Proyecto de big data.
Clasificador de noticias por categoría

Daniel Enrique Pazmiño Ortega
Juan Felipe Mazo Sanchez
Universidad Autónoma de Manizales
Manizales, Colombia
daniele.pazminoo@autonoma.edu.co
juanf.mazos@autonoma.edu.co

***Resumen*— En este informe se presenta la elaboración de un clasificador de noticias por categoría haciendo uso del machine learning mediante Spark y sus instancias como el procesamiento de los datos, el entrenamiento del modelo,el testeo de los mismos, uso de una librería para interactuar con el cuaderno los resultados y las conclusiones que se pudo sacar a través de este proyecto, además de los códigos empleados en los anexos.**

Palabras clave: Entrenamiento de máquina, Naive Bayes, clasificador mediante machine learning, TF-IDF.

I. INTRODUCCIÓN

A través de los años y con el acogimiento de internet y la tecnología, se ha implementado sistemas para facilitar el manejo de los datos, por ejemplo las EPS en colombia que necesitan tener la información de todos sus afiliados y para ello es necesario hacer uso de bases de datos, pero trabajar con toda esa información se vuelve tedioso y es ahí en donde entra Big Data siendo un conjunto de técnicas que ayudan al manejo de grandes volúmenes de datos y dependiendo de la aplicación se pueden elegir las herramientas a usar.

El big data nace con esa necesidad de facilitar ese manejo de mucha información y generar valor para los usuarios, ya sea en la toma de decisiones, en el análisis de la información y otras aplicaciones; en ocasiones los datos guardados se deben tratar de cierta forma para que sea compatible con las técnicas o tenga mejor eficacia a la hora de analizar los resultados, a esto se le llama preprocesamiento de datos, en esta sección se puede quitar ciertas palabras, ejecutar filtros, generar agrupaciones entre otras cosas y el utilizado en el proyecto TF-IDF.

Además de las herramientas que nos proporciona el big data se puede hacer uso de el entrenamiento de máquina para generar respuestas ante una entrada sin la necesidad de la intervención humana, a esto se le conoce como Machine learning que a groso modo se basa en que el pc observa un conjunto de datos construye un modelo con esos datos (operaciones estadísticas) y utiliza ese modelo como una hipótesis acerca del mundo y una pieza de software que puede resolver problemas [1].

El machine learning tiene varias aplicaciones y una de ellas es la clasificación, la cual como su nombre lo indica se encarga de separar y seleccionar datos por similitudes o características compartidas esto tiene varias aplicaciones como lo puede ser un buscador de un motor como google o una barra de búsqueda en un sitio web de compras entre otros y uno de los modelos utilizados para la clasificación es *naive bayes*. este asume que el efecto de una característica particular en una clase es independiente de otras características. Por ejemplo, un solicitante de préstamo es deseable o no dependiendo de sus ingresos, historial de préstamos y transacciones anteriores, edad y ubicación[2].

II. METODOLOGÍA

Como primera medida se eligió la base de datos a usar que fue un dataset presente en Kaggle titulado *BBC News Classification.* En este se usa un conjunto de datos públicos de la BBC compuesto por 2225 artículos, cada uno etiquetado en una de las 5 categorías: negocios, entretenimiento, política, deportes o tecnología.

El conjunto de datos se divide en 1490 registros para entrenamiento y 735 para prueba, cada división en un archivo csv diferente. El archivo de entrenamiento tiene 3 columnas: id del artículo, el cuerpo de la noticia y la categoría, para el caso del archivo de entrenamiento no tiene la columna de categoría. El objetivo fue construir un sistema que pueda clasificar con precisión artículos de noticias nunca antes vistos en la categoría correcta.Para lograr el objetivo se usaron principalmente herramientas de Pyspark directamente relacionados con el aprendizaje de máquina.

Como primera medida se crea una sesión de spark con ayuda del constructor, seguidamente se carga el archivo, el cual está guardado en el entorno de Google Colab y se hace una separación del archivo que brinda Kaggle en un 70% entrenamiento y un 30% de testeo. Ahora bien, se debe hacer una preparación de los datos que consta de varios pasos, como lo son separar el texto por palabras, remover stopwords (palabras que por sí solas no tienen significado), realizar una vectorización de las palabras, aplicar el idf a la vectorización realizada (esta operación da como resultado el TF-IDF) , todo esto mediante funciones que brinda la herramienta y añadiendo cada una los elementos en una columna nueva. Adicionalmente se realizó un tratamiento a la categoría que constó de asignarle un valor numérico del 0 al 4 mediante la función *StringIndexer* a la cuál se especificó que lo hiciera en orden alfabético ascendente, esto con el fin de poder establecer fácilmente a qué categoría corresponde cada número y por último se utilizó una función de machine learning de pyspark que se usa para clasificar llamado Naive Bayes en el que le ingresamos la columna de categoría y la columna donde está el TF-IDF.

Todos los pasos anteriores se integran dentro de un pipeline con el cual se crea el modelo completo haciendo el entrenamiento a este modelo con la parte de testeo del archivo de entrenamiento. Con el modelo creado el siguiente paso es aplicar el modelo a la sección de prueba del archivo de entrenamiento, generando una predicción, además se realiza una evaluación del modelo, calculando la precisión general y la precisión de acuerdo a cada categoría

Por último, se creó un botón con las librerías de ipywidgets

para cargar archivos desde el computador, además, en la función que se ejecuta con el botón se guarda el archivo en el entorno de Colab, se genera la predicción para este archivo con el modelo entrenado, se cambia el número que genera la predicción por la categoría que le corresponde haciendo uso de un diccionario y fue necesario que el dataframe de predicción se pasará a rdd para ejecutar este cambio. Como última instancia se imprime el archivo ingresado con la categoría que predijo el modelo y se imprime el número de noticias por categoría.

III. RESULTADOS

Con el modelo entrenado se obtuvo una precisión general de 96.8%

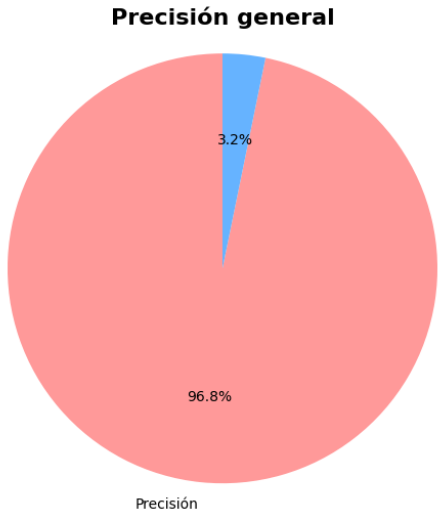

Figura 1. Diagrama de torta - precisión general

Adicionalmente, la precisión por categorías está dado de la siguiente manera:

- Bussines: 95%
- Entertainment: 97%
- Politics: 95%
- Sports: 99%
- Tech: 97%

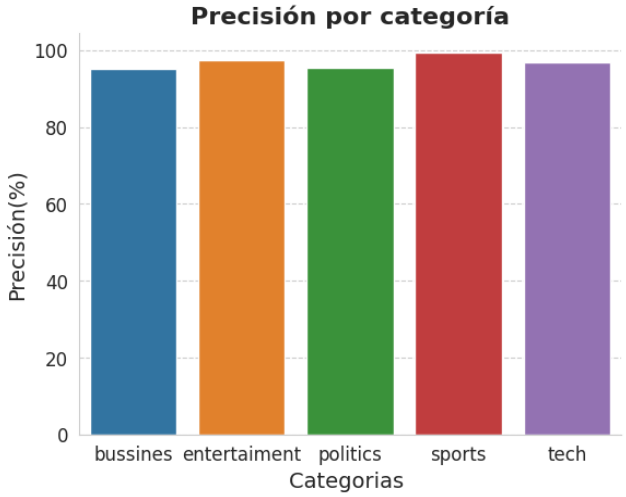

Figura 2. Diagrama de barras - precisión por categoría

Categorías	Conteo
Business	168
Entertainment	110
Sports	165
Tech	143
Politics	149

Tabla 1. Conteo de noticias por categoría

IV. CONCLUSIONES

- El modelo alcanzó una precisión aceptable dadas las condiciones y al tratamiento simple que se le aplicó.
- El hecho de que la precisión de la categoría sports haya sido del 99% se puede atribuir a que para esta se maneja un lenguaje con algunos términos diferenciables en comparación a las otras
- La precisión para las categorías business y politics fue menor en comparación a las otras y esto se podría deber a la ambigüedad que presentan las noticias de este tipo con términos como ejemplo PIB, que puede interpretarse tanto para business como para politics.
- Hacer una clasificación de un conjunto de datos conlleva segmentar, comparar y agrupar lo cual puede ser engorroso y la cantidad de información puede llegar a ser muy grande, lo cual hace que intentar realizarlo de forma manual sea una tarea casi imposible, es allí donde el Big Data y el Machine Learning cobran especial relevancia, optimizando este proceso y permitiendo generar un valor a esta tarea, no solo para la clasificación de textos, sino para otro tipo de aplicaciones en donde este sea solo un paso dentro del modelo como puede ser un sistema de detección de alertas tempranas clasificando ciertas características sea capaz de detectar que el sistema va a fallar en algún rango de tiempo.
- A partir del conteo que se realiza a cada categoría se podría llegar a observaciones valiosas para el contexto donde se realice la predicción, por ejemplo, si todas las noticias son de un periodo de tiempo en específico podrían llegar a identificarse sucesos extraordinarios como pueden ser descubrimientos tecnológicos importantes, crisis económicas, eventos deportivos de gran alcance, eventos de belleza o de moda, temporada de elecciones.

REFERENCIAS

[1] “Artificial Intelligence: A Modern Approach, 4th US ed,” Berkeley.edu. [Online]. Available: http://aima.cs.berkeley.edu.

[Accessed: 14-May-2023].

[2] L. Gonzalez, “Naive Bayes – Teoría,” Aprende IA, 20-Sep-2019. [Online]. Available: https://aprendeia.com/algoritmo-naive-bayes-machine-learning/. [Accessed: 14-May-2023].

[3] K. Sparck Jones, “A statistical interpretation of term specificity and its application in retrieval,” J. Doc., vol. 28, no. 1, pp. 11–21, 1972.