Identificación de los sesgos ideológicos en colecciones literarias nacionales entre 1942 y 1958, utilizando técnicas de Natural Language Processing (NLP) y modelos de clasificación.

Eliana Donneys Bastidas
Natalia Viáfara Delgado

Trabajo de grado para optar al título de
Máster en Ciencia de Datos

Director:
Andrés Aristizábal

Co-director:
Uram Aníbal Sosa

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS
MAESTRÍA EN CIENCIA DE DATOS
SANTIAGO DE CALI
2023

Resumen

En el presente trabajo de grado se formuló una propuesta en aras de abordar la ausencia de herramientas para identificar, automáticamente, los sesgos ideológicos en colecciones literarias nacionales entre 1942 y 1958 en Colombia. Para hacerlo, se propuso una metodología que consideraba:

A) La selección de los libros que fueron escritos en los siglos donde se desarrolló el conservadurismo y el liberalismo
B) La clasificación de los libros en dos categorías (liberal y conservador), basado en el gobierno que lo publicó
C) La búsqueda de los libros, sea encontrando libros digitalizados o digitalizando aquellos que no lo están
D) La extracción del texto de los libros digitalizados para su posterior exploración
E) La limpieza de los datos obtenidos y su posterior procesamiento, empleando técnicas de Natural Languaje Processing (NLP)
F) La vectorización de la información textual usando algoritmos y técnicas de NLP
G) El uso de modelos de clasificación para predecir la clase a la que pertenecen los libros, usando los vectores obtenidos anteriormente
H) Evaluación del rendimiento del modelo basado en las siguientes métricas: ROC AUC, accuracy, recall, especificidad y Kappa de Cohen

Los modelos, técnicas y herramientas de la ciencia de datos usados para abordar la solución al problema fueron:

1. El uso de herramientas de reconocimiento óptico de caracteres (OCR) para obtener el texto de los libros dentro de los archivos PDF.
2. Se usó el lenguaje de programación Python como herramienta base para todo el proceso posterior, iniciando con la extracción de texto de los PDF.
3. Para la identificación de temas se emplearon 3 técnicas de NLP orientadas al modelado de temas (Topic modeling), las cuales se mencionaron y explicaron previamente en el marco teórico: a) LDA (Latent Dirichlet Allocation), b) TF-IDF (Term Frequency-Inverse Document Frequency) y c) NMF (Non-negative Matrix Factorization).
4. Para la vectorización de la información textual se usó el algoritmo de NLP, roBERTa.
5. Se probaron los siguientes modelos de clasificación: LDA (Linear Discriminant Analysis), QDA (Quadratic Discriminant Analysis), SVC (Support Vector Classification), Regresión logística, ElasticNet Linear Regression, XGBoost (Extreme Gradient Boosting), MLP Classifier (Multi-Layer Perceptron y Árbol de decisión.
6. Los modelos se entrenaron usando el protocolo de validación Leave One Out Cross Validation (LOOCV), buscando maximizar el valor de la precisión (accuracy) empleando optimización bayesiana.

Los resultados obtenidos muestran que el modelo con mejor rendimiento fue el Support Vector Classifier (SVC), con un ROC AUC de 0.9679, accuracy de 0.8929, recall de 0.8182, especificidad de 0.9412 y un Kappa de Cohen de 0.7717.

Tabla de contenido

1. Integrantes y directores del trabajo de grado.

Integrante(s):
Eliana Donneys Bastidas
Natalia Viáfara Delgado

Director: Andrés Aristizábal
Codirector: Aníbal Sosa

2. Título del trabajo de grado.

Identificación de los sesgos ideológicos en colecciones literarias nacionales entre 1942 y 1958, utilizando técnicas de Natural Language Processing (NLP) y modelos de clasificación.

3. Contexto y antecedentes.

En el presente proyecto de grado se tomará como unidad de análisis 40 de los 113 títulos de la Biblioteca de Autores Colombianos (BAC) y 70 de los 161 títulos de la Biblioteca Popular de Cultura Colombiana (BPCC), las cuales contienen diversos géneros literarios como novelas, poesía, historia, biografías, entre otras. Ambas colecciones fueron aportes editoriales muy relevantes para la historia de Colombia, los cuales se generaron durante el período comprendido entre 1942 a 1958, bajo el liderazgo de gobiernos liberales y conservadores de turno. Dichos compendios literarios tenían como objetivo la difusión de la cultura a los ciudadanos y ciudadanas alfabetizadas de la época, a través de la publicación de obras de autores colombianos y extranjeros a precios asequibles, tanto así que para algunos ejemplares se llegó a emplear el trueque o intercambio.

La Biblioteca Popular de Cultura Colombiana (BPCC) fue creada desde 1942 y contempló más de once series de impresiones. Esta colección se gestó durante los gobiernos liberales de Eduardo Santos y Alberto Lleras Camargo, y su ministro de educación Germán Arciniegas -quien también fue editor de la colección-, como una iniciativa para acercar la cultura a las clases populares del país. Según la investigadora Paula Andrea Marín (2017) del Instituto Caro y Cuervo:

> Arciniegas asume el cargo de ministro de Educación (para los gobiernos de la República Liberal: 1930-1946), cuando el liberalismo vuelve a asumir el poder político en el país, tras el período de la Regeneración y de la Hegemonía Conservadora (1886-1930). Dos de las campañas más importantes de los gobiernos liberales de esta época fueron la ampliación de la cobertura educativa y el incremento de las tasas de alfabetización. Lo anterior no quiere decir que en los años precedentes no se hubiese hecho adelantos en estos aspectos, pero, entre 1930 y 1946, el aumento en el presupuesto para el funcionamiento del Ministerio de Educación resultó clave en la ejecución de estas campañas, pese a que este aumento no haya sido suficiente para cumplir con todas las metas propuestas. La atención a la cobertura educativa y a las campañas de alfabetización estuvo acompañada por lo que se podría resumir como un proyecto para hacer del libro un objeto cotidiano entre las “masas populares”. Dicho proyecto se concentró en la creación de las Bibliotecas Aldeanas y de las Misiones Culturales, desde 1934, y

> tuvo entre sus resultados más importantes el incremento del autodidactismo y la búsqueda del lector por fuera del espacio oficial de la biblioteca (Silva, 2012: 63, 143). Durante este período también se introducirán los mayores cambios acerca del reconocimiento y pago de los derechos de autor en la primera mitad del siglo XX en Colombia. Arciniegas será, pues, uno de los artífices de estas transformaciones en el ámbito social y cultural colombianos, cuando asuma la ejecución del proyecto de la BPCC.

Ahora bien, posteriormente este proyecto liberal de colección literaria es heredado por el gobierno conservador de Laureano Gómez y su ministro de educación, Rafael Maya. Sin embargo, estos no le dan la continuidad esperada, y por el contrario crean un compendio editorial propio que se constituye como la Biblioteca de Autores Colombianos (BAC) (Marín Colorado, 2017):

> Cuando ocurre el cambio de los gobiernos liberales a, nuevamente, los conservadores (1946), la dirección de la BPCC pasa a manos del poeta Rafael Maya (…) Entre 1946 y 1950, no se pone en circulación ningún volumen de la BPCC y es solo hasta este último año que se reactiva el proyecto de esta colección y se publica el resto de los volúmenes que había dejado preparados Arciniegas (serie doce). Entre 1951 y 1952, se publican los últimos títulos de la BPCC: la última serie preparada por Arciniegas y 31 títulos más (productos del criterio de selección de Rafael Maya); precisamente, este año marcará el inicio de una nueva colección oficial: la Biblioteca de Autores Colombianos, que se puede entender como una respuesta a la BPCC, elaborada con un criterio de selección más conservador, en relación con las tradiciones literaria e histórica colombianas

Ambas bibliotecas fueron fundamentales en la difusión de la literatura y la cultura colombiana en el siglo XX. La BAC publicó obras de autores como Jose Eusebio Caro, Rafael Pombo, Víctor Frankl y José Asunción Silva, mientras que la BPCC publicó obras de autores como Francisco José de Caldas, Jorge Isaacs y Tomás Carrasquilla. Sin embargo, es importante cuestionar el fin de estos esfuerzos de impresión editorial desde el marco estatal, como lo señala Gloria J. Morales (2020):

> Sea que se piense como macropráctica o como estrategia, el rol de Arciniegas y Maya es central en esta historia intelectual: ambos pertenecen a la generación de los años 20 en Colombia y colaboraron en una revista cultural llamada Los Nuevos, que proponía la necesidad de renovar literariamente el país, alejándose del modernismo nacionalista e hispanista y del bipartidismo (Rodríguez Morales)(...)
>
> En un contexto de alfabetización tardía, pero en el que las políticas culturales liberales de los años 30 habían allanado un poco el camino para que los libros nacionales llegaran a todos los rincones del país, presenciamos esta empresa canónica y editorial de reimprimir la historia letrada del país en 161 (Marín Colorado, Un Momento 33) y 113 títulos, respectivamente.

4. Planteamiento del problema y justificación

Ahora bien, es menester destacar que el presente proyecto de grado se ha planteado como parte de una disertación de tesis doctoral en literatura de la autora Gloria Morales (2020). Por tanto, la justificación y el planteamiento del problema están alineados con el trabajo en mención, el cual otorga una preponderancia especial a los sucesos acontecidos en la historia colombiana comprendida entre el periodo de 1942 y 1958. Así pues, a continuación, se presentarán algunos apartados de la tesis doctoral de la que se parte para entender el problema y la justificación del mismo (Morales Osorio, 2020):

> "Para algunos historiadores de las ideas, la violencia política bipartidista de mitad del siglo XX borroneó la potencia cultural del país (Sánchez Gómez, 1998) y extravió la función de los intelectuales en la sociedad colombiana (Loaiza Cano, 2014). Lo que estas colecciones señalan es que antes y después del 9 de abril de 1948, fecha en que se inició el estallido social violento conocido como El Bogotazo, algunos intelectuales lideraron macroprácticas culturales de difusión del libro y usaron su lugar estratégico como editores del Estado para reconstruir un pasado nacional determinado (…) pues mientras las colecciones BPCC y BAC se seleccionaban, se confeccionaban y circulaban por el país y otras regiones, Colombia experimentaba una polarización política creciente, el estallido social del Bogotazo (1948), una guerra civil bipartidista de casi diez años y el inicio de una dictadura militar en 1957. Junto con la Selección Samper Ortega (1928-1937), la BPCC y la BAC se ubican como esfuerzos estatales exclusivos de ordenar y serializar textos para recontar el pasado nacional en medio del caos social que produjeron sucesivas guerras civiles (Pineda 70).
>
> Este es un esfuerzo de la ciudad letrada de rescribir una historia nacional título a título que, en palabras de De Certeau (1996) se catalogaría como una estrategia de la vida social, en la que agentes como Maya o Arciniegas -los ministros de educación- usaron un principio de poder para construir un lugar propio y un orden de cosas a través de libros."

Ahora bien, lo anterior, tomando las ideas centrales de Morales, nos hace preguntarnos si **¿es posible comprobar que los sesgos ideológicos (liberal y conservador), consolidados en unas versiones puntuales de historia literaria, rigen estas colecciones curadas por Germán Arciniegas y Rafael Maya?**

La justificación del presente proyecto de grado toma en cuenta varios aspectos. En primer lugar, el hecho de que dos intelectuales, en sus complejas relaciones con el poder estatal, diseñaran e imprimieran colecciones de libros que proponen versiones particulares del pasado nacional. Esto es un problema para los sujetos subrepresentados en la historia del país y la literatura. Como lo afirma (Morales Osorio, 2020):

> "El esfuerzo de Maya y Arciniegas de crear una colección literaria, en un contexto de vulnerabilidad y desigualdad (Moretti, 2013), institucionaliza una versión particular de la literatura colombiana —sin voces indígenas, escasa nominación afrocolombiana y poca representación femenina. Ahora bien, lo enunciado también puede ser un problema para las personas interesadas en profundizar sobre la historia política y literaria del país, a través del contenido de las colecciones de libros y las influencias que moldearon estas colecciones.

En segundo lugar, las variables ideológicas cobran importancia al dictar qué libros la ciudadanía debería leer para conocer su historia y su presente, personificando a los partidos políticos con agencia (Latour, 2001) y voluntad de memoria que privilegian autoritariamente unos títulos sobre otros. En consecuencia, las colecciones mencionadas tenían el poder de influenciar el pensamiento de las masas alfabetizadas de la época, las cuales generalmente pertenecían a la élite, influían en la toma de decisiones e impartían el conocimiento (maestros, doctores, etc.) En este contexto, se emplea estratégicamente la cultura impresa, marcando debates sociales sobre quién puede escribir y leer, qué se debe leer y qué intelectuales están legitimados para decidirlo. Como lo sostiene (Morales Osorio, 2020):

> "La BPCC y la BAC se ubican como esfuerzos estatales exclusivos de ordenar y serializar textos para recontar el pasado nacional en el periodo del Frente Nacional. Lo anterior, en un contexto de alfabetización tardía, en el que el 41% de la población no sabía leer ni escribir (Banco de la República, 2013), estos intelectuales reeditaron y distribuyeron gratuitamente o con un valor simbólico de un peso (Marín Colorado, 2017) libros coleccionables escritos por autores y autoras colombianos(…) Dichas colecciones, cuyo contenido posiblemente dependía del partido político de turno, tenían la capacidad de moldear imaginarios ideológicos y establecer una narrativa particular de la historia nacional inclinada a los intereses partidistas. Esta revisión de textos de dos colecciones diferentes (liberal y conservadora), ofrece una mirada amplificada de la historia de Colombia, y contribuye a la comprensión de un periodo convulso en la historia nacional"

Debido a lo anterior, es relevante el desarrollo de modelos de analítica de datos no estructurados (libros) como parte de la disertación doctoral, pues permitiría una comprensión más profunda y amplia de las narrativas ideológicas y las perspectivas históricas que se transmiten a través de las colecciones de libros en cuestión. Al aplicar técnicas de procesamiento del lenguaje natural (de ahora en adelante NLP, por sus siglas en inglés, *Natural Language Processing*), se pueden identificar patrones y tendencias en los textos, como las palabras y los temas más relevantes, al igual que convertir los contenidos de los libros en vectores numéricos, que posteriormente pueden ser utilizados para la ejecución de algoritmos de clasificación. Estas herramientas pueden ayudar a descubrir sesgos y exclusiones en la selección de los textos, y proporcionar una visión más completa y crítica de la historia y la literatura colombiana. Además, la inclusión de modelos de analítica de datos en el estudio literario no solo proporciona una nueva forma de entender los textos, sino que también permite una mayor accesibilidad y difusión de la investigación en esta materia.

5. Objetivos del proyecto

5.1. Objetivo general

Identificar los sesgos ideológicos en colecciones literarias nacionales entre 1942 y 1958, empleando algoritmos de NLP y modelos de clasificación.

5.2. Objetivos específicos

5.2.1. Encontrar, a través de NLP, los temas más relevantes que permitan identificar el sesgo ideológico en colecciones literarias entre 1942 y 1958.
5.2.2. Transformar el contenido de los libros (texto) a vectores numéricos, utilizando técnicas de NLP.
5.2.3. Entrenar un algoritmo de clasificación, que permita identificar la ideología del libro, basado en la colección a la que pertenece.
5.2.4. Predecir, con el mayor nivel de exactitud y a partir de vectores numéricos, a qué ideología pertenece el libro de la colección analizado**.**
5.2.5 Evaluar y escoger el modelo con el mejor rendimiento en la labor de clasificación de los libros.

6. Marco teórico

6.1. Dominio del problema

La selección de los 110 títulos (40 pertenecientes a la BAC y 70 a la BPCC), se basó en un criterio de filtrado por el periodo de publicación, abarcando los siglos XIX y XX; al igual que la disponibilidad -física o digital- de los libros. Simultáneamente, como se mencionó, es imperativo explorar los posibles sesgos ideológicos presentes en los libros que conforman estas colecciones. Este análisis se llevará a cabo mediante un enfoque de aprendizaje supervisado aplicado a textos históricos, con el propósito de validar la capacidad de un algoritmo de clasificación para identificar, de manera automatizada, dichos sesgos en colecciones literarias nacionales producidas entre 1942 y 1958. Para lograr lo mencionado, se debe procesar el texto de los libros utilizando técnicas de NLP. A continuación, se detallarán los algoritmos empleados para este propósito:

6.1.1. Algoritmos de Procesamiento de Lenguaje Natural

El campo del Procesamiento de Lenguaje Natural, en particular en las áreas aplicadas de Informática e Inteligencia Artificial, tiene como propósito comprender, interpretar e incluso generar/emular el habla común a través del uso de computadoras (Bird, Klein, & Loper, Natural Language Processing With Python, 2009). Se define como lenguaje natural al medio que posee el ser humano para “comunicarse con los demás a través del sonido articulado o de otros sistemas de signos” (Real Academia Española, s.f), lo que incluye idiomas como el inglés, español, japonés, entre otros.

En la actualidad, el NLP encuentra aplicaciones en diversos ámbitos como motores de búsqueda, chatbots, asistentes virtuales, sistemas de traducción, entre otros. Su principal objetivo es comprender de manera efectiva el lenguaje humano, lo que se logra mediante técnicas de procesamiento de texto, análisis estadístico y la aplicación de modelos de aprendizaje automático. A continuación, se detallarán algunos de los algoritmos y técnicas pertenecientes al ámbito del NLP que resultan relevantes para el análisis de textos históricos dentro del contexto de este proyecto de grado:

6.1.1.1. Tokenización (Tokenization en inglés)

La tokenización es el procedimiento de fragmentar el texto en unidades más pequeñas conocidas como "tokens", los cuales serán posteriormente objeto de procesamiento (Bird, Klein, & Loper, Regular Expressions for Tokenizing Text, 2009). Estos "tokens"

pueden ser elementos individuales como palabras, puntuación, números, letras e incluso "subpalabras". Este proceso no solo implica la división del texto en diferentes partes, sino también la exclusión de elementos como signos de puntuación que no serán considerados en el análisis.

La tokenización es el primer paso en el procesamiento del lenguaje natural, pues permite que el texto sea analizado de manera más efectiva, por ende, debe realizarse antes de emplear la mayoría de los algoritmos de NLP o de normalizar cualquier texto. La teoría sobre esta técnica distingue tres categorías (Manning, Raghavan, & Schütze, 2009):

- **Token:** se trata de una instancia particular de una palabra o un símbolo dentro de un texto.
- **Término:** constituye una representación normalizada de un token, útil para eliminar variaciones ortográficas o morfológicas.
- **Tipo:** hace referencia a una instancia única de un término en un texto.

A pesar de las distinciones entre estos tres términos, a menudo se emplean de manera intercambiable en el contexto de la tokenización y el Procesamiento del Lenguaje Natural (NLP). Dentro del ámbito del NLP, existen diversas técnicas de tokenización, entre las que se incluyen la tokenización basada en reglas, la tokenización basada en estadísticas y la tokenización basada en aprendizaje profundo. Cada una de estas técnicas presenta sus propias ventajas y desventajas.

La tokenización basada en reglas implica la creación de un conjunto de reglas para segmentar el texto en diferentes tokens, siendo especialmente útil en idiomas con patrones de escritura consistentes. Por otro lado, la tokenización basada en estadísticas hace uso de modelos de lenguaje para dividir el texto en tokens.

Por último, la tokenización basada en aprendizaje profundo emplea redes neuronales profundas para fragmentar el texto en tokens, identificando previamente patrones en el texto. Esta forma de tokenización, en general, mejora la exactitud del proceso, al ser capaz de reconocer distintos tipos de palabras o incluso palabras compuestas en un texto. Para lograr este propósito, se utilizan modelos como las redes neuronales convolucionales o recurrentes.

6.1.1.2. Modelado de temas (Topic Modeling en inglés)

Otra técnica de procesamiento del lenguaje natural que ha demostrado ser útil en la evaluación de textos históricos es el modelado de temas. Esta es una forma de clasificación no supervisada, la cual permite identificar los tópicos principales de un texto y agrupar aquellos que sean similares (Silge & Robinson, 2017).

El modelado de temas utiliza algoritmos de aprendizaje automático para identificar patrones en el texto y agrupar estos en diferentes tópicos que componen el documento evaluado. Los temas se definen como agrupaciones de palabras que suelen aparecer juntas en los documentos. Sus limitaciones incluyen: dificultad para interpretar los resultados del modelado de temas, determinar la relevancia de cada uno de los temas identificados dentro del texto y la no identificación de todos los temas cuando se usan textos que abarcan varios temas o que poseen una estructura compleja.

Algunas técnicas de modelado de temas son: LDA (Latent Dirichlet Allocation), Probabilistic Latent Semantic Analysis (pLSA) y Non-negative Matrix Factorization (NMF).

6.1.1.3. LDA (Latent Dirichlet Allocation)

LDA es uno de los algoritmos más comunes para realizar modelado de temas. Es un modelo de aprendizaje automático utilizado en el procesamiento del lenguaje natural para la identificación de temas dentro de los textos. Este modelo es capaz de analizar grandes cantidades de texto y agrupar aquellos documentos que tratan sobre temas similares. Para el modelo LDA cada documento es tomado como un conjunto de temas y cada tema es representado como un conjunto de palabras (Silge & Robinson, 2017).

El modelo LDA funciona mediante la asignación de palabras a temas, en función de su probabilidad de aparición en cada tópico. Una vez que se han asignado las palabras a los temas, se pueden identificar los tópicos principales en el corpus de texto. Cabe resaltar que LDA es una técnica útil en la evaluación de textos históricos en español, ya que permite identificar patrones y tópicos en grandes cantidades de texto. Esto puede ayudar a los investigadores a identificar tendencias y eventos importantes en la historia.

Según David M. Blei, Andrew Y. Ng y Michael I. Jordan en su artículo "Latent Dirichlet Allocation" (2003), el algoritmo de LDA funciona de la siguiente manera: En primer lugar, se proporciona un conjunto de documentos de entrada al algoritmo. Cada documento se representa como una lista de palabras. A continuación, el algoritmo de LDA genera un conjunto de temas latentes que se espera expliquen la distribución de palabras en los documentos de entrada. El número de temas es un parámetro que se establece previamente a la ejecución del algoritmo. Cada tema se representa como una distribución de probabilidad sobre las palabras en el vocabulario de los documentos. Posteriormente, LDA asigna cada palabra en cada documento a uno de los temas, de acuerdo con una distribución de probabilidad sobre los temas para esa palabra. El algoritmo estima la distribución de probabilidad conjunta de los temas y las palabras para el conjunto de documentos de entrada y, por último, arroja la distribución de probabilidad condicional de los temas para cada documento de entrada. Esta distribución indica la importancia relativa de cada tema en cada documento.

Es importante resaltar que, para realizar los pasos previamente mencionados, el algoritmo de LDA utiliza un enfoque de inferencia Bayesiana para estimar las distribuciones de probabilidad. El algoritmo LDA, además, posee extensiones como Hierarchical Dirichlet Process (HDP) y Correlated Topic Model (CTM), que permiten moldear textos cuyas estructuras son más complejas.

El HDP es una extensión no paramétrica de LDA, lo que permite descubrir automáticamente el número de temas a partir de los datos (ya que esto es desconocido y debe deducirse) y, permite que el parámetro de número de temas no deba ser fijado antes de ejecutar el algoritmo (Teh, Jordan, Beal, & Blei, 2006). Adicional, esta extensión también puede modelar subtemas dentro de los temas principales, lo que lo hace útil para la exploración de temas más detallados.

Por otro lado, el CTM es una extensión que permite modelar la correlación entre los temas, lo cual es útil ya que el algoritmo de LDA asume que los temas son

independientes entre sí. Lo anterior se logra mediante la introducción de un término de covarianza en el modelo de LDA. El término de covarianza permite modelar la correlación entre las distribuciones de probabilidad de las palabras en diferentes temas. Como resultado, CTM puede capturar mejor la estructura latente en los datos de texto y es particularmente útil para modelar relaciones semánticas entre los temas (Blei & Lafferty, A Correlated Topic Model of Science, 2007).

6.1.1.4. TF-IDF (Term Frequency-Inverse Document Frequency)

El *Term Frequency-Inverse Document Frequency*, es una técnica estadística ampliamente utilizada en el procesamiento de lenguaje natural para representar documentos y palabras en un espacio vectorial. Su objetivo principal es convertir datos textuales en una forma numérica, permitiendo que los documentos se comparen entre sí en función de su contenido. La metodología TF-IDF pondera las palabras no solo en función de su frecuencia en un documento particular (Term Frequency, TF), sino también en relación con su presencia en toda una colección de documentos (Inverse Document Frequency, IDF) (Ramos, 2003). De esta forma, las palabras que son comunes en un documento, pero raras en la colección total de documentos reciben una ponderación más alta. El componente IDF tiene como objetivo reducir la importancia de palabras que aparecen frecuentemente en muchos documentos, como "el" o "y", que no aportan información significativa sobre el contenido de un texto.

6.1.1.5. Non-negative Matrix Factorization (NMF)

La Factorización de Matrices No Negativas (Non-negative Matrix Factorization o NMF) es una técnica utilizada en el procesamiento del lenguaje natural para la descomposición de una matriz no negativa en dos matrices de menor rango. Esta técnica es útil para la extracción de características y la reducción de la dimensionalidad de los datos en tareas de análisis de textos (Lee & Seung, 1999).

La NMF encuentra aplicaciones en diversas áreas del procesamiento del lenguaje natural, como la extracción de tópicos en documentos, la recuperación de información y la recomendación de contenido, ya que es capaz de identificar patrones y temas latentes en grandes conjuntos de datos textuales. Cuando se aplica a matrices de término-documento (por ejemplo, obtenidas mediante TF-IDF), NMF puede descubrir tópicos o temas representativos dentro del conjunto de documentos. Una ventaja significativa de NMF es que los tópicos y las asignaciones de tópicos a documentos resultantes son no negativos y, por lo tanto, fáciles de interpretar en términos de la contribución absoluta de cada término al tópico (Lee & Seung, 1999)

6.1.1.5. Análisis de sentimiento (Sentiment Analysis en inglés)

El análisis de sentimiento es una técnica de NLP que “analiza las opiniones, sentimientos, evaluaciones, valoraciones, actitudes y emociones” expresadas en un texto (Liu, 2012). Esta técnica se utiliza para analizar grandes cantidades de texto y clasificar al mismo dentro de un sentimiento neutral, negativo o positivo.

Existen múltiples técnicas de análisis de sentimiento, entre las cuales están: análisis basado en reglas, análisis basado en *Machine Learning*, y análisis de emociones. El primero utiliza un conjunto de reglas predefinidas para determinar la polaridad de un

texto. Las reglas pueden ser creadas a partir de un diccionario de palabras positivas y negativas (*bag of words*), o a partir de patrones gramaticales que indican una polaridad específica.

El análisis basado en *Machine Learning* utiliza algoritmos de procesamiento de texto (tokenización, stemming, lematización, entre otros) para preprocesar los datos, los cuales posteriormente se usarán para entrenar modelos de clasificación, teniendo en cuenta que los textos seleccionados deben tener una etiqueta para poder cumplir con la clasificación. Lo expuesto anteriormente es un problema de aprendizaje supervisado.

El análisis de emociones se enfoca en captar la emoción del texto, más que su polaridad, intentando identificar emociones como la felicidad, la tristeza, la ira o el miedo, entre otros.

6.1.1.6. BERT (Bidirectional Encoder Representations from Transformers)

BERT (Bidirectional Encoder Representations from Transformers) es un modelo de lenguaje desarrollado por Google en el año 2018, el cual utiliza transformadores, una arquitectura de red neuronal, para comprender el contexto de palabras en un texto a través de una representación bidireccional (Devlin et al., 2018). A diferencia de otros modelos que leen el texto secuencialmente (de izquierda a derecha o de derecha a izquierda), BERT analiza las palabras en relación con todas las otras palabras en una oración, en lugar de una a una en orden. Se entrenó en BookCorpus y Wikipedia, permitiéndole lograr resultados excelentes en 11 tareas de procesamiento de lenguaje natural.

Una de las características distintivas de BERT, y a su vez una de sus mayores ventajas, es su capacidad para preprocesar el texto de manera interna. En lugar de depender de representaciones predefinidas o de la necesidad de que se preprocese el texto previamente (tokenización, lematización o conversión a minúsculas), BERT utiliza un tokenizador WordPiece que descompone una palabra en sub-palabras y tokens conocidos (Devlin et al., 2018). Por ejemplo, una palabra desconocida se podría fragmentar en subcomponentes que el modelo reconoce. Este enfoque le permite manejar una amplia gama de tareas de NLP sin requerir adaptaciones específicas en el preprocesamiento. Esta característica es útil para que el modelo abarque idiomas y contextos variados, y minimiza la necesidad de intervenciones manuales; además, al él mismo preprocesar el texto, incrementa el rendimiento del modelo.

6.1.1.7. roBERTa (Robustly Optimized BERT Pretraining Approach)

roBERTa (Robustly Optimized BERT Pretraining Approach) es una variante optimizada de BERT desarrollada por Facebook AI en 2019. roBERTa modifica BERT al entrenar el modelo con más datos, durante más tiempo y con una tasa de aprendizaje más grande (Liu et al., 2019). Además, roBERTa rediseña el preentrenamiento de BERT al omitir la tarea de predicción de la siguiente oración y entrenar con secuencias más largas.

Por otra parte, roBERTa utiliza un tokenizador basado en bytes (Byte-Pair Encoding, BPE) en lugar del tokenizador WordPiece utilizado en BERT. Esta estrategia de tokenización, que es la misma usada por GPT-2, divide las palabras en subword units, que pueden ser tan cortas como un carácter o tan largas como una palabra (Liu et al.,

2019). Al igual que BERT, roBERTa es capaz de manejar muchos aspectos del preprocesamiento internamente, gracias a este tokenizador, permitiendo roBERTa supere a BERT en una variedad de tareas de procesamiento de lenguaje natural y que el modelo sea robusto ante una variedad de entradas, incluidas aquellas con palabras fuera del vocabulario.

6.1.2. Modelos de clasificación

Los algoritmos de clasificación son una técnica de aprendizaje automático que se emplea en NLP para clasificar textos en diferentes categorías. El objetivo de estos radica en predecir la clase a la que pertenece la data que se está evaluando. Es crucial señalar que, para aplicar estos algoritmos, los datos deben estar etiquetados previamente antes del análisis. Este requisito sitúa su utilización dentro del contexto del aprendizaje supervisado. A continuación, se proveerá una breve explicación del modelo de Support Vector Classification, pues obtuvo el mejor resultado en la tarea para el dataset analizado.

6.1.2.1. Support Vector Classification (SVC)

El Support Vector Classification (SVC) es un algoritmo de clasificación que está relacionado con el modelo de Support Vector Machines (SVM), considerablemente empleado en procesos de aprendizaje automático y, por ende, en NLP. Este algoritmo tiene como objetivo encontrar el hiperplano óptimo que mejor divide un conjunto de datos en las diferentes clases o categorías (Awad & Khanna, Support Vector Machines for Classification, 2015).

Así pues, en el contexto de NLP, el SVC se emplea en aras de categorizar textos en diferentes clases, basándose en la representación de los datos en un espacio de características adecuadamente seleccionado. A diferencia de otros métodos, el SVC tiene la ventaja de encontrar un límite de decisión que maximiza la distancia entre las clases, lo que puede resultar en una clasificación más correcta y generalizable. Asimismo, la elección del kernel es un aspecto crucial en el uso del SVC en NLP. Diferentes tipos de kernel (lineal, polinómico, radial, entre otros) pueden ser aplicados para adaptarse a la naturaleza de los datos y a la complejidad del problema. Esto brinda una gran flexibilidad al algoritmo, permitiendo su aplicación en una amplia gama de tareas de clasificación de texto.

Es importante mencionar que, al igual que otros algoritmos de clasificación, el uso del SVC requiere que los datos estén previamente etiquetados antes del análisis, lo que implica su aplicación en el marco del aprendizaje supervisado.

7. Estado del Arte

Pese a que existen múltiples fuentes sobre NLP y su uso en diferentes tipos de texto, incluidos los históricos, la cantidad de trabajos o investigaciones que emplean estos algoritmos para identificar sesgos ideológicos en textos históricos son muy pocos. Así pues, la muestra se reduce aún más si se decide segmentar geográficamente, ya que los estudios para Latinoamérica y específicamente para Colombia son escasos.

A continuación, se presentará el resumen de algunas investigaciones y artículos que se relacionan con el problema de investigación expuesto:

7.2. Trabajos seleccionados

7.2.1. Social Media Rumor Refuter Feature Analysis and Crowd Identification Based on XGBoost and NLP (Li, Zhang, Wang, & Wang, 2020)

Este artículo utiliza la técnica de procesamiento de lenguaje natural BaiduNLP para convertir los datos de texto en data numérica, que posteriormente pudiese ser procesada. Utilizan este algoritmo para leer los microblogs que el usuario decide compartir o no en sus redes sociales, para obtener el sentimiento del usuario y sus intereses. Luego, esos resultados se combinaron con otras características del usuario, como el género o la edad para clasificar a los lectores en dos categorías: aquellos que eran “refutadores” de noticias falsas o “rumores” y los “sofocadores”, aquellos que compartían dichos rumores o noticias falsas.

7.2.2. Sentiment Analysis of News Articles in Spanish using Predicate Features (Tamayo, Londoño, Burgos, & Quiroz, 2019)

Este artículo utiliza las técnicas de procesamiento de lenguaje natural SentiWordNet y ML-Senticon, para realizar análisis de sentimiento sobre diferentes partes de diferentes artículos de periódicos colombianos que estaban relacionados con la gestión de finanzas públicas. Posteriormente, utilizan un algoritmo de Support Vector Machines (SVM) para clasificar los textos como positivos (si la opinión de quien escribió la noticia está a favor sobre el tema de la misma noticia) o negativos (si el escritor del artículo no está de acuerdo con el tema de la misma noticia.

7.2.3. Content Analysis of Textbooks via Natural Language Processing: Findings on Gender, Race, and Ethnicity in Texas U.S. History Textbooks (Lucy, Demszky, Dorottya, & Jurafsky, 2020)

Este artículo aplica 4 técnicas de procesamiento de lenguaje natural a 15 libros de historia estadounidense que comúnmente se utilizan en el Estado de Texas a nivel educativo. Las cuatro técnicas son: *lexicons*, *word embeddings, topic models (LDA)* y *named entity recognition.* El propósito de este artículo es estudiar la representación de grupos históricamente marginados. Este estudio se diferencia de los otros dos escogidos en que este no busca clasificar los textos bajo distintas etiquetas, por lo cual no utiliza algoritmos de clasificación; no obstante, posee un enfoque histórico y diferencial por poblaciones marginadas.

7.2.4. Inference of Media Bias and Content Quality Using Natural-Language Processing (Chao, Molitor, Needell, & Porter, 2022)

Este artículo aplica un modelo de redes neuronales recurrentes de tipo *long short-term memory* (LSTM) para clasificar la ideología de los tuits de los medios de comunicación entre ideología de derecha o de izquierda. Dichos tuits corresponden a las fechas entre agosto de 2016 y mayo de 2020 y cuentan con las siguientes etiquetas dentro de su dataset: izquierda más extrema, izquierda hiperpartidista, izquierda sesgada, neutral, derecha sesgada, derecha hiperpartidista y derecha más

extrema. Para poder utilizar el modelo de redes neuronales, los tuits fueron procesados utilizando las técnicas de procesamiento de lenguaje natural de *bag of words* y tokenización.

7.2.5. Improving Linguistic Bias Detection in Wikipedia using Cross-Domain Adaptive Pre-Training (Madanagopal & Caverlee, 2022)

Este artículo detalla cómo detectar sesgos lingüísticos en Wikipedia. Para esto, explora el potencial del preentrenamiento interdominio en aras de aprender características de sesgo desde diversas fuentes, incluyendo artículos de noticias y declaraciones ideológicas de figuras políticas. El artículo logra evaluar la efectividad de la detección de prejuicios mediante el preentrenamiento interdisciplinario de modelos de transformación profunda. Los resultados revelan que el clasificador de sesgos entre dominios, utilizando el modelo RoBERTa preentrenado de forma continua, logra una correctitud del 89% con una puntuación F1 del 87%.

7.2.6. Retweet-BERT: Political Leaning Detection Using Language Features and Information Diffusion on Social Networks

Este artículo logra estimar las inclinaciones políticas de los usuarios de twitter, a través de las descripciones de los perfiles de los usuarios (hashtags, biografías, etc) y los medios de comunicación mencionados en sus cuentas (a través de retweets o respuestas). Ahora bien, en el artículo se detalla el uso de la técnica de BERT, Word Embbeding (Word2Vec, GloVe), Transformers, entre otros, a la par que emplea un modelo de regresión logística como técnica de clasificación de los usuarios en el espactro ideológico de izquierda o derecha. El modelo propuesto llamado Retweet-BERTdemuestra un rendimiento competitivo, alcanzando un 96%-97% en la métrica F1 en dos conjuntos de datos recientes de Twitter, uno relacionado con COVID-19 y otro centrado en las elecciones presidenciales de Estados Unidos de 2020.

Tabla No. 1 Resumen de los criterios de comparación entre los artículos seleccionados y el proyecto de grado.

	7.2.1. Social Media NLP and XGBoost	**7.2.2. Sentiment Analysis of News Articles in Spanish**	**7.2.3. Content Analysis of Texas US History Textbooks via NLP**	**7.2.4. Inference of Media Bias and Content Quality Using NLP**	**7.2.5. Improving Linguistic Bias Detection in Wikipedia using Cross-Domain Adaptive Pre-Training**	**7.2.6. Retweet-BERT: Political Leaning Detection Using Language Features andInformation Diffusion on Social Networks**	**Proyecto de grado propio**
Fecha de publicación	2020	2019	2020	2022	2022	2023	2023
País	China	Colombia	EE. UU	EE. UU	EE. UU	EE. UU	Colombia
Idioma	Chino	Español	Inglés	Inglés	Inglés	Inglés	Español
Técnica de NLP usada	Sentiment Analysis (Baidu NLP)	Sentiment Analysis (SentiWordNet y ML-Senticon)	Named Entity Recognition. Word Embbeding, LDA, lexicons	Bag of words y tokenización	RoBERTa model	BERT, Word Embbeding (Word2Vec, GloVe), Transformers	RoBERTa

Algoritmo de clasificación usado	XGBoost	SVM	No utilizan	RNN de tipo LSTM	Técnica cross-domain bias classifier	Modelo de regresión logística	SVC
Tipo de texto (género)	Microblogs, tweets	Partes de artículos de periódicos colombianos.	Libros históricos	Tuits de medios de comunicación	Artículos de internet (Wikipedia)	Descripciones de perfiles en twitter	Textos históricos
¿Identifica sesgo ideológico?	No	No	No	Sí	Sí	Sí	Sí

Fuente: Elaboración propia

8. Marco metodológico

La metodología que se empleará será SCRUM, la cual se define como un enfoque ágil para la gestión y desarrollo de proyectos de software y otros proyectos de procesamiento de aprendizaje supervisado y no supervisado. Uno de los puntos a favor de la metodología SCRUM es su flexibilidad para adaptarse a diferentes tipos de proyectos y equipos (Estrada Velasco, Núñez Villacis, Saltos Chávez, & Cunuhay Cuchipe, 2021):

> Scrum da prioridad a los individuos y las interacciones sobre los procesos y las tareas; es decir, que gran parte del éxito del proyecto se fundamenta en la forma de cómo el equipo se organiza para trabajar, poniendo énfasis en la cohesión del equipo, ya que el triunfo no es individual, sino de la colaboración de todo el equipo. Scrum Alliance (2012) sostiene que Scrum, utiliza un enfoque incremental fundamentado en la teoría de control empírico de procesos, que está basada en transparencia, inspección y adaptación. La transparencia garantiza la visibilidad en el proceso de las amenazas que pueden afectar el resultado. La inspección ayuda a detectar variaciones indeseables en el proceso; mientras que la adaptación permite realizar los ajustes pertinentes para minimizar el impacto de las mismas.

SCRUM se divide en ciclos de trabajo llamados "sprints". Un sprint es un período de tiempo fijo en el que se realiza un trabajo específico. La duración típica de un sprint es de 2 a 4 semanas. Durante cada sprint, el equipo trabaja en un conjunto de elementos del producto que se seleccionan del backlog del producto. El backlog del producto es una lista ordenada de elementos que describen el trabajo pendiente.
SCRUM se basa en roles claros y definidos, y cada uno de ellos tiene responsabilidades específicas, las cuales se describirán a continuación:

- Product Owner: Es responsable de definir y priorizar los elementos del backlog del producto, asegurándose de que el equipo esté trabajando en las características más importantes.
- Scrum Máster: Es el facilitador del equipo y ayuda a asegurarse de que el equipo esté siguiendo las reglas y prácticas de SCRUM. También ayuda a resolver problemas y a mantener el enfoque en los objetivos del sprint.
- Equipo de Desarrollo: Es el grupo de personas responsables de entregar los elementos del producto al final de cada sprint.

El ciclo de SCRUM incluye los siguientes pasos:

- Planificación del Sprint: El Product Owner trabaja con el equipo de desarrollo para seleccionar los elementos del producto que se trabajarán en el sprint y se establece el objetivo del sprint.

- Sprint: El equipo de desarrollo trabaja en los elementos del producto durante el sprint.

- Revisión del Sprint: Al final del sprint, el equipo de desarrollo presenta el trabajo completado al Product Owner y a otras partes interesadas. El objetivo de la revisión es obtener comentarios y hacer ajustes al backlog del producto.

- Retrospectiva del Sprint: Después de la revisión, el equipo de desarrollo celebra una retrospectiva para evaluar lo que funcionó bien y lo que no funcionó tan bien durante el sprint. El objetivo de la retrospectiva es mejorar continuamente el proceso.

9. Metodología

Con el propósito de cumplir con los objetivos planteados en el trabajo de investigación, se propusieron los siguientes pasos:

9.1. Elección de los libros.
9.2. Obtención de los datos.
9.3. Extracción del texto que contienen los libros de cada colección
9.4. Limpieza y perfeccionamiento de los datos (texto de los libros).
9.5. Exploración del contenido de los libros utilizando diferentes algoritmos de NLP.
9.6. Transformación del texto a vectores numéricos.
9.7. Entrenamiento de modelos de clasificación.
9.8. Validación.
9.9. Elección del modelo.

A continuación, se detallará el proceso ejecutado en cada uno de los pasos.

9.1. Elección de los libros

La colección Biblioteca de Autores Colombianos (BAC) posee 113 libros y la colección Biblioteca Popular de Cultura Colombiana (BPCC) posee 163, para un total de libros a considerar de: 276. Pese a que es un gran número, no todos estos libros son aptos para buscar sesgo ideológico, ya que algunos se escribieron antes del siglo XIX, siglo en el cual el liberalismo y el conservadurismo empezaron a forjarse en Colombia, particularmente durante las primeras décadas posteriores a la independencia de España en 1819. Debido a lo anterior, se quedó con un total de 87 libros seleccionados para la BAC y 135 para la BPCC, escogidos por la temporalidad de su escritura y publicación.

Por último, se verificó cuáles libros estaban disponibles (digitalizados y en bibliotecas), ya que, al ser textos tan antiguos, el siguiente reto consistió en encontrarlos.

9.2. Obtención de los datos

El primer paso para obtener los libros fue buscarlos en internet, tanto descargar aquellos que estaban digitalizados como identificar cuáles no lo estaban, pero se encontraban en bibliotecas.

Es pertinente mencionar que existen libros que no están digitalizados y tampoco cuentan con presencia en las bibliotecas. Algunas de las razones fueron: libros en mantenimiento y reparación, libros en colecciones especiales a los que no puede acceder el público general y libros cuyas copias no existen en la ciudad de Cali. Al final de la recolección de los libros, se encontraron 70 libros pertenecientes a la colección liberal (BPCC) y 40 pertenecientes a la colección conservadora (BAC).

La totalidad de los libros de la BPCC que se usarán en este trabajo de investigación fueron encontrados en internet, en la Biblioteca Virtual del Banco de la República. No obstante, casi ninguno de los libros de la BAC se encontraba digitalizado, por lo cual se procedió a digitalizar los libros que se encontraban en las bibliotecas de la ciudad de Cali. De los 40 libros empleados, 3 libros se encontraron en internet, en la Biblioteca Virtual Miguel de Cervantes; los restantes 37 fueron digitalizados directamente en la Biblioteca Departamental Jorge Garcés Borrero por las autoras de este trabajo de investigación. La mayoría de estos libros pertenecían a colecciones especiales, las cuales no permitían el préstamo externo de los libros.

Este fue el paso más demandante del proceso, ya que la digitalización de cada libro requería entre 90 y 180 minutos, tiempo que dependía del número de páginas y del estado del libro, ya que, al ser tan antiguos, algunos estaban en malas condiciones y debían ser tratados con extrema delicadeza.

9.3. Extracción del texto que contienen los libros de cada colección

Para los libros que se digitalizaron manualmente, se utilizó el reconocimiento óptico de caracteres (OCR) mediante Adobe Acrobat Reader, que es el proceso por el cual se convierte una imagen de texto en un formato de texto que pueden leer las máquinas (AWS, s. f.). Este proceso fue importante para que el siguiente paso fuese exitoso: extraer el texto de los PDF y guardarlos en archivos txt.

El proceso de OCR para cada libro tomaba desde 20 hasta 30 minutos, dependiendo de la cantidad de páginas y de la calidad de las fotografías utilizadas para digitalizar el libro.

Desde este paso en adelante se utilizó el lenguaje de programación Python en el IDE (*interactive development environment*) Jupyter Notebook para el procesamiento, exploración, limpieza, transformación de los datos, entrenamiento y evaluación de modelos. La versión de Python empleada fue 3.11.4. El sistema operativo utilizado fue Windows 10 Pro, en su versión 22H2. La CPU (Central Processing Unit) empleada para el procesamiento es una AMD Ryzen 7 5700X 8-Core Processor (3.40 GHz) y la GPU (Graphics Processing Unit) utilizada es una NVIDIA GeForce RTX 2060. De este momento en adelante, los procesos se realizarán con la CPU a menos que, en algunos pasos, se especifique el uso de la GPU. Se emplearon semillas para controlar los efectos de la aleatoriedad en la

división de la base de datos, el entramiento y los resultados de los modelos. La semilla utilizada fue 30102023.

Para la extracción del texto de los archivos PDF se usó la biblioteca de Apache Tika, llamado “Tika” para Python, el cual detecta y extrae metadatos y texto de más de mil tipos de archivos diferentes (como PPT, XLS y PDF).

Para exportar el texto extraído a diferentes archivos txt, se empleó la biblioteca “os”, la cual proporciona una manera de interactuar con el sistema operativo en el que está ejecutando el código.

9.4. Limpieza y perfeccionamiento de los datos (texto de los libros)

El paso de limpieza del texto consistió en:

9.4.1. Transformar todo el texto a minúsculas con el propósito de homogeneizar el formato del texto.
9.4.2. Eliminación de signos de puntuación, exclamación e interrogación.
9.4.3. Eliminación de los números de las páginas.
9.4.4. Eliminación de la frase “Este libro fue Digitalizado por la Biblioteca virtual Luis Àngel Arango del Banco de la República, Colombia”, la cual se encontraba en todos los libros digitalizados por esta entidad.
9.4.5. Tokenización.
9.4.6. Eliminación de *stopwords*:

> Las *stopwords* son palabras comunes en un idioma que, por su frecuencia de aparición y posible carencia de significado, suelen ser ignoradas en procesos de análisis de texto y minería de datos. Estas palabras, como "y", "o", "la", "el", entre otras, generalmente no aportan significado distintivo en el contenido y, por lo tanto, se eliminan para reducir la dimensión del análisis y mejorar la eficiencia del procesamiento.

9.4.7. Lematización de los verbos y las palabras en plural:

> La lematización es un proceso lingüístico que consiste en reducir las palabras a su forma base o "lema", eliminando inflexiones o variantes morfológicas. Por ejemplo, los verbos "corriendo", "correrá" y "corrió" serían lematizados a la forma "correr", así como las palabras pluralizadas como “gatos” pasarían a ser “gato”.

La realización de este proceso de limpieza se logró usando el paquete “spaCy”, el cual es una biblioteca de NLP para Python, diseñada específicamente para producir aplicaciones de NLP de alta calidad y de uso industrial. Adicionalmente, es reconocida por su buen rendimiento en los tiempos de procesamiento de texto.

En este paso también se probó con la librería “NLTK” (Natural Language Toolkit), la cual es una biblioteca líder en Python para el procesamiento del lenguaje natural (NLP); sin embargo, se decidió proceder con “spaCy” por su velocidad de procesamiento y porque tenía un mejor rendimiento en la limpieza del texto. Se debe aclarar que “spaCy” no cuenta con una lista de *stopwords* en idioma español, por lo cual se utilizó la lista de *stopwords* que provee “NLTK”.

9.5. Exploración del contenido de los libros utilizando diferentes algoritmos de NLP

Con el objetivo de identificar y conocer los temas de los libros, para posteriormente clasificarlos por sesgo ideológico dependiendo de la colección a la que pertenecen, inicialmente se realizó análisis exploratorio de los libros de cada colección.

En las conversaciones que se mantuvieron con la candidata a doctorado Gloria Morales se indicaron 3 temas cuya presencia sería clave en ambas colecciones de libros para determinar la presencia de sesgo ideológico: a) lo bolivariano, o de Simón Bolívar; b) el papel del Estado y de la constitución y c) la religión.

Uno de los objetivos de este análisis exploratorio fue evaluar si usando algoritmos de NLP se podían identificar los temas mencionados. Pese a que se intentó utilizar la técnica de conteo de palabras, la agrupación de términos por tema reveló mejores *insights* sobre el contenido de los libros.

Para la identificación de temas se emplearon 3 técnicas de NLP orientadas al modelado de temas (*Topic modeling*), las cuales se mencionaron y explicaron previamente en el marco teórico: a) LDA (*Latent Dirichlet Allocation*), b) TF-IDF (*Term Frequency-Inverse Document Frequency*) y c) NMF (*Non-negative Matrix Factorization*).

Las técnicas de NLP anteriormente mencionadas se emplearon en todo el conjunto de libros, sin importar su etiqueta (conservadora o liberal), para evaluar si se encontraban estos temas relevantes que mencionó la experta. Tanto el LDA como el TF-IDF combinado con el NMF mostraron que existen los siguientes temas principales:

LDA:

Tabla No. 2. LDA - Principales temas y tokens

Tema 1 - 44% de los tokens		
Derecho	Colombia	Política
Ley	Reforma	Gobierno
Nacional	Legislativo	Artículo
Tema 2 - 35.6% de lo tokens		
Bolívar	General	Gobierno
Libertador	Coronel	Guerra
Ejército	Hombres	Tropas
Tema 3 - 20.4% de los tokens		
Alma	Dios	Luz
Ángel	Amor	Padre
Cielo	Señor	Vida

Fuente: elaboración propia

Se fijó el hiperparámetro del número de tópicos o temas en 3. Los resultados fueron 3 temas, el primero que cuenta con el 44% de los tokens del corpus, es decir, de la

totalidad de los documentos; el segundo cuenta con el 35.6% y el tercero con el 20.4%. Dentro de la Tabla No. 2, se encuentran los términos o tokens más importantes dentro de cada tema.

Se evidencia que las palabras más importantes del tema 1 están relacionadas con el Estado, la política, y los elementos de la constitución, como los artículos y las leyes. El tema 2 está relacionado con Simón Bolívar gracias a la presencia de palabras como Bolívar, libertador, general, guerra y ejército. Por último, el tema 3 parece que está relacionado con la religión, ya que está compuesto de palabras como alma, Dios, padre y ángel. Los resultados de los temas obtenidos por el algoritmo de LDA coinciden con los tres temas que la experta Gloria Morales señaló como los tópicos más importantes que deberían encontrarse en los libros de las colecciones.

Al visualizar los resultados de esta técnica, se ajustó la lambda (λ) a un valor de: 0.55, el cual se utiliza para explorar los términos que son "relevantes" para cada tópico. Cuando lambda está configurado en 1, los términos se clasifican únicamente por su probabilidad dentro del tema o tópico, la cual se refiere a qué tan probable es que se encuentre la palabra X dentro del tópico en cuestión. Por el contrario, cuanto más pequeño es el valor de lambda, más peso se da a los términos que son distintivos para el tópico respecto al corpus completo, que se refiere a cuán distintiva o única es una palabra para un tópico en particular en relación con su presencia en otros tópicos o en el corpus completo (Sievert & Shirley, 2014).

NMF utilizando TF-IDF:

Tabla No. 3. TF IDF - Principales temas y tokens

Tema 1		
Gobierno	Tropas	Popayán
Santa Fé	Coronel	Nueva Granada
Bolívar	España	Federación
Tema 2		
Gobierno	Política	República
País	Partido	Nacional
Presidente	Social	Constitucional
Tema 3		
Sol	Poesía	Cielo
Río	Luz	Agua
Riviera	Corazón	Indio

Fuente: elaboración propia

A diferencia del algoritmo de LDA, el NMF no brinda un resultado que diga explícitamente cuáles temas son más relevantes dependiendo del número de tokens, por lo cual el orden “tema 1”, “tema 2” y “tema 3” cumple un propósito de clasificación y no representa, de ninguna manera, un orden de importancia.

Se evidencia que las palabras más importantes del tema 1 están relacionadas con el Simón Bolívar diferentes espacios geográficos, ya que las palabras más relevantes son Bolívar, España, Nueva Granada o tropas. El tema 2 está relacionado con el papel del Estado y la constitución, atribuido a la presencia de palabras como gobierno, política, república, presidente y país. Por último, el tema 3 parece que está relacionado con la naturaleza o con la literatura, ya que está compuesto de palabras como sol, poesía, cielo, río y corazón. Dos de los 3 temas obtenidos coinciden con los temas que la experta Gloria Morales señaló como los tópicos más importantes.

Para ejecutar estos algoritmos de NLP orientados al modelado de temas, se utilizaron las siguientes librerías de Python:

- LDA (*Latent Dirichlet Allocation*): La clase “LdaModel” pertenece al módulo “models” de la biblioteca “gensim”, especializada en modelado de tópicos y procesamiento de lenguaje natural. Adicional, para la visualización de los temas, se utilizó la biblioteca pyLDAvis, la cual ofrece una interfaz visual interactiva para analizar y visualizar los tópicos resultantes de modelos de LDA.
- TF-IDF (*Term Frequency-Inverse Document Frequency*): La clase “TfidfVectorizer” es parte del módulo “feature_extraction.text” de la biblioteca “sklearn” (Scikit-learn), una de las bibliotecas esencial para el aprendizaje automático.
- NMF (*Non-negative Matrix Factorization*): La clase “NMF” hace parte del módulo “decomposition”, que también hace parte de la biblioteca “sklearn” (Scikit-learn).

9.6. Transformación del texto a vectores numéricos

El proceso de transformar el texto de los libros a vectores numéricos es vital, ya que de este paso depende que los resultados de los modelos de clasificación sean óptimos y que estos puedan diferenciar a qué colecciones pertenecen los libros.
Para lograr esta transformación, se aplicaron las siguientes alternativas: a) conteo de palabras, b) vectores TF-IDF, c) BERT y d) RoBERTa.

En el contexto de ML (*Machine Learning*) se le llama "feature" a una propiedad individual medible, característica o atributo de un fenómeno que se observa. Este concepto generalmente es utilizado para hacer referencia a una variable de entrada o una columna en el conjunto de datos.

La decisión sobre cuál de las alternativas usar se basó en: a) el número de *features* que arrojaba cada técnica, b) la complejidad del algoritmo y su capacidad de extraer información del texto; y c) su rendimiento (velocidad de procesamiento).

La opción del conteo de palabras no fue la más adecuada, ya que generaba una cantidad de *features* mayor a 500, donde cada *feature* representaba una palabra y cada fila representaba el número de veces que dicha palabra aparecía en su correspondiente libro. Lo anterior tomaba un alto tiempo de procesamiento y adicionalmente, al poseer tanta información, probablemente se dificultaría la correcta ejecución de los modelos de clasificación.

La opción de los vectores TF-IDF tenía la ventaja de que extraía menos *features* que el conteo de palabras, pero su capacidad de extraer información era muy baja, teniendo en cuenta que el objetivo era resumir cada libro en vectores numéricos que los describieran.

Por último, se intentó inicialmente con BERT y, en unas últimas pruebas, se optó por roBERTa como la técnica adecuada. Lo anterior debido a que roBERTa es una versión mejorada de BERT desarrollada por Facebook AI. Adicional a lo anterior, el modelo "roberta-large" permitía obtener un vector de 1024 dimensiones, mientras que BERT, independientemente del uso del modelo "bert-base" o "bert-large", solo producía un vector de 768 dimensiones. El incremento en el número de dimensiones que entregaba el modelo tuvo un impacto significativo en la etapa de entrenamiento de los modelos de clasificación, ya que mejoraba considerablemente las predicciones que estos realizaban, debido a que, al ser un vector más largo -como ya se afirmó anteriormente-, ofrece más información del contenido de los libros.

Otra de las ventajas de roBERTa es que es un modelo que cuenta con su propio proceso de preprocesamiento del texto, por lo cual es una buena práctica pasarle el texto crudo. Este modelo cuenta con su propio tokenizador, es sensible al contexto gracias a la gran cantidad de datos con el que fue entrenado por Facebook AI, misma razón por la cual está familiarizado con una amplia variedad de estilos y estructuras de texto. Debido a lo anterior, el preprocesamiento realizado en el paso 4: "Limpieza y perfeccionamiento de los datos (texto de los libros)" no se aplicó al texto de los libros que se ingresaron en el modelo roBERTa.

Para la ejecución de BERT se empleó la clase "BertTokenizer" y "BertModel" de la biblioteca "Transformers". De la misma forma, se usó "RobertaTokenizer" y "RobertaModel" pertenecientes a la misma biblioteca. Adicional, se importó la biblioteca "torch" (PyTorch), la cual es utilizada para la construcción y entrenamiento de modelos de redes neuronales, y es conocido por su flexibilidad y capacidad para trabajar con cálculos en GPU (Graphics Processing Unit).

Se importa la biblioteca de PyTorch debido a que la biblioteca transformers, desarrollada por Hugging Face, está construida sobre PyTorch. Los modelos como BERT, RoBERTa, y otros dentro de la biblioteca, están implementados utilizando la estructura de red neuronal de PyTorch. Por lo tanto, para cargar, procesar y entrenar estos modelos, es indispensable instalar e importar "torch".

Producto de lo anteriormente mencionado, para este paso se empleó la GPU mencionada en el paso 9.3: "Extracción del texto que contienen los libros de cada colección". Los tiempos de procesamiento se aceleraron cuando se empleó la GPU:

Tabla No. 4. Tiempos de procesamiento GPU vs CPU

Device	CPU	GPU	Mejora tiempo
BERT	2.98 minutos	2.20 minutos	26.2%
roBERTa	6.27 minutos	1.65 minutos	73.7%

Fuente: elaboración propia

La velocidad de procesamiento se incrementó en un 26.2% para BERT y en un 73.7% para roBERTa.

9.7. Entrenamiento de modelos de clasificación

Inicialmente, se dividió la base de datos de 110 libros en dos partes: una de entrenamiento y otra de validación. La base de entrenamiento consistía en el 75% de la base total, es decir, de 82 libros; por su parte, la base de datos de validación contó con el 25% de la base total, es decir, 28 libros. Para esta etapa de entrenamiento del modelo, se empleó el conjunto de entrenamiento.

El conjunto de datos posee dos etiquetas: liberal y conservador, donde liberal es 1 y conservador es 0.

Se entrenaron los siguientes modelos de clasificación:

- LDA (*Linear Discriminant Analysis*).
- QDA (*Quadratic Discriminant Analysis*).
- SVC (*Support Vector Classification*).
- Regresión logística.
- *ElasticNet Linear Regression*.
- XGBoost (*Extreme Gradient Boosting*).
- MLP Classifier (*Multi-Layer Perceptron*): Una clase de red neuronal artificial de tipo *feedforward*.
- *Random Forest*.
- Árbol de decisión.

Se empleó el método de Optimización Bayesiana para la elección de los valores de los hiperparámetros de cada modelo, en aras de mejorar los resultados de estos. La optimización bayesiana es un método de búsqueda global probabilístico basado en el teorema de Bayes para actualizar la distribución de probabilidad de los hiperparámetros a medida que se recopila más información, generalmente a través de la evaluación de una función objetivo en diferentes puntos. Este método se destaca por su capacidad de encontrar el óptimo global con un número mínimo de evaluaciones de la función objetivo (Brochu, Cora, & de Freitas, 2010). La Optimización Bayesiana emplea un modelo probabilístico para predecir los resultados en puntos no muestreados y decide de manera inteligente el siguiente punto a muestrear basándose en ciertos criterios, como el balance entre exploración de regiones desconocidas y explotación de regiones conocidas con buenos resultados.

La ejecución de la Optimización Bayesiana se llevó a cabo empleando la clase “BayesianOptimization” de la biblioteca “bayes_opt”. Adicional, se empleó la clase “JSONLogger”, perteneciente a la misma biblioteca, para registrar los resultados del ejercicio en un archivo JSON. Por último, se empleó la clase “UtilityFunction” perteneciente al módulo “utils” de la misma biblioteca, en aras de ejecutar la función de utilidad necesaria en la optimización bayesiana.

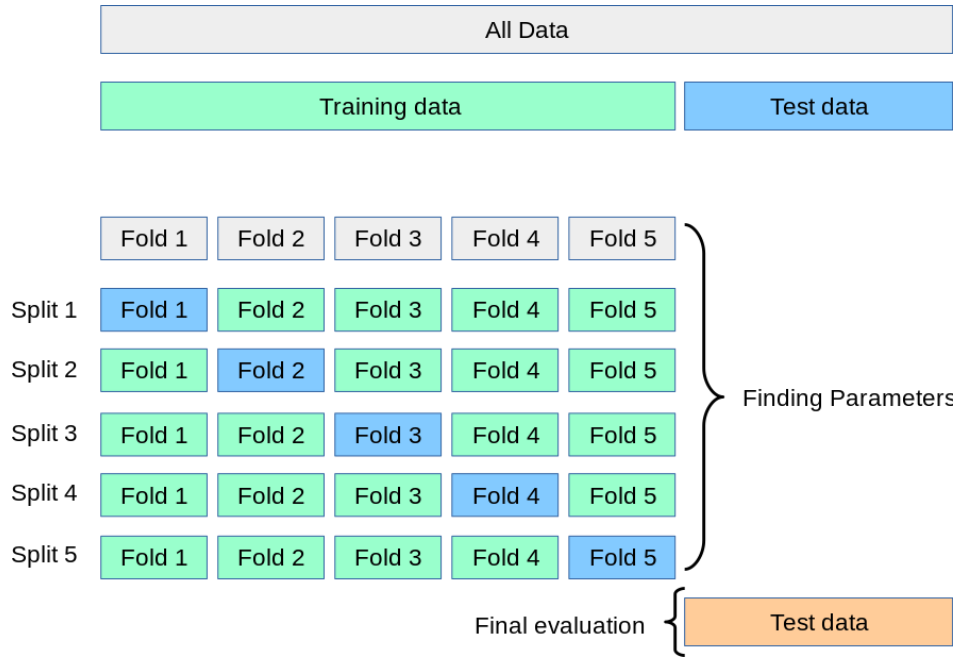
9.8. Validación

La validación del rendimiento de los modelos se realizó utilizando la técnica de K-Fold Cross Validation, más puntualmente con la variación del Leave-One-Out Cross Validation (LOOCV), y la métrica de la exactitud (accuracy).

Es menester destacar que la técnica "K-fold Cross Validation" es utilizada para evaluar la capacidad de un modelo de generalizar a datos no vistos. El objetivo principal de esta técnica es dividir el conjunto de datos original en un número K de subconjuntos (o "folds") de igual tamaño o casi igual. Luego, el proceso de validación se realiza K veces: en cada iteración, K-1 subconjuntos son utilizados para entrenar el modelo y el subconjunto restante es empleado como conjunto de prueba para validar el rendimiento del modelo (Kohavi, 1995).

El uso de K-fold Cross Validation proporciona varias ventajas por las cuales se escogió sobre el método "Hold-out" u otros métodos. Primero, al emplear diferentes subconjuntos para el entrenamiento y la validación, se maximiza el uso de los datos disponibles, lo que es valioso cuando se cuenta con un conjunto de datos limitado, como es el caso de los 110 libros de las colecciones BAC y BPCC en el presente trabajo de investigación. Además, al promediar los resultados de las K iteraciones, se minimizan las variaciones que podrían surgir si se dependiera de una única división aleatoria entre entrenamiento y prueba.

Imagen No. 1. Ilustración de procedimiento K-fold cross validation



Fuente: Tomado de https://scikit-learn.org/stable/modules/cross_validation.html

Por su parte, en la variante especializada del LOOCV, cada punto de datos del conjunto de entrenamiento se utiliza como conjunto de prueba una vez, mientras que el resto se emplea para entrenar el modelo, es decir, cada fold se constituye de un individuo y no de un grupo de individuos. Este proceso se repite K veces, donde K es igual al número total de observaciones en el conjunto de datos original. El

LOOCV proporciona una evaluación exhaustiva del rendimiento del modelo, ya que se asegura de que cada punto de datos/individuo sea probado.

En la presente investigación se optó por la técnica de LOOCV en lugar de K-Fold Cross Validation, gracias a la naturaleza del dataset (información no estructurada) y dada la relativamente limitada cantidad de datos en comparación con otros conjuntos más extensos. Así pues, al utilizar LOOCV fue posible tener una evaluación exhaustiva del rendimiento del modelo, al garantizar que cada libro se utilice como conjunto de prueba exactamente una vez. A la par, dado que el LOOCV realiza iterativamente K entrenamientos con K-1 puntos de datos, se maximiza el aprovechamiento de la información contenida en cada libro. Aunque el K-Fold Cross Validation es una técnica valiosa y ampliamente utilizada, en este contexto específico el LOOCV ofrece una evaluación más minuciosa y apropiada dada la naturaleza del conjunto de datos y el objetivo de aprovechar al máximo la información disponible para mejorar el rendimiento del modelo.

Con el objetivo de evaluar el rendimiento de los modelos en el conjunto de datos de libros, se utilizó la métrica de exactitud (accuracy). La exactitud es una medida fundamental que indica qué tan acertado estuvo el modelo considerando todas las predicciones que hizo. Entre más cerca de 1 esté el resultado de exactitud, mejor performance tendrá el modelo. En el contexto de la presente investigación la exactitud es una métrica especialmente relevante, ya que uno de los objetivos específicos apunta a determinar con exactitud la categoría o clasificación correcta de cada libro. Dada la variedad de géneros y temáticas presentes en las colecciones de libros (BAC-BPCC), es esencial contar con una métrica que permita cuantificar de manera precisa el éxito del modelo en la tarea de clasificación. Al utilizar la exactitud como métrica de evaluación, es posible capturar la capacidad del modelo para acertar en la categorización de cada libro en su respectiva clase.

En comparación con otras métricas como la sensibilidad o la especificidad, la exactitud se destaca por su capacidad para proporcionar una evaluación global del rendimiento del modelo en una variedad de escenarios. Dada la naturaleza diversa de los libros en el conjunto de datos trabajado, la exactitud es la métrica más pertinente ya que permite cuantificar de manera efectiva el grado de éxito del modelo en su tarea de clasificación de dichos individuos de las colecciones literarias.

En el contexto de un problema de clasificación binaria, donde las clases son comúnmente etiquetadas como positivas y negativas, la exactitud puede desglosarse aún más en términos de verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN). La fórmula entonces se representa como:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Verdaderos Negativos (VN): Son los libros correctamente clasificados como no pertenecientes a un género específico por el modelo.

Verdaderos Positivos (VP): Representan los libros correctamente clasificados como pertenecientes a un género específico por el modelo.

Falsos Positivos (FP): Son los libros incorrectamente clasificados como pertenecientes a un género específico cuando en realidad no lo son.

Falsos Negativos (FN): Representan los libros incorrectamente clasificados como no pertenecientes a un género específico cuando en realidad sí lo son.

El código implementado en Python para este trabajo de investigación usó en su protocolo de entrenamiento el método de validación cruzada "Leave-One-Out Cross-Validation" (LOOCV) para evaluar la capacidad predictiva de un modelo dentro de un contexto de optimización bayesiana. Se escogió este método debido a la limitada cantidad de libros que se poseen, dado que cada iteración utiliza todo el conjunto de datos menos 1, por lo cual se maximiza el uso de datos disponibles para el entrenamiento. Inicialmente, se definió un contador para llevar un registro de cuántas veces el modelo predice correctamente el dato de validación. A través de un bucle for, se recorren todos los índices del conjunto de entrenamiento, separando en cada iteración un dato para validación y utilizando el resto para entrenar el modelo. Una vez entrenado el modelo con el subconjunto de entrenamiento, se realiza una predicción sobre el dato de validación y, si la predicción es correcta, se incrementa el contador. Después de recorrer todos los datos, se calcula la tasa de exactitud (accuracy) como el cociente entre el número de predicciones correctas y el total de datos. Este valor de la exactitud es la métrica que se intenta maximizar en la función objetivo dentro de la optimización bayesiana. Este protocolo proporciona una evaluación exhaustiva del modelo, dado que cada dato se utiliza exactamente una vez como prueba, garantizando así una evaluación detallada y rigurosa dentro del proceso de optimización bayesiana.

Para la técnica *Leave-One-Out Cross-Valida*tion (LOOCV) se emplearon las clases “LeaveOneOut” y “cross_val_score”, las cuales hacen parte del módulo “model_selection” de la biblioteca “sklearn” (Scikit-learn).

9.9. Elección del modelo más eficaz

En esta etapa de evaluación del rendimiento de los modelos, se empleó el conjunto de datos de validación, que constaba de 28 libros (25% de la base total). Se seleccionó el mejor modelo, basado en el rendimiento de este en las siguientes métricas: ROC AUC, accuracy, recall, especificidad y Kappa de Cohen.

En la sección anterior ya se explicó en detalle a la métrica del accuracy y las razones de su escogencia. Ahora bien, a continuación, se brindará una breve descripción de las métricas restantes empleadas:

El Kappa de Cohen es una métrica empleada en estadísticas para la concordancia entre dos clasificadores en una tarea de clasificación, cuya etiqueta es mutuamente excluyente. Este coeficiente toma en cuenta tanto la concordancia observada como la concordancia esperada por azar (Cohen, 1960). La concordancia observada es la proporción de casos en los que los clasificadores están de acuerdo, mientras que la concordancia esperada por azar se calcula considerando la proporción de

acuerdo que se esperaría si los clasificadores estuvieran seleccionando las categorías al azar o si se asignaran al *baseline* (clase mayoritaria). El valor del Kappa de Cohen varía entre -1 y 1. Un valor de -1 indica un desacuerdo perfecto, un valor de 0 indica un acuerdo aleatorio y un valor de 1 indica un acuerdo perfecto. Este coeficiente es valioso en tareas de clasificación que poseen desbalance en las categorías de clasificación.

Por su parte, el Área bajo la Curva ROC (ROC AUC) es una métrica esencial utilizada para evaluar el rendimiento de modelos de clasificación, incluido el análisis de géneros literarios en el conjunto de datos de 110 libros. El ROC AUC permite conocer la capacidad del modelo para distinguir entre las diferentes clases de manera efectiva. La curva ROC es una representación gráfica de la sensibilidad frente a la tasa de falsos positivos a medida que se varía el umbral de decisión del modelo. Cuanto mayor sea el valor del ROC AUC (que varía entre 0 y 1), mejor será la capacidad del modelo para discriminar entre las clases. Para la presente investigación, un alto valor de ROC AUC indicaría que el modelo es capaz de realizar clasificaciones precisas y fiables de los géneros literarios de los libros.

El Recall, también conocido como “Sensibilidad” o “Tasa de Verdaderos Positivos”, es una métrica que evalúa la proporción de instancias positivas correctamente identificadas por el modelo en relación con el total de instancias positivas presentes en el conjunto de datos. En el contexto de la presente investigación sobre sesgos ideológicos en colecciones literarias, el Recall sería especialmente relevante para identificar la capacidad del modelo para detectar con exactitud todos los libros de una colección específica -por ende, con un sesgo específico-. Un valor alto de Recall indica que el modelo es altamente efectivo en identificar los libros pertenecientes a una categoría en particular.

En lo que respecta a la Especificidad, también conocida como “Tasa de Verdaderos Negativos”, da cuenta de la proporción de instancias negativas correctamente identificadas por el modelo en relación con el total de instancias negativas presentes en el conjunto de datos. En el contexto de la presente investigación, la Especificidad sería relevante en aras de evaluar la capacidad del modelo para identificar correctamente los libros que no pertenecen a una categoría específica. Un alto valor de Especificidad indica que el modelo es altamente efectivo en evitar clasificar incorrectamente los libros en una categoría errónea.

A continuación, se presenta una tabla con los resultados de los modelos de clasificación utilizados en las métricas descritas:

Tabla No. 5. Resultados de modelos empelados y métricas de calidad

Modelo	Roc_auc	Accuracy	Recall	Especificidad	Cohen_kappa
LDA	0.9305	0.8571	0.9091	0.8235	0.7098
QDA	0.6738	0.6429	0.8182	0.5294	0.3171
MLPClassifier	0.9465	0.8571	0.8182	0.8824	0.7005
Regresión logística	0.9572	0.8571	0.8182	0.8824	0.7005
SVC	0.9679	0.8929	0.8182	0.9412	0.7717
XGBOOST	0.8610	0.8214	0.8182	0.8235	0.6316
Random Forest	0.8877	0.8571	0.8182	0.8824	0.7005
Decision tree	0.7861	0.7857	0.8182	0.7647	0.5648
Elastic Net	0.9572	0.8571	0.8182	0.8824	0.7005

Fuente: elaboración propia

El modelo con mejor rendimiento fue el Support Vector Classifier (SVC), con un ROC AUC de 0.9679, accuracy de 0.8929, recall de 0.8182, especificidad de 0.9412 y un Kappa de Cohen de 0.7717.

Se debe resaltar que el modelo SVC obtuvo el mejor resultado en las métricas de: ROC AUC, accuracy, especificidad y Kappa de Cohen; el recall obtenido fue mejor en el modelo Linear Discriminant Analysis (LDA), con un 0.9091 en comparación con el 0.8182 obtenido con el SVC. Pese a lo anterior, se decide que el SVC es el mejor modelo, ya que supera el performance de los otros modelos en 4 de las 5 métricas mencionadas. Se muestra a continuación la matriz de confusión del modelo SVC:

Tabla No. 6. Matriz de confusión del modelo SVC

Matriz de confusión		**Predicción**	
		Conservador	**Liberal**
Real	**Conservador**	16	1
	Liberal	2	9

Fuente: elaboración propia

De los 28 libros del conjunto de evaluación, 17 eran liberales y 11 eran conservadores. El modelo SVC clasificó correctamente a 16 de los 17 libros liberales y a 9 de los 11 libros conservadores.

Dentro del protocolo de entrenamiento establecido se buscó optimizar los hiperparámetros del modelo SVC empleando Optimización Bayesiana. Para ello, se definió un rango de valores posibles para cada hiperparámetro:

Hiperparámetro C: Es una constante de regularización que controla el equilibrio entre maximizar el margen y minimizar la clasificación errónea en el conjunto de entrenamiento. Valores más bajos de C intentan maximizar el margen, permitiendo algunas clasificaciones erróneas, mientras que valores más altos intentan minimizar las clasificaciones erróneas, incluso a costa de elegir un margen más pequeño.

Se buscó optimizar este valor en el rango entre 0.001 y 10. El resultado de la optimización para este hiperparámetro fue de 5.24.

Hiperparámetro Gamma: Se usa solo para un modelo SVC con un kernel rbf. Determina la influencia de un único punto de entrenamiento. Valores bajos indican que cada punto tiene un rango de influencia grande, resultando en límites de decisión más suaves, mientras que valores altos implican que los puntos tienen un rango de influencia más limitado, lo que podría resultar en límites de decisión más complejos.

Se buscó optimizar este valor en el rango entre 0 y 5. El resultado de la optimización para este hiperparámetro fue de 0.032.

Hiperparámetro Kernel: Se diseñó dentro del código para elegir entre tres tipos de kernel, 'linear', ‘poly’ y 'rbf'.

Se definió en un rango de (0,1), donde los valores entre 0 y 0.333 indicaban el uso de un kernel ‘linear’; entre 0.333 y 0.666 indicaban el uso de un kernel ‘poly’ y entre 0.666 y 1 indicaban el uso de un kernel ‘rbf’. El resultado de la optimización para este hiperparámetro fue de 0.86, es decir, el uso de un kernel ‘rbf’.

Para obtener todas las métricas de evaluación se usó el módulo “metrics” de la biblioteca “sklearn” (Scikit-learn). Para la ejecución del modelo SVC se empleó la clase “SVC” del módulo “svm” de la biblioteca “sklearn” (Scikit-learn).

10. Conclusiones

A modo de conclusión del presente trabajo de investigación, se presentan los resultados más destacados del proceso:

- Fue posible identificar con alta exactitud los sesgos ideológicos (liberal-conservador) en colecciones literarias nacionales entre 1942 y 1958, empleando algoritmos de NLP -como roBERTa-, y modelos de clasificación -como SVC-.

- Para llegar a los resultados esperados, fue necesario extraer los temas más relevantes a través de los modelos de LDA y TF-IDF, por tanto, fue posible observar que los temas que los algoritmos identifican en los libros coinciden con los temas sugeridos por la experta (candidata a Doctora en Literatura), lo que apoya la idea de la presencia de sesgo ideológico en los 110 libros examinados.

- El mayor reto, en lo que respecta a procesamiento de los datos, residió en buscar y encontrar un algoritmo de NLP cuyo resultado en vectores numéricos representara el contenido de dichos textos, puesto que las fechas de elaboración de estas colecciones datan de los siglos XIX y XX, lo que dificulta que una herramienta actual comprenda la estructura, términos acuñados, entre otros. Ahora bien, gracias a grandes corporaciones como Google y Facebook IA, se puede tener acceso a algoritmos pre-entrenados como BERT y roBERTa, los cuales demuestran un buen rendimiento en esta tarea.

- El entrenamiento y rendimiento de los algoritmos de clasificación dependió de la calidad de los vectores numéricos obtenidos. Por tanto, se entrenaron 9 modelos (LDA, QDA, SVC, Regresión logística, ElasticNet Linear Regression, XGBoost, MLP Classifier, Random Forest, Árbol de decisión). Todos los modelos, a excepción del QDA, demostraron una excelente capacidad de predicción, con una exactitud (accuracy) mínima del 78,6% (árbol de decisión) y máxima de 89,3% (SVC).

- Pese a que el dataset se encuentra desbalanceado, pues se trabajó con 40 libros de la colección conservadora (BAC) y 70 de la colección liberal (BPCC), el valor del kappa de cohen para todos los modelos, a excepción de QDA, arrojó un buen performance con un mínimo de 0,56 (árbol de decisión) y un máximo de 0,77 (SVC). Lo descrito indica que el modelo escogido (SVC) clasifica un 77% mejor a los datos de la base en contraposición a la clasificación que estos podrían obtener gracias al baseline (porcentaje de la clase mayoritaria, en este caso Liberal-BPCC).

- Debido al performance de los modelos probados, es posible afirmar que, aparentemente, la frontera de decisión más adecuada para el dataset trabajado es similar a una de tipo lineal. Esto debido a que algunos de los modelos con mejores resultados en sus métricas, como LDA y ElasticNet Linear Regression, suponen una frontera de decisión de ese tipo; adicional, QDA que supone una frontera de decisión cuadrática, fue el modelo con el rendimiento más bajo en métricas.

- Las bases de datos con frontera de decisión lineal generalmente presentan mayor sesgo y menor varianza, pero se puede observar que la frontera de decisión del dataset empleado no es completamente lineal, ya que el kernel óptimo para el modelo SVC fue el de “RBF”, el cual no necesariamente es usado para casos lineales, lo que indica que la base de datos empleada puede que tenga mayor varianza que sesgo. Por lo anterior, SVC fue el mejor modelo y no los modelos de decisión completamente lineales, sin descartar que la frontera de decisión sea similar, pero no por esto completamente lineal.

- En lo que respecta a los resultados de la varianza nombrados anteriormente, es posible afirmar que la sensibilidad de dicha métrica se ve influenciada por la limitada cantidad de datos con los que se cuentan y no por los datos en sí mismos, ya que cualquier individuo (libro) que se añada al dataset puede influir en los resultados.

- La elección del modelo más eficiente no solo debe basarse en las métricas de calidad sino también debe considerar los recursos empleados. El modelo cuyo entrenamiento requirió más recursos fue la ElasticNet Linear Regression, con un tiempo de 49 minutos para 100 ejecuciones empleando la optimización bayesiana. Por su parte, el modelo de SVC, fue el modelo con el mejor rendimiento en 100 ejecuciones de optimización bayesiana, al tomar cinco minutos.

- Los buenos resultados de los modelos en las métricas de calidad podrían llegar a sugerir que se presenta sobreaprendizaje (overfitting), por lo cual se escogió el protocolo de evaluación LOOCV, para mitigar dicho inconveniente, en el cual los modelos aprenden a describir los errores aleatorios o el “ruido” del conjunto de entrenamiento. Sin embargo, se reconoce que es imperante aumentar el dataset para asegurar que esto no sea tan probable.

- La presente investigación pone de manifiesto la importancia crítica de reconocer y abordar el sesgo ideológico en colecciones literarias y en la labor de curaduría por parte de representantes estatales. La literatura no solo refleja la historia y la cultura de una sociedad, sino que también puede influir en la percepción y la comprensión valores y de la realidad. Ser consciente de la presencia de sesgo ideológico en la literatura es esencial para garantizar una representación justa y equitativa de diversas perspectivas.

- La presente investigación ha demostrado el gran aporte del trabajo interdisciplinar, al combinar la tecnología y la ciencia de datos con el ámbito de las ciencias sociales, específicamente en la literatura. La capacidad de analizar y comprender masivamente diferentes colecciones de libros a través de herramientas de Procesamiento de Lenguaje Natural y aprendizaje automático representa un avance significativo. Esto no solo permite una investigación más eficiente, sino también una comprensión más profunda de la influencia ideológica en la literatura a lo largo del tiempo. El cruce entre la tecnología y las ciencias sociales es una dirección prometedora para futuras investigaciones interdisciplinarias.

- Un aspecto crucial manifiesto por la presente investigación es la noción de que ciertos sujetos pueden haber sido subrepresentados o incluso marginados a lo largo de la historia literaria en Colombia, debido a las ideologías predominantes en determinados periodos. Esto destaca la importancia de cuestionar y revisar las colecciones literarias y las curadurías existentes en busca de sesgos ideológicos. Este conocimiento puede fomentar un enfoque más equilibrado y diverso en la representación de la historia y las voces literarias.

- Es menester apuntar que ideologías del liberalismo y el conservadurismo, aunque han experimentado cambios a lo largo del tiempo, siguen siendo fundamentales para la política y la cultura colombiana. Estas ideologías no solo influyen en la literatura, sino que también desempeñan un papel importante en la conformación de la realidad política del país. El conocimiento de estas dinámicas es esencial para comprender las tendencias ideológicas contemporáneas y sus implicaciones en la sociedad.

- Por último, los resultados de la presente investigación plantean diversas áreas de trabajo futuro. La aplicación de esta metodología a otras colecciones literarias de los siglos XIX y XX podría proporcionar una imagen más completa de la evolución de las ideologías a lo largo del tiempo en Colombia. Además, la generación de una API (Interfaz de Programación de Aplicaciones) basada en este modelo permitiría evaluar libros más actualizados, lo que resultaría en una comprensión más contemporánea de las tendencias ideológicas predominantes y su influencia en la literatura e historia del país. Además, se podría explorar la sensibilidad a variables sociopolíticas y hechos específicos más recientes, como la influencia del narcotráfico o la actividad de grupos al margen de la ley, que pueden haber dejado huella en la literatura, sin embargo, el periodo de estudio no las contempla. A la par evolución de las ideologías, particularmente en el contexto de cambios políticos y sociales, como la pluralidad de partidos y reformas estatutarias de 2003, podría ser un área interesante para futuras investigaciones.

11. Bibliografía

Awad, M., & Khanna, R. (2015). Support Vector Machines for Classification. En Efficient Learning Machines (pp. 39–66). Apress. Recuperado 8 de octubre de 2023, de https://link.springer.com/chapter/10.1007/978-1-4302-5990-9_3

AWS. (s. f.). ¿Qué es el reconocimiento óptico de caracteres (OCR)? Amazon.com. Recuperado 8 de octubre de 2023, de https://aws.amazon.com/es/what-is/ocr/

Bird, S., Klein, E., & Loper, E. (2009). En Natural Language Processing With Python (Primera ed.). O'Reilly. Recuperado el 15 de abril de 2023, de https://tjzhifei.github.io/resources/NLTK.pdf

Bird, S., Klein, E., & Loper, E. (2009). Regular Expressions for Tokenizing Text. En Python, natural Language Processing With Python (Primera ed., pág. 109). O'Reilly. Recuperado el 2023 de abril de 2023, de https://tjzhifei.github.io/resources/NLTK.pdf

Blei, D., & Lafferty, J. (junio de 2007). A Correlated Topic Model of Science. The Annals of Applied Statistics, 1(1), 17-31. Recuperado el 15 de abril de 2023, de https://projecteuclid.org/journals/annals-of-applied-statistics/volume-1/issue-1/A-correlated-topic-model-of/10.1214/07-AOAS114.full

Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 993-1022. Recuperado el 15 de abril de 2023, de https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

Brochu, E., Cora, V. M., & de Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599. Recuperado 8 de octubre de 2023, de https://browse.arxiv.org/pdf/1012.2599.pdf

Chao, Z., Molitor, D., Needell, D., & Porter, M. (1 de diciembre de 2022). Inference of Media Bias and Content Quality Using Natural-Language Processing. Recuperado el 20 de abril de 2023, de arXiv: https://arxiv.org/abs/2212.00237

Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and psychological measurement, 20(1), 37-46. Recuperado 8 de octubre de 2023, de https://journals.sagepub.com/doi/10.1177/001316446002000104

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. Recuperado el 8 de octubre del 2023, de http://arxiv.org/abs/1810.04805

Estrada Velasco, M. V., Núñez Villacis, J. A., Saltos Chávez, P. R., & Cunuhay Cuchipe, W. C. (2021). Revisión Sistemática de la Metodología Scrum para el Desarrollo de Software. Dominio de las Ciencias.

Guestrin, C., & Chen, T. (2016). XGBoost: A Scalable Tree Boosting System. KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794. Recuperado el 15 de abril de 2023, de https://dl.acm.org/doi/pdf/10.1145/2939672.2939785

Jiang, J., Ren, X., & Ferrara, E. (2023). Retweet-BERT: Political leaning detection using language features and information diffusion on social networks. Proceedings of the International AAAI Conference on Web and Social Media, 17, 459–469. Recuperado 8 de octubre de 2023, de https://doi.org/10.1609/icwsm.v17i1.22160

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755), 788-791. Recuperado 8 de octubre de 2023, de http://www.cs.columbia.edu/~blei/fogm/2020F/readings/LeeSeung1999.pdf

Li, Z., Zhang, Q., Wang, Y., & Wang, S. (2020). Social Media Rumor Refuter Feature Analysis and Crowd Identification Based on XGBoost and NLP . Applied Sciences, 4711-4726. Recuperado el 20 de abril de 2023

Liu, B. (2012). Sentiment Analysis: A Fascinating Problem. En Mining, Sentiment Analysis and Opinion (págs. 7-16). Morgan & Claypool Publishers. Recuperado el 15 de abril de 2023, de https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. Recuperado el 8 de octubre del 2023, de http://arxiv.org/abs/1907.11692

Lucy, L., Demszky, D., Dorottya, P., & Jurafsky, D. (2020). Content Analysis of Textbooks via Natural Language Processing: Findings on Gender, Race, and Ethnicity in Texas U.S. History Textbooks. AERA Open, 1-27. Recuperado el 20 de abril de 2023, de https://journals.sagepub.com/doi/epdf/10.1177/2332858420940312

Madanagopal, K., & Caverlee, J. (2022). Improving linguistic bias detection in Wikipedia using cross-domain adaptive pre-training. Companion Proceedings of the Web Conference 2022. Recuperado 8 de octubre de 2023, de https://dl.acm.org/doi/abs/10.1145/3487553.3524926

Manning, C., Raghavan, P., & Schütze, H. (2009). Tokenization. En An Introduction to Information Retrieval (págs. 22-23). Recuperado el 15 de abril de 2023, de https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf

Marín Colorado, P. A. (2017). La colección Biblioteca Popular de Cultura Colombiana (1942-1952). Ampliación del público lector y fortalecimiento del campo editorial colombianos. Información, cultura y sociedad: revista del Instituto de Investigaciones Bibliotecológicas.

Morales Osorio, G. J. (2020). Documento interno.

Ramos, J. (2003). Using TF-IDF to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning (Vol. 242, p. 133). Recuperado 8 de octubre de 2023, de https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b3bf6373ff41a115197cb5b30e57830c16130c2c

Raschka, S., & Mirjalili, V. (2019). Classifying Images with Deep Convolutional Neural Networks. En Python Machine Learning (Tercera ed., págs. 517-564). Packt Publishing. Recuperado el 15 de abril de 2023

Raschka, S., & Mirjalili, V. (2019). Modeling Sequential Data Using Recurrent Neural Networks. En Python Machine Learning (Tercera ed., págs. 567-618). Packt Publishing. Recuperado el 15 de abril de 2023

Real Academia Española. (s.f). Definición de "lenguaje". Recuperado el 15 de abril de 2023, de Diccionario de la lengua española: https://dle.rae.es/lenguaje?m=form

Silge, J., & Robinson, D. (2017). Topic Modeling. En Text Mining with R: A Tidy Approach. O'Reilly. Recuperado el 15 de abril de 2023, de https://www.tidytextmining.com/topicmodeling.html

Sievert, C., & Shirley, K. (2014). LDAvis: A method for visualizing and interpreting topics. Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces. Recuperado 8 de octubre de 2023, de https://nlp.stanford.edu/events/illvi2014/papers/sievert-illvi2014.pdf

Tamayo, A., Londoño, J., Burgos, D., & Quiroz, G. (2019). Sentiment Analysis of News Articles in Spanish using Predicate Features. Lenguaje, 235-267. Recuperado el 20 de abril de 2023, de http://www.scielo.org.co/pdf/leng/v47n2/0120-3479-leng-47-02-00235.pdf

Teh, Y. W., Jordan, M., Beal, M., & Blei, D. (diciembre de 2006). Hierarchical Dirichlet Processes. Journal of the American Statistical Association, 101(476), 1566-1581. Recuperado el 15 de abril de 2023, de

https://www.stat.berkeley.edu/~aldous/206-Exch/Papers/hierarchical_dirichlet.pdf