relevantes o que no son clickbait se recopilan de muchos sitios de noticias confiables, como 'WikiNews', 'New York Times', 'The Guardian' y 'The Hindu'.

El dominio se centra en la gran cantidad de información a la que se ven expuestos los usuarios. El reto que persigue el presente trabajo es poder ofrecer una confianza a los usuarios de si están delante de noticias verdaderas o de clickbait.

2. **La aplicación: Identificar qué tipo de aplicación / aplicaciones estáis abordando: clasificación, clustering, asociación, etcétera.**

Las técnicas que aplicaríamos para hacer frente el reto descrito anteriormente sería el Procesamiento de lenguaje natural. El objetivo final de nuestros algoritmos sería la capacidad de clasificar las noticias según su categoría, es decir en función de si sus titulares son clickbait o no. Se trata en este caso, de un problema de clasificación binaria.

3. **El dataset: Detallar el proceso de recolección de los datos. Y la descripción y el tratamiento de los mismos.**

El set de datos que vamos a utilizar se ha extraído de la página kaggle. Este conjunto de datos contiene titulares de varios sitios de noticias como 'WikiNews', 'New York Times', 'The Guardian', 'The Hindu', 'BuzzFeed', 'Upworthy', 'ViralNova', 'Thatscoop', 'Scoopwhoop' y 'Historias virales'. Tiene dos columnas, la primera contiene titulares y la segunda tiene etiquetas numéricas de clickbait en las que 1 representa que es clickbait y 0 representa que no es un titular de clickbait. El conjunto de datos contiene un total de 32000 filas, de las cuales el 50 % son clickbait y el otro 50 % no son clickbait, por lo que sabemos que el dataset está balanceado en ambas categorías.

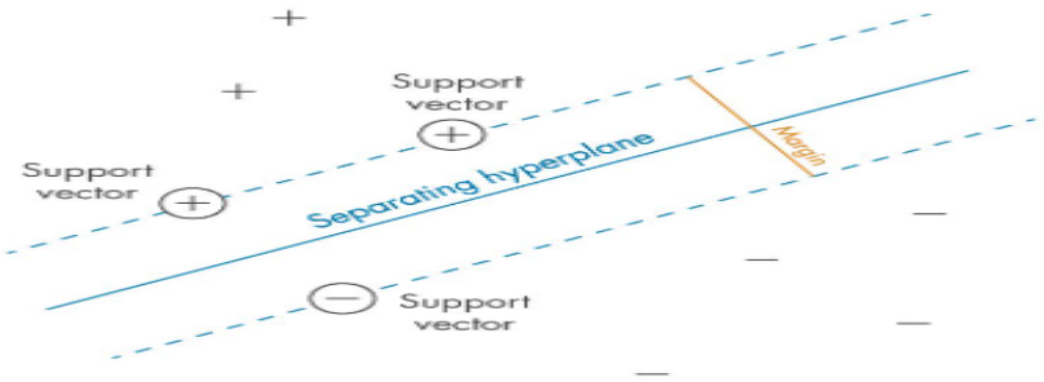
4. **El algoritmo o algoritmos: Elegid y justificad qué algoritmos son los más adecuados para extraer información del dataset. Si los algoritmos se deben parametrizar, deberéis pensar o experimentar sobre los mejores valores para los parámetros. Recordad que, no necesariamente, se debe implementar el algoritmo. Podéis usar fuentes ya implementadas o herramientas como el Weka/R/Python**

Para clasificar el texto de los titulares de las noticias finalmente se han usado los siguientes Algoritmos:

Support Vector Machine

Support vector machine (SVM) es un algoritmo de aprendizaje automático que es muy utilizado en problemas de clasificación y regresión. El objetivo principal de este algoritmo es encontrar un hiperplano donde se separen dos clases diferentes de puntos de datos. Este algoritmo solamente encuentra este hiperplano en problemas que permiten una separación lineal. En la mayoría de casos, el modelo maximiza el margen flexible permitiendo un pequeño número de los vectores de soporte que hacen referencia a un subconjunto de observaciones de entrenamientos. La versión estándar de este algoritmo está diseñado para problemas de clasificación binaria. Este modelo pertenece a una clase de algoritmos de ML conocidos como métodos kernel. Las funciones de kernel asignan los datos a un espacio dimensional diferente, que suele ser superior, con el objetivo de que resulte más fácil la separación de las clases después de esta transformación, simplificando los límites de decisión más complejos (Mathworks, sf).

Imagen 1. Definición del “margen” entre clases: el criterio que los SVM intentan optimizar



Fuente: mathworks (sf)

Haciendo referencia a los creadores de este algoritmo, Boser, Guyon y Vapnik (1992), este modelo ajusta automáticamente la capacidad de la función de clasificación maximizando el margen entre los ejemplos de entrenamiento y el límite de la clase. La función de clasificación depende solo de los llamados patrones de apoyo. Estos patrones de apoyo o support vector son aquellos ejemplos de entrenamiento que están más cercanos al límite de decisión y suelen ser un pequeño subconjunto de los datos de entrenamiento. El modelo opera con una gran clase de funciones de decisión que son lineales en sus parámetros pero que no se limitan a dependencias lineales en los componentes de entrada.

Naive-Bayes

El clasificador Naive-Bayes es un algoritmo que se basa en el teorema de Bayes para la clasificación de los datos. En la detección de clickbait se cuenta el número de veces que aparece una palabra en un titular de clickbait. Después, se calcula la probabilidad de que un titular sea clickbait en comparación con la probabilidad de que sea un titular real (Sim, 2020).

Siguiendo a Yildirim (2020), este algoritmo utiliza funciones para predecir la variable objetivo al igual que otros algoritmos de aprendizaje supervisado. La diferencia recae en que el algoritmo Naive-Bates asume que las características son independientes entre sí y no hay correlación entre las características. El desempeño de este algoritmo se basa en el cálculo de la probabilidad de una clase dado un conjunto de datos característicos. La suposición de que todas las funciones son independientes hace que el algoritmo bayesiano ingenuo sea **más rápido** en comparación con otros algoritmos. Además, tiene un buen desempeño con datos de gran dimensión, como clasificación de texto o detección de correo no deseado. Por otro lado, la suposición de que todas las funciones son independientes hace que el algoritmo bayesiano sea menos preciso que los otros algoritmos.

XGBoost

XGBoost es un clasificador de árbol de decisión con refuerzo de gradiente. En boosting, los árboles se construyen secuencialmente de forma que cada árbol posterior tiene el objetivo de reducir los fallos del árbol anterior. Estos árboles posteriores se llaman aprendices débiles. Cada uno de estos aprendices aporta una información relevante para la predicción. Esto permite a la técnica de boosting producir un aprendiz fuerte resultante de la combinación eficaz de los aprendices débiles.La potencialidad de este algoritmo se basa en su escalabilidad, que fomenta un

aprendizaje rápido a través de la computación paralela y distribuida con un uso eficiente de memoria (Diveki, 2019).

Siguiendo a Greatlearning (2020), dentro de las características de este algoritmo encontramos: un aprendizaje regularizado donde el término de regularización ayuda a suavizar los pesos finales aprendidos para evitar el sobre ajustamiento; aumento de gradiente de árbol donde el modelo se entrena de forma aditiva debido a que el modelo de conjunto de árbol no se puede optimizar utilizando métodos de optimización tradicionales en el espacio euclidiano. Por último, encontramos el submuestreo de columna y contracción. Además del objetivo regularizado, se utilizan técnicas adicionales para evitar aún más el sobreajuste.

Modelo Preentrenado Spacy

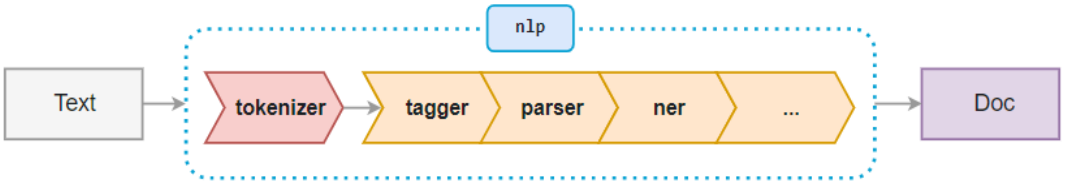
SpaCy es una biblioteca gratuita de código abierto para el procesamiento avanzado del lenguaje natural (NLP) en Python. spaCy está diseñado específicamente para uso en producción y lo ayuda a crear aplicaciones que procesan y "entienden" grandes volúmenes de texto.

A continuación se detallan las características y funcionalidades de Spacy según su documentación oficial Spacy(2022):

- **Tokenización**: segmenta el texto en palabras, signos de puntuación etc.
- **Part-of-speech (POS)**: asigna tipologías de palabras a tokens, como verbo o sustantivo
- **Dependency Parsing**: asigna etiquetas de dependencia sintáctica, que describen las relaciones entre tokens individuales, como sujeto u objeto-
- **Lemmatization**: asignación de la forma básica de las palabras.
- **Sentence Boundary Detection (SBD)**: encuentra y segmenta oraciones individuales
- **Named Entity Recognition (NER)**: etiqueta objetos del mundo real con nombre, como personas, empresas o ubicaciones.
- **Entity Linking (EL)**: elimina ambigüedades de entidades textuales a identificadores únicos en una base de conocimientos
- **Similarity**: compara palabras, extensiones de texto y documentos y qué tan similares son entre sí.
- **Text Classification**: asigna categorías o etiquetas a un documento completo o partes de un documento.
- **Rule-based Matching**: encuentra secuencias de tokens en función de sus textos y anotaciones lingüísticas, similares a las expresiones regulares
- **Training**: actualiza y mejora las predicciones de un modelo estadístico.
- **Serialization**: guardará objetos en archivos o cadenas de bytes.

Por otro lado, esta librería también nos proporciona otras funcionalidades como la de pipeline o tubería. Esto hace referencia cuando se llama nlp a un texto, spacy primero tokeniza el texto para producir un Doc objeto. Luego se procesa en varios pasos diferentes; esto también se conoce como canalización de procesamiento. La canalización utilizada por los modelos predeterminados consta de un etiquetador, un analizador y un reconocedor de entidades. Cada componente de canalización devuelve el procesado Doc, que luego se pasa al siguiente componente (spaCy, 2022).

Imagen 2. Estructura y funcionamiento de una tuberia (pipline)



Fuente: Spacy annotations (2022).

5. **Proceso: Detallad el proceso global de vuestro proyecto.**

Primer paso consistiría en preprocesar el texto, eliminar toda aquella información que introduzca ruido como por ejemplo las mayúsculas, contracciones de palabras, conjugaciones, “stopwords”, signos de puntuación, espacios vacíos, dígitos, etc. Segundo paso consistiría en analizar y entender el conjunto de palabras. A continuación, el siguiente paso consiste en vectorizar las palabras de nuestro conjunto de datos y por último se procedería a entrenar diferentes modelos y entender y comparar los resultados.

6. **Diseño experimental: deberéis pensar cómo se debe entrenar el algoritmo y cómo se testea. Esto incluye decidir qué métricas de evaluación se usarán y qué tipo de estimadores (holdout, cross-validation), etc. Diseñad el juego de pruebas que convendría para validar vuestro sistema.**

Nuestro conjunto de datos será divido en dos partes, una parte de train y otra de prueba y la métrica será la accuracy. También usaremos visualizaciones como la matriz de confusión, con tal de tener una idea de qué instancias son las que se clasifican erróneamente y también usaremos la curva ROC. Por otro lado, hemos de tener en cuenta que los modelos mencionados constan de parámetros que vamos a tener que optimizar usando métodos iterativos como GridSearchCV de Sklearn.

7. **Resultados y análisis: una vez pensado el diseño experimental, se deberá realizar las pruebas, obtener los resultados y analizar la información que habéis extraído, el porqué, etc.**

Tras la aplicación de los modelos a nuestro set de datos, se exponen a continuación los resultados obtenidos.

Tabla 1. Resultados de los modelos aplicados

Metodo	Acurracy	Precision	Recall	F-1 Score	ROC-AUC
Support Vector Machine	95%	95%	95%	95%	0.96
Naive Bayes	96%	96%	96%	96%	0.96
Xgboost	84%	85%	85%	85%	0.85
spaCy	97%	97%	97%	97%	----

Fuente: Elaboración propia

En cuanto a las métricas de evaluación que comparten los cuatro modelos, se observa que los resultados son positivos y altos. Esto puede deberse a que el set de datos con el que hemos trabajado proveniente de la página web Kaggle, es un dataset perfectamente balanceado y cuya preparación pueda perseguir unos resultados óptimos. Por otro lado, la elección correcta de los hiperparametros de cada algoritmo puede favorecer en los resultados obtenidos. En cuanto a las métricas de Accuracy, Precision, Recall y F-1 Score el mejor algoritmo es el de spaCy. Aunque los resultados no difieren mucho entre los diferentes modelos, esta mayor puntuación de spaCy también puede estar relacionada con las funcionalidades específicas de este modelo en el preprocesamiento del texto y su aplicación del modelo.

Por otro lado, en cuanto a la curva Roc y el área bajo la curva, podemos observar que los modelos SVM y Naive Bayes consiguen una misma puntuación de 0.96. Esto nos está indicando que nuestros modelos son efectivos en la distinción de las categorías de titulares.

Para el modelo de spaCy encontramos el parámetro de Loss o pérdida. Este hace referencia a la minimización del error. Loss es un valor resultante en donde la función de pérdida es reducida a un solo número, un valor escalar que nos permite clasificar y comparar las conclusiones candidatas (Brownlee, 2019). De las cinco iteraciones utilizadas en el modelo de spaCy se ha obtenido un valor máximo de 7.3 y un valor mínimo de 0.23.

8. Referencias

- Potthast, M., Köpsel, S., Stein, B. y Hagen, M. (2016). Detección de clickbait. ECIR .
- Abhijnan Chakraborty , Bhargavi Paranjape , Sourya Kakarla , Niloy Ganguly- https://arxiv.org/abs/1610.09786
- Mathworks(sf)https://es.mathworks.com/discovery/support-vector-machine.html
- (paper original) Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh (1992)
- Sim(2020).https://towardsdatascience.com/using-bayesian-classifiers-to-detect-fake-news-3022c8255fba
- Yildirim(2020),https://towardsdatascience.com/naive-bayes-classifier-explained-50f9723571ed#:~:text=Naive%20Bayes%20is%20a%20supervised,prediction%20on%20a%20target%20variable.
- DIVEKI(2019)https://www.kaggle.com/code/diveki/classification-with-nlp-xgboost-and-pipelines/notebook
- (greatlearning,2020)https://www.mygreatlearning.com/blog/xgboost-algorithm/#:~:text=XGBoost%20is%20an%20optimized%20distributed,a%20fast%20and%20accurate%20way.
- (Chaudhary,2020)https://medium.com/@cmukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a
- SPACY
- (programmerclick, sf). https://programmerclick.com/article/37861127741/
- (Jayaswal,2020)https://towardsdatascience.com/laplace-smoothing-in-na%C3%AFve-bayes-algorithm-9c237a8bdece
- Fray, 2018. https://medium.com/all-things-ai/in-depth-parameter-tuning-for-svc-758215394769
- Wikipedia,2020 https://en.wikipedia.org/wiki/Regularization_%28mathematics%29#Regularization_in_statistics_and_machine_learning
- (DmlcXgboost,2022).https://xgboost.readthedocs.io/en/latest/python/python_api.html#xgboost.XGBClassifier
- (Spark,2017).https://blog.cambridgespark.com/hyperparameter-tuning-in-xgboost-4ff9100a3b2f
- (Shrivarsheni,2020)https://www.machinelearningplus.com/nlp/custom-text-classification-spacy/