UNIVERSIDAD DE LA REPÚBLICA URUGUAY

FACULTAD DE INGENIERÍA UDELAR

INTRODUCCIÓN A LA CIENCIA DE DATOS

Informe Tareas 1 y 2

Modelos de clasificación de texto aplicados a discursos presidenciales de EEUU 2020

Diagnóstico de calidad de los datos, procesos de limpieza, y etapas de entrenamiento y validación de modelos de clasificación.

Ing. Isabella Buschiazzo

Ing. Lucía Gutierrez

Junio 2025

Índice

Tarea 1

Tarea 2

Tarea 1

1. Introducción

La recolección de datos es un paso crucial en la generación de conocimiento científico. Sin embargo, es importante tener presente que los datos por sí solos no tienen valor sin una gestión adecuada. La calidad de la información obtenida depende de cómo se gestionan esos datos, y esta gestión incluye diversas dimensiones, entre las que destacan el modelado, la calidad, la integración, así como la evaluación de aspectos éticos, de privacidad y de sesgo. Estas consideraciones son fundamentales para garantizar que los datos sean útiles, precisos y responsables, alineándose con las mejores prácticas en la ciencia y la investigación.

El presente trabajo se centra particularmente en el análisis de la calidad de los datos. Esta dimensión permite determinar qué tipo de información puede extraerse de un conjunto de datos y cuáles son sus limitaciones. Es común encontrar bases de datos que, si bien están disponibles, presentan desorden, incompletitud, inconsistencias o ruido. Dichos problemas pueden originarse en distintas etapas del ciclo de vida de los datos: producción, procesamiento, almacenamiento o utilización. Por lo tanto, mejorar la calidad de los datos requiere identificar la fuente de los problemas y aplicar medidas correctivas adecuadas. Sin embargo, en muchos casos se dispone únicamente de los conjuntos de datos, sin posibilidad de intervenir en su origen. El presente informe se enmarca en dicho contexto, por lo que el objetivo principal del mismo es plantear estrategias que permitan evaluar y mejorar la calidad de los datos disponibles, maximizando así su valor informativo a pesar de las limitaciones.

El análisis de calidad mencionado se realiza para un conjunto de datos que contiene un listado de discursos pronunciados por diferentes candidatos en las elecciones presidenciales de Estados Unidos del año 2020. Los datos recolectados incluyen: el nombre del candidato que pronunció el discurso, la fecha y el lugar junto con el título del discurso, su clasificación y una transcripción del mismo.

A modo de contexto, resulta interesante señalar que el año 2020 estuvo marcado por múltiples acontecimientos de gran relevancia política, económica y social en Estados Unidos. En primer lugar, en el marco de las elecciones presidenciales, hubo un cambio en el mando político del país, resultando electo como nuevo presidente Joe Biden (candidato del Partido Demócrata), y se llevó a cabo un juicio político (*impeachment*) contra el presidente cesante Donald Trump. Ese mismo año, se produjeron masivas y en ocasiones violentas protestas a lo largo del país tras el asesinato de George Floyd, un ciudadano afroamericano, a manos de un oficial de policía en la ciudad de Minneapolis. Por otra parte, se decretó la pandemia de COVID-19, generando una crisis sanitaria sin precedentes a nivel mundial.

Los datos utilizados en este trabajo incluyen la transcripción de 269 discursos (así como su clasificación, fecha y lugar de ocurrencia). A partir de este conjunto, se busca realizar una limpieza preliminar del texto, con el objetivo de extraer información relevante, así como presentar distintas visualizaciones a partir de los discursos de los cinco candidatos con mayor cantidad de intervenciones registradas (asociando esta frecuencia con su popularidad).

Los datos se cargaron en formato Data Frame dentro de un notebook de Jupyter.

2. Desarrollo

2.1. Calidad de datos

En primer lugar, se realizó una inspección visual general del Data Frame cargado con los discursos de los candidatos, con el objetivo de observar su estructura, clasificación y orden. Luego, se examinó la primera columna *speakers* del Data Frame, correspondiente a los nombres de los candidatos presidenciales que pronunciaron discursos durante el año 2020, y se identificaron irregularidades y errores.

Para evaluar la calidad de los datos en la columna *speakers*, se utilizó la función *value_counts* para visualizar los objetos (*inputs*) únicos registrados, así como la cantidad de apariciones de cada uno. De esta manera, se logró reducir los datos a analizar de 269 (cantidad de discursos totales registrados) a 72 (cantidad de nombres únicos de candidatos a presidente que pronunciaron dicho discurso), haciendo mucho más sencilla la inspección visual para detección de anomalías en los datos cargados.

En la Tabla 1 se muestra el resultado. Como se puede observar, figuran los nombres de varios candidatos tanto del Partido Demócrata como del Republicano. Como era de esperarse, los candidatos más relevantes, que aparecieron en mayor cantidad de entrevistas y contaron con mayor cobertura publicitaria y presencia en redes sociales durante la campaña, se encuentran presentes en esta lista, por lo que a priori no se detectan sesgos o preferencias por uno u otro partido político en cuanto a recolección de datos.

Sin embargo, también se detectaron valores faltantes, ingresados al Data Frame como NaN (un total de 3) y una entrada "???" haciendo también referencia a la ausencia del autor de dicho discurso.

Además, se identificaron discursos pronunciados en conjunto por más de un candidato, ya fuera estos en el marco de debates, actos de campaña compartidos o eventos que involucraron a representantes de distintos partidos. En estos casos, el discurso se cargó en el Data Frame como una entrada (*input***)** única bajo el nombre de todos los participantes (separados por comas), en lugar de cargar una entrada por cada participante, con el mismo discurso, lugar, fecha y clasificación pero bajo el nombre de cada candidato (*speaker*) separadamente. También aparecen registros con denominaciones genéricas como “Democratic Candidates” o “Multiple Speakers”, que no permiten identificar cuál o cuáles candidatos los pronunciaron y, por lo tanto, no permiten agruparlos como los demás discursos de los candidatos listados.

Esta variación en los criterios de presentación de los nombres de los candidatos constituye una inconsistencia dentro del conjunto de datos, ya que refleja una falta de uniformidad en el modo en que se presentan casos similares.

Todo lo mencionado genera cierta incertidumbre respecto a la validez de estos registros.

Tabla 1: Entradas únicas de la columna “speakers” y la cantidad de repeticiones.

Candidatos	Cantidad de discursos
Joe Biden	71
Donald Trump	53
…	…
Democratic Candidates	8
Multiple Speakers	5
Joe Biden, Kamala Harris	4
…	…
NaN	3
…	…
???	1

Antes de continuar con el análisis de las restantes columnas del Data Frame, se realizó una inspección visual de los datos para detectar posibles errores e inconsistencias y se observó que estos solamente se debían a datos faltantes y algunas inconsistencias.

Por ejemplo, se verificó que en los casos en que el discurso o debate fuera interpretado por más de un candidato, la transcripción incluyera los nombres de estos mismos y no otros, lo mismo en el caso de entrevistas de candidatos individuales.

También se verificó que la fecha que figura en el título de los diferentes discursos coincidiera con la fecha listada en la columna *date* del Data Frame; así como que el tipo de discurso coincidiera con la locación. Por ejemplo, que no aparecieran listadas cadenas de televisión para discursos de campaña presenciales o movilizaciones de votantes; solo para entrevistas o debates, que son instancias más factibles de ocurrir en el estudio de una cadena de televisión.

Al verificar lo anterior, se observó, en la columna *location*, una inconsistencia asociada a la falta de homogeneidad en los criterios de registro: en algunos casos se indicaba la ciudad donde se llevó a cabo el discurso, mientras que en otros se mencionaba una cadena de televisión.

Es verdad que un discurso llevado a cabo al aire libre o en algún recinto oficial no es lo mismo que uno realizado en el estudio de una compañía de televisión, pero se podría también incluir la localización de este estudio de grabación.

Además, se observó que en el caso de instancias que no se realizaron de forma presencial se contaba con la designación "Virtual" como dato de localización. Este último término es algo menos específico que los demás (cadena de televisión de la entrevista o lugar físico del discurso de campaña o movilización), pero designa situaciones diferentes a éstas a las que se puede dar un dato de localización más específico. Para resolver esta heterogeneidad, se propone la incorporación de una nueva columna que indique la modalidad del discurso (por ejemplo, presencial, virtual o presencial en estudio), lo cual permitiría establecer una clasificación más clara y dar datos adicionales de localización espacial de los candidatos al momento del discurso.

A su vez, esta distinción podría ser útil para identificar fácilmente las relaciones entre el tipo de discurso y el lugar en que fue emitido. Por ejemplo, es probable que los debates y entrevistas se realicen mayoritariamente en estudios de televisión, mientras que los actos de campaña tengan lugar de forma presencial en distintas ciudades.

Una vez analizados estos errores de inconsistencia, se procedió a buscar la cantidad de datos faltantes del conjunto de datos con la función *isna().sum()*. Fuera de la columna *speaker*, no se detectaron entradas erróneas sustituyendo datos faltantes como "???", por lo que esta función fue suficiente para cuantificar todos los datos no proporcionados. Los resultados de esta búsqueda se presentan en la Tabla 2. Una consideración importante es que aquellos datos faltantes presentes en discursos compartidos que se multiplicaron, se cuentan una vez por cada orador participante.

Tabla 2: Cantidad de datos faltantes por columna de datos brindados.

Speakers	Title	Text	Date	Location	Type
3	0	0	0	19	23

Con esto se observa que solamente en las columnas de *location* y *type* hay información que no se completó, por lo que no faltan datos de títulos, transcripciones o fechas de discursos.

Resulta relevante mencionar que, si bien existen datos faltantes en las columnas *location* y *type*, las filas correspondientes no fueron eliminadas, dado que contienen información importante como la transcripción del texto, la fecha y el candidato, que constituyen el foco principal de este trabajo.

2.2. Visualización de cantidad discursos

Tras una primera depuración del conjunto de datos, se seleccionaron los discursos de los 5 candidatos con mayor frecuencia de aparición, con el objetivo de facilitar el tratamiento

posterior de la información y otorgar mayor claridad y relevancia a los análisis y visualizaciones subsiguientes.

Los candidatos resultantes fueron: Joe Biden, Donald Trump, Mike Pence, Bernie Sanders y Kamala Harris.

Cabe señalar que, en los casos donde un mismo discurso figuraba bajo la autoría de más de un candidato (con sus nombres separados por comas en la columna *speaker*), dicho discurso fue asignado a ambos. Esto se realizó mediante la combinación de las funciones *str.split(',')* y *explode*. No obstante, aquellos discursos registrados bajo las etiquetas "Multiple Speakers" o "Democratic Candidates" no pudieron ser asignados a ningún candidato en particular, debido a la falta de información específica.

En la Figura 1 se presenta a modo de *scatter plot* la cantidad de discursos por candidato agrupados mensualmente. Las elecciones presidenciales en Estados Unidos se llevaron a cabo el 3 de noviembre de 2020. Septiembre fue el mes con mayor actividad discursiva, seguido por octubre, coincidiendo con el periodo más intenso de la campaña electoral.

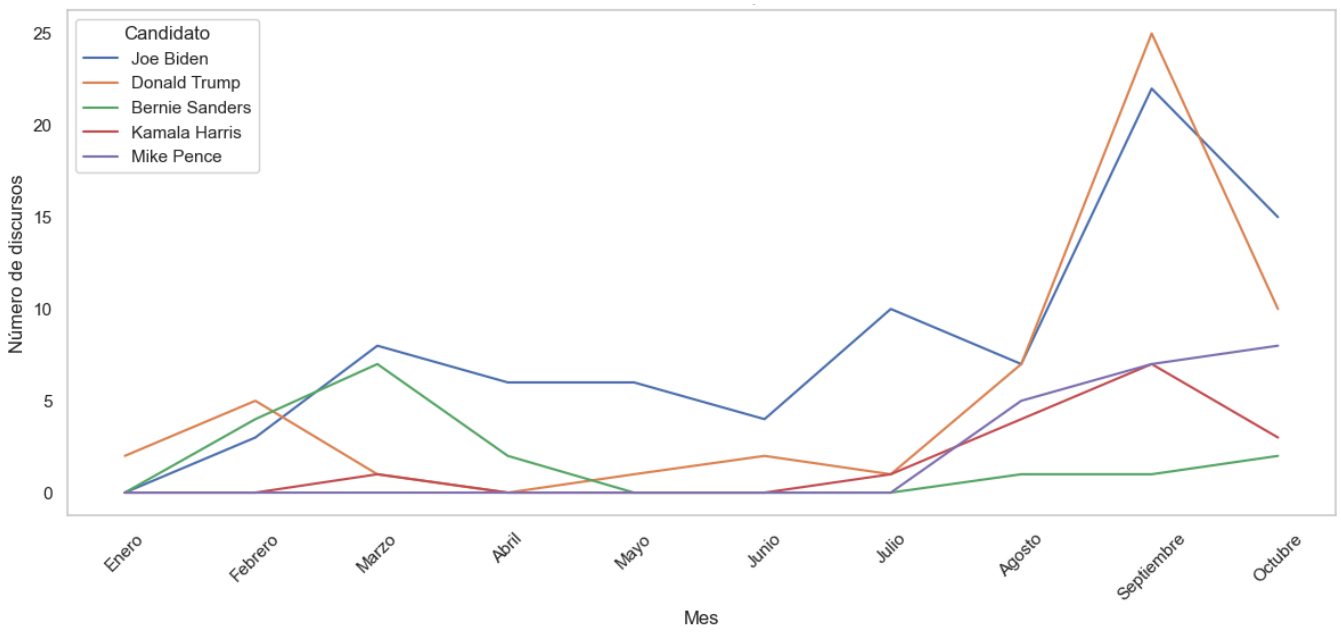

Figura 1: Cantidad de discursos pronunciados por cada candidato a presidente de los Estados Unidos durante el año 2020.

También resulta interesante visualizar individualmente la cantidad y distribución temporal de los discursos de cada candidato, superpuesto al gráfico del total de discursos de los cinco candidatos con más discursos en el período a modo de comparativa.

Esto puede hacerse para todos los (5) candidatos. A modo de ejemplo, en la Figura 2 se presenta a modo de *scatter plot* la comparación para el candidato Joe Biden, los demás gráficos se obtienen de forma análoga.

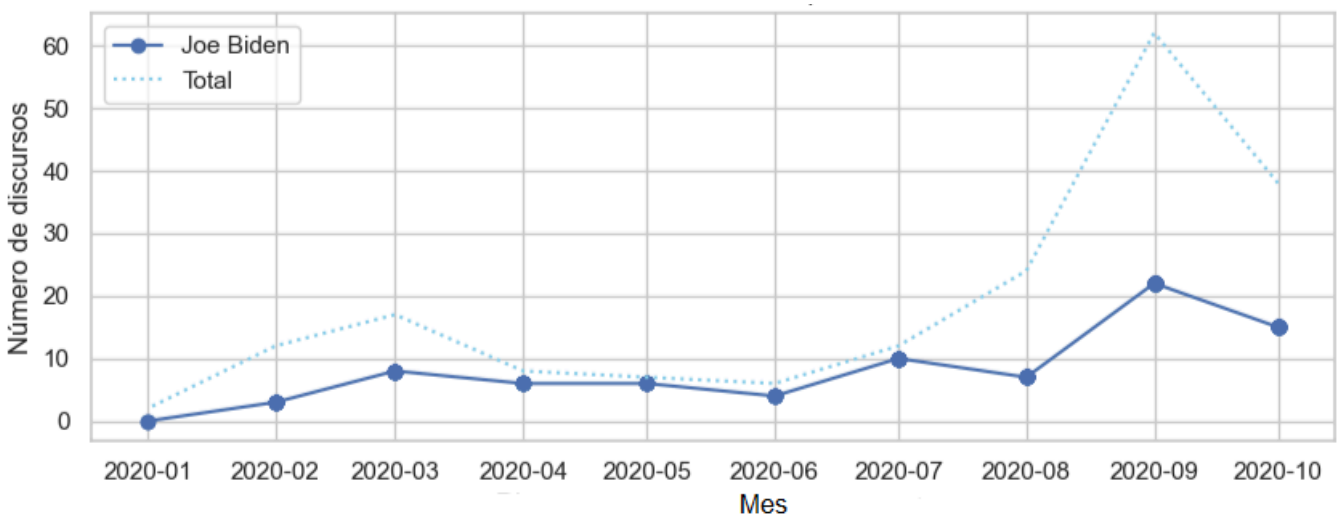

Figura 2: Cantidad de discursos pronunciados por Joe Biden comparado con el total de discursos pronunciados por los 5 candidatos con más discursos durante el año 2020.

2.3. Limpieza del texto

Una de las visualizaciones más relevantes que pueden generarse a partir de este conjunto de datos es aquella que muestra la frecuencia con la que cada candidato menciona determinadas palabras clave, o el conteo de las palabras más utilizadas por cada candidato.

Para llevar adelante este análisis, es necesario normalizar el texto, es decir, convertir todas las palabras a minúsculas y eliminar signos de puntuación como puntos y comas. Con este fin se utilizó la función personalizada *clean_text* para eliminar las primeras palabras hasta el primer \n (normalmente correspondientes a identificadores de la transcripción del discursos, y no a contenido del propio discurso en sí), convertir todo el texto a minúsculas y completar signos de puntuación faltantes. El texto "limpio" de cada discurso se almacenó en una nueva columna *CleanText* en el Data Frame original.

Además de esas modificaciones, se eliminaron algunos puntos que se detectaron por inspección visual para evitar que palabras como "you" y "you." fueran contabilizadas separadamente, utilizando la función personalizada *limpiar_puntos*. También, se hizo manualmente una lista de las contracciones más comunes y utilizadas del idioma inglés con el objetivo de implementar una función que las expandiera (por ejemplo, sustituir "didn't" por "did not"). Si bien "didn't" y "did not" son fonéticamente diferentes, a la hora de contabilizar palabras utilizadas representan el mismo significado ("did" y "not"), por lo que se consideró contabilizarlas junto con sus dos palabras constitutivas y no como una palabra propia. Para esto se utilizó la función personalizada *expand_contractions*.

Finalmente, se utilizó la función *str.split* para separar el texto limpio y homogeneizado en palabras individuales y listarlas. Esto se incluyó en una nueva columna, *WordList,* del Data Frame.

Cabe resaltar que existe otro problema, no necesariamente de calidad de datos, que se encuentra presente en la transcripción del texto de los discursos y que podría afectar al resultado de conteo de palabras. En los casos de entrevistas o debates o incluso algún discurso en que interviene más de una persona, no solo se transcribe lo pronunciado por la persona a quien se atribuye dicho discurso sino también lo pronunciado por el entrevistador, periodista, adversario o participante que le contesta.

Entonces se encuentran textos del estilo: "Donald Trump: …, Speaker 1: …, "Donald Trump": …, "Speaker 1: …". Y esto presenta dos problemas, por un lado el hecho que Donald, Trump, Speaker y 1 son contabilizadas como palabras pronunciadas cuando en realidad no lo fueron y solamente están presentes a modo de organización de la transcripción, y segundo el hecho que todo lo pronunciado por el Speaker 1 será erróneamente atribuido a Donald Trump cuando en realidad fue dicho por otra persona.

Sin embargo, para poder corregir este error se debería conocer todas y cada una de las personas (aparte de los candidatos presidenciales) que participaron de todos y cada uno de los discursos, entrevistas o debates (o incluso tener en cuenta denominaciones genéricas como *Speaker, Interviewer, Person,* etc.) para poder eliminar intervenciones no pertenecientes a los candidatos de todos los debates. Ya que si solo se consideraran algunos (por ejemplo, solo Speaker 1), se corregirían algunos discursos y otros no, generando un sesgo. La forma de considerar a todos los participantes que no son candidatos presidenciales es examinar la totalidad de los discursos uno por uno y, dada su extensión, aún realizándose manualmente existe el riesgo de cometer errores u omitir participantes.

Por lo tanto, se opta por omitir esta corrección y realizar el conteo de palabras con el texto tal cual se obtuvo luego de la anterior limpieza, asumiendo el error en el conteo de palabras como "Donald", "Trump", "Joe" o "Biden" que, además de ser verdaderamente dichas por los propios candidatos, pueden aparecer en la transcripción del texto como elementos de orden o dichos por otro participante (entrevistador, periodista o persona externa).

2.4. Conteo de palabras y visualización

A partir del texto procesado en el ítem anterior, y la columna se construyeron distintas visualizaciones. En primer lugar, se implementó la función personalizada *total_word_speaker* para visualizar la cantidad de palabras totales pronunciadas por cada candidato. Los resultados se presentan ordenados de mayor a menor en la Tabla 3. En esta se puede observar que las de Donald Trump (candidato que pronunció más palabras) difieren en un orden de magnitud con las de Bernie Sanders (candidato que pronunció menos palabras).

Este resultado coincide con la percepción que se tiene del candidato Donald Trump, persona mediática que genera polémica y que, por lo tanto, se pronuncia con mayor frecuencia.

Tabla 3: Cantidad de palabras pronunciadas por candidato.

Candidato	Total de Palabras
Donald Trump	579305
Joe Biden	468916
Mike Pence	121546
Kamala Harris	89725
Bernie Sanders	68693

A continuación, se contaron las palabras más frecuentemente mencionadas por cada candidato, utilizando la función personalizada *rank_words_speaker*. En la Figura 3 se presentan las 10 palabras más mencionadas en general (ordenadas de mayor a menor) y la cantidad de veces que fue mencionada por cada candidato, a modo de gráfico de barras.

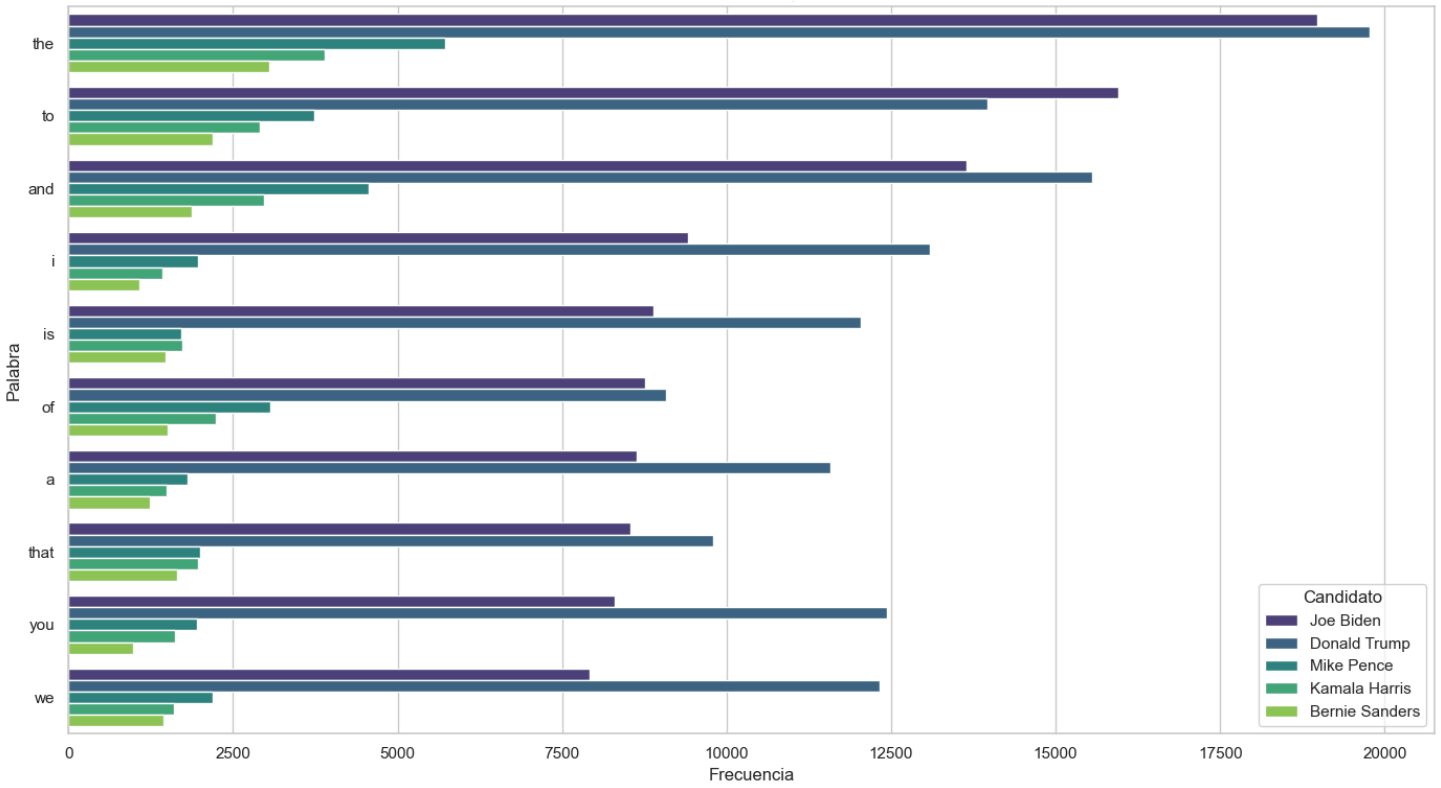

Figura 3: Palabras más frecuentes en el conjunto total de discursos, diferenciadas por candidato.

Cabe resaltar que en esta visualización se cuenta las 10 palabras más mencionadas entre el total de los candidatos para luego contar la cantidad de menciones de cada uno. Pero no necesariamente las 10 palabras más mencionadas globalmente se corresponden con las 10 palabras más mencionadas por cada candidato particular.

La función personalizada *rank_words_speaker* permitió contar las palabras más mencionadas en todos los discursos y ordenarlas según la frecuencia con que fueron utilizadas por cada candidato. A modo de ejemplo se presentan en la Tabla 4 las de Kamala Harris. Un procedimiento análogo se realizó para los demás candidatos, pero se incluye en el informe únicamente este ejemplo con el objetivo de resaltar el hecho de que las palabras más pronunciadas globalmente no tienen porqué coincidir con las más pronunciadas por cada candidato. Esta información se puede observar en la Figura 4, donde se cuentan las palabras más mencionadas por el candidato, en particular. Como se mencionó anteriormente, esta gráfica difiere de la anterior, ya que muestra únicamente las diez palabras más mencionadas por cada candidato. En el caso de la palabra “it”, solo aparece el dato correspondiente a Donald Trump, ya que esta palabra figura entre sus diez más utilizadas. Esto no implica que los demás candidatos no la hayan mencionado, sino que no se encuentra entre sus términos más frecuentes.

Tabla 4: Palabras más utilizadas por Kamala Harris en sus discursos.

Palabra	Cantidad de menciones
the	3899
and	2978
to	2911
of	2238
that	1967
is	1729
you	1618
we	1606
a	1488
i	1429

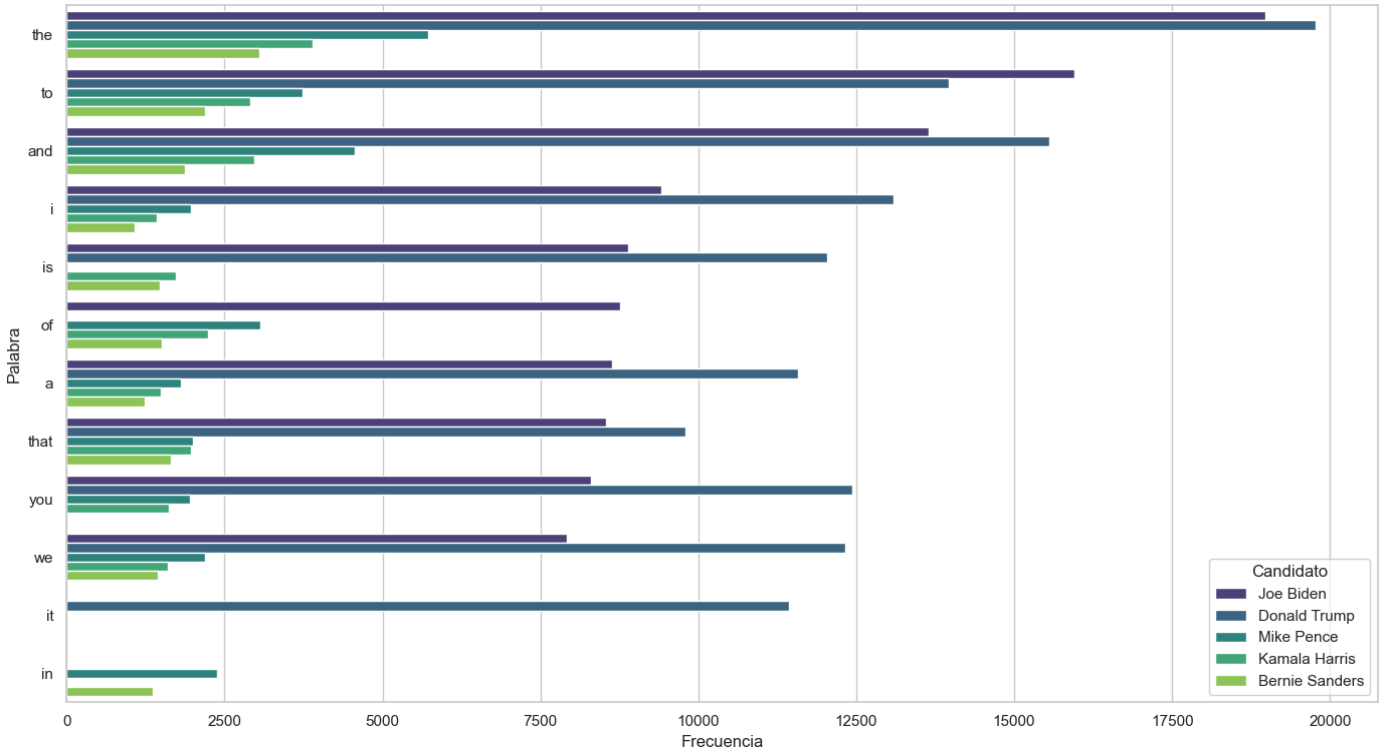

Figura 4: Palabras más frecuentes en los discursos de cada candidato.

Es interesante mencionar que uno de los problemas de visualización que surge al contar palabras de esta manera es que al contar las palabras de todos los discursos en general, se está comparando magnitudes que no son necesariamente equivalentes. Esto se debe a que algunos candidatos mencionaron más una palabra en particular simplemente porque dieron más discursos y pronunciaron más palabras en total, y no necesariamente porque la utilicen con mayor frecuencia que otros candidatos que la pronunciaron menos.

Para corregir este problema, se pueden normalizar los datos. Dado que esta gráfica simplemente incluye conectores funcionales, verbos o pronombres sin valor semántico, no se realizó esta corrección, pero sí es una consideración que se tuvo en cuenta en gráficos que se presentarán más adelante en este informe que cuentan palabras de mayor interés.

También se incluye en la Figura 5 una visualización en forma de gráfica de barras de las 10 palabras más utilizadas en este caso por Mike Pence. Nuevamente se observa que no necesariamente coinciden con las 10 más mencionadas globalmente por Kamala Harris.

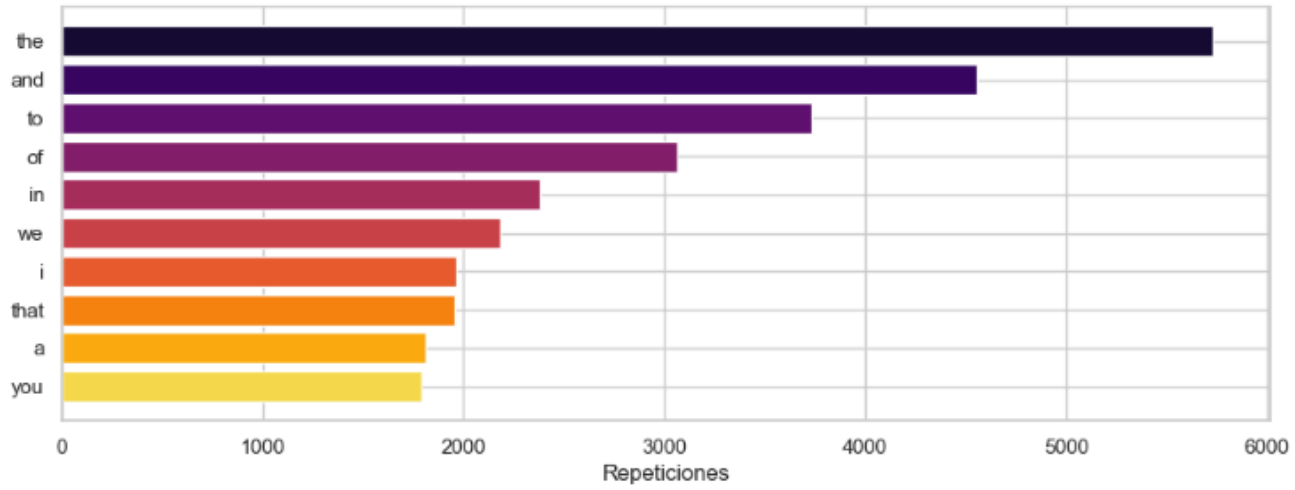

Figura 5: Palabras más utilizadas por Mike Pence

No obstante, como se mencionaba respecto a la Figura 3 y 4 (y también es aplicable para la Figura 5), este enfoque inicial de conteo de palabras sin ningún criterio resultó limitado, dado que las palabras más frecuentes son generalmente términos funcionales como “a”, “the” o “to”, comunes a todos los discursos y sin contenido semántico relevante.

El análisis de frecuencia puede resultar de mayor utilidad o interés al enfocarse en términos sustantivos vinculados a asuntos de interés público para los Estados Unidos, como la salud, el desempleo o la inmigración. Para evaluar esto, se estudió la cantidad de veces que cada candidato mencionó determinados términos clave asociados a estos temas. Para esto se definió una lista de palabras clave seleccionadas manualmente de acuerdo a consideraciones personales de lo que debería formar parte de la discusión política en el marco de una campaña electoral: "democracy", "covid", "economy", "education", "security", "unemployment", "rights", "healthcare", "racial", "immigration", "wall", "climate change", "poverty", "war". Se implementó la función personalizada *count_ keywords_per_speaker* para realizar el conteo de las mismas entre los discursos.

Finalmente, en este caso sí se normalizaron los datos para comparar frecuencia de menciones por cada candidato y no solo cantidad de menciones totales, para evaluar a qué asuntos de interés público cada candidato da más importancia y dedica más tiempo en sus discursos. En la Figura 6 se grafica la cantidad de repeticiones de estas palabras clave seleccionadas por cada 1000 palabras pronunciadas por cada candidato.

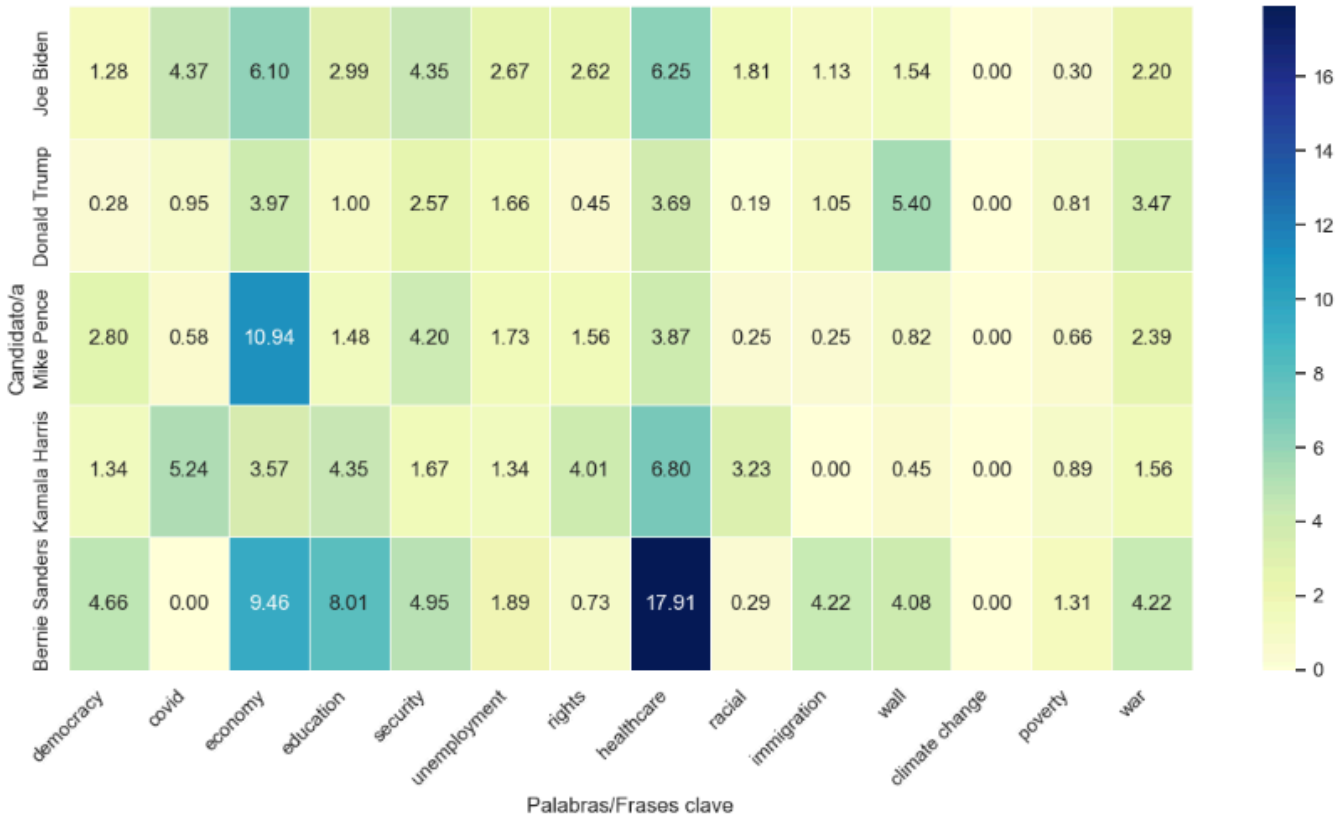

Figura 6: Conteo de palabras clave mencionadas por candidato

Es interesante observar la relevancia que el candidato Bernie Sanders otorga al asunto de la salud pública y el protagonismo que le da en sus discursos, lo cual es coherente con su orientación Demócrata. Sanders es el presidente que presenta un protagonista más marcado (dentro de los términos considerados, por supuesto), seguido por Mike Pence y su enfoque en la economía, también coherente dada su orientación Republicana y el propio interés popular en el tema (es de esperar que enfaticen en aquellos asuntos de mayor preocupación popular).

Si bien este análisis ya brinda información más valiosa que el simple conteo de palabras más reptiles sin ningún criterio realizado anteriormente, una estrategia sin duda más robusta podría consistir en ampliar el conjunto de términos representativos de cada categoría temática. Por ejemplo, para analizar el interés por el tema de la economía del país se puede considerar los siguientes términos afines: “economic”, “growth”, “inflation”, “gdp”, “jobs”, “employment”, “unemployment”, “wages”, “recession” y “recovery”.

Esto permite cuantificar aquellas menciones al mismo asunto hechas con términos no exactamente iguales a los considerados originalmente como palabras clave. El resultado se presenta en la Figura 7 y se puede observar algunos cambios en las abundancias relativas respecto a los resultados expuestos en la Figura 6 anterior.

Figura 7: Conteo de categorías clave abordadas por candidato

Si bien los temas más frecuentes parecen ser los mismos para todos los candidatos con este nuevo análisis, las abundancias se vieron ligeramente modificadas debido a la consideración de diferentes acepciones para el mismo tema.

Como este se pueden realizar otros análisis semejantes, agrupando la información disponible de acuerdo con otros criterios más allá de los temas de interés público. Un análisis interesante sería, por ejemplo, analizar la distribución en el tiempo de las menciones de ciertos términos, tal vez concentradas cerca de la fecha de ocurrencia de algún hecho.

El 3 de marzo de 2020, la Organización Mundial de la Salud (OMS) declaró oficialmente al COVID-19 como una pandemia global. Con el objetivo de analizar la importancia que cada candidato le otorgó a este tema a lo largo del año 2020, se identificaron en sus discursos los siguientes términos relacionados con la pandemia: “covid”, “coronavirus”, “pandemic”, “quarantine”, “vaccine”, “virus”, “mask” y “contagion”.

Para hacer comparables los resultados entre candidatos y a lo largo del tiempo, la frecuencia de mención fue normalizada según el número de discursos pronunciados por cada uno. Los resultados se presentan en la Figura 8, donde se observa que la mayoría de los candidatos comenzaron a referirse al tema a partir de marzo. En particular, Joe Biden mantuvo un número constante y elevado de menciones por discurso, mientras que Donald Trump —a pesar de ser presidente en ese momento— mostró picos de menciones aislados y, en general, una frecuencia inferior a la del candidato anteriormente mencionado.

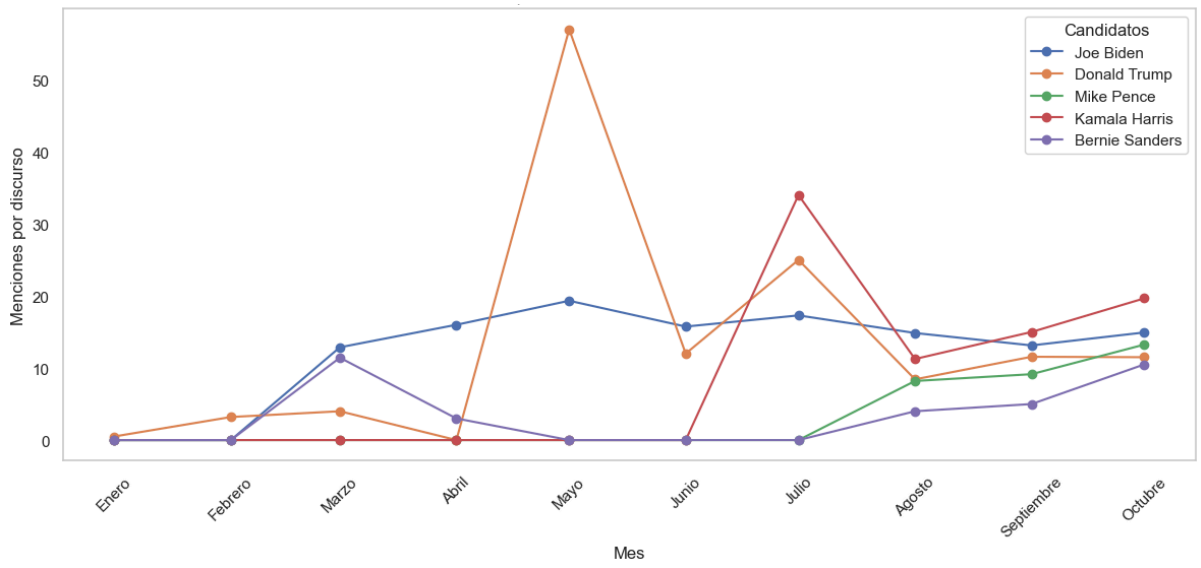

Figura 8: Menciones sobre la pandemia del COVID 19 en los discursos de los candidatos.

Otro hecho que marcó la política y la situación social de Estados Unidos fue el asesinato de George Floyd, ocurrido el 25 de mayo de 2020, cuando el oficial de policía Derek Chauvin presionó su rodilla contra su cuello durante más de nueve minutos durante un arresto. De forma análoga al análisis anterior, se realizó una búsqueda de términos relacionados con este acontecimiento en los discursos y se normalizó la frecuencia de mención. Los resultados se presentan en la Figura 9.

Como era esperable, las menciones comienzan en mayo, mes en que ocurrió el hecho, y alcanzan su punto máximo en junio, destacándose Donald Trump —entonces presidente— como el candidato con más referencias al tema. Llama la atención que Bernie Sanders no haya realizado ninguna mención, lo cual podría atribuirse a un sesgo en la selección de los términos utilizados para el análisis.

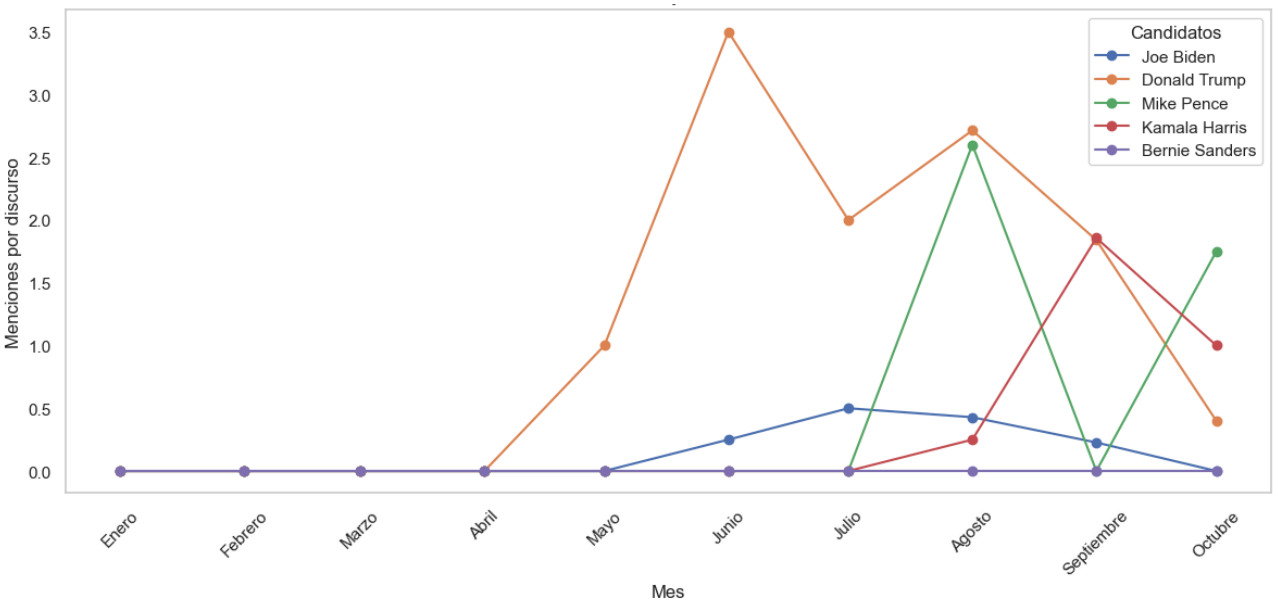

Figura 9: Menciones sobre el caso Floyd en los discursos de los candidatos.

Por otra parte, el análisis puede extenderse a nivel partidario. En lugar de evaluar individualmente a cada candidato, es posible agrupar los candidatos de acuerdo con su partido político y contabilizar las menciones de ciertos términos por bloque partidario. Esto permitiría identificar diferencias ideológicas o programáticas entre los principales partidos políticos estadounidenses de forma más ordenada.

2.5. Menciones cruzadas

Otro tema de interés a analizar puede ser la cantidad de menciones de cada candidato a los demás (dentro de los 5 candidatos que más discursos tienen).

Para eso se contabilizó la cantidad de veces que cada candidato mencionó el nombre completo, solo el nombre o solo el apellido (evitando contabilizar tres veces menciones a Nombre Apellido como “Nombre Apellido”, “Nombre” y “Apellido) de otro candidato. El resultado se presenta en la Figura 10 en formato de *heat map*.

Para este análisis se consideraron nulas las auto-menciones. Si bien es posible que, por ejemplo, “Donald Trump” haga referencia a la “Administración Trump”, en cuyo caso se mencionaría una vez a sí mismo, no se considera relevante para este análisis. Además, al omitir las menciones propias dentro del discurso, se evita el sesgo generado por palabras utilizadas en el ordenamiento del texto, como en los patrones: 'Donald Trump: ...', 'Speaker 1: ...', o similares, donde podrían contabilizarse erróneamente menciones del candidato a sí mismo.

Por otro lado, no se eliminaron del análisis las palabras correspondientes a intervenciones de personas externas (entrevistadores, periodistas u otros), lo que introduce un posible error asociado a la mención de un candidato por alguien que no es el orador principal. Esto se hizo para evitar el sesgo que se podría introducir al no considerar absolutamente todos los hablantes externos posibles.

Igualmente, es importante tener en cuenta que esta consideración puede generar ciertas desviaciones en los resultados presentados en la Figura 10.

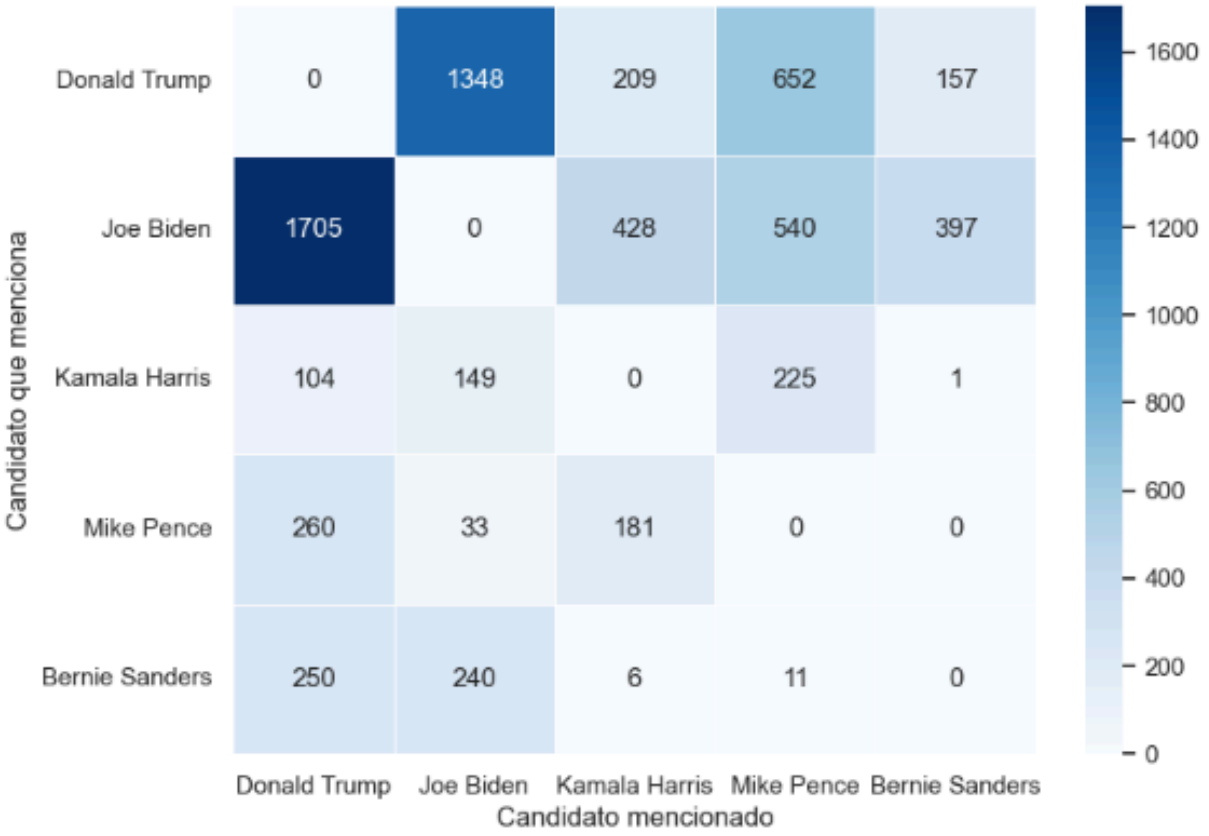

Figura 10: Menciones de unos candidatos a otros (ignorando auto-menciones) en discursos.

2.6. Preguntas a responder

En línea con el análisis anterior, algunas preguntas que surgen y pueden ser respondidas mediante el análisis del set de datos anteriormente presentado (y que van en la misma línea que algunas estrategias ya implementadas) son:

1. ¿Qué temas de interés público dominan los discursos de cada candidato?
   Esto fue lo que se pretendió evaluar con los *heat maps* presentados anteriormente.
2. ¿Cómo varía la frecuencia de ciertos temas a lo largo del tiempo?
   Por ejemplo, ¿cuándo comienza a aumentar el uso de términos asociados al COVID-19 o al asesinato de George Floyd?
3. ¿Qué candidato/a menciona más la palabra "justicia" o "economía"?
   Podría llevarse a cabo un análisis que hiciera énfasis en ciertos valores simbólicos importantes ya sea para el entrevistador o para un público específico al que se quiere apuntar una campaña electoral.
4. ¿Qué tan diverso es el vocabulario utilizado por cada candidato?
   Mediante métricas como riqueza léxica (número de palabras únicas / número total de palabras) o entropía (mide cuán uniformemente se usan las palabras en un texto).
   Útil si fuera de interés un análisis del estilo de oratoria de cada candidato. Por ejemplo, para evaluar por qué hay más personas que concuerdan o se oponen a las ideologías planteadas por algún candidato; o si alguno suena más convincente.
5. ¿Hay correlación entre ciertos temas?
   Por ejemplo, ¿los discursos que hablan de inmigración también tienden a hablar de seguridad o de desempleo?
6. ¿Cómo varía la longitud de los discursos según el orador? ¿Hay alguno que repita la misma idea una y otra vez dentro del discurso? ¿Quién es el más sintético?
7. ¿Se observa un cambio de tono o enfoque en discursos de campaña en comparación con entrevistas o debates?
8. ¿Hay algún entrevistador o cadena de televisión que presente un sesgo o mayor enfoque e interés hacia ciertos temas? ¿Todos hablan de los mismos?
9. ¿Qué temas se intensifican en los discursos más cercanos a las elecciones ?
   Por ejemplo en el mes de septiembre que es el de mayor cantidad de discursos o en el mes de octubre que es el que es inmediatamente anterior a las elecciones.
10. ¿Hay diferencias de estilo entre hombres y mujeres candidatos?
    Puede referir al uso de ciertos términos o conectores con mayor frecuencia, o también a menciones a ciertos temas de interés público (como cuidado, familia, comunidad).
11. ¿Qué candidato es más mencionado por otros candidatos?
12. ¿A qué cierto candidato menciona más otro candidato en particular?
13. Se puede evaluar la estrategia discursiva de Joe Biden, presidente electo del 2020, según su frecuencia, su oratoria, su léxico, el lugar donde los realiza y los temas en los que centra sus declaraciones.

3. Conclusión

El trabajo realizado para el presente informe permitió llevar a cabo un abordaje preliminar a la limpieza y transformación de un conjunto de datos compuesto por discursos políticos de presidentes de los Estados Unidos en el marco de las elecciones presidenciales del año 2020. Esto se realizó con el objetivo de habilitar su análisis cuantitativo, permitiendo comparar la intervención o relevancia de cada participante, así como los temas o palabras más empleadas por cada uno.

A partir de la detección de errores de inconsistencia y datos faltantes así como la estandarización del formato del campo que contenía las transcripciones de los discursos, fue posible construir representaciones estructuradas que facilitaron la exploración temática del contenido.

Se realizó un conteo general de palabras sin ningún criterio a priori y, en base a las conclusiones obtenidas, se definieron categorías semánticas relevantes (como salud, economía, educación, seguridad, entre otras) y se agruparon palabras clave bajo cada una de ellas para obtener información de mayor significado.

Esto permitió calcular y visualizar la frecuencia relativa con la que cada candidato abordó distintos temas. Las visualizaciones resultantes (por ejemplo, *heat maps*) ofrecieron una mirada clara sobre las prioridades discursivas de cada orador y permitieron detectar patrones interesantes.

Además del análisis temático, se exploró la dinámica de las menciones cruzadas entre candidatos. Este enfoque reveló qué figuras políticas fueron más mencionadas por sus pares. Incluso podría extenderse el análisis a la evaluación de en qué contextos se mencionó a otro candidato, permitiendo observar interacciones discursivas y estrategias de confrontación o alusión y, por lo tanto, aportando una dimensión relacional al análisis del lenguaje utilizado en campaña.

En síntesis, el trabajo mostró cómo una adecuada preparación del conjunto de datos, combinada con herramientas básicas de análisis de texto, permite extraer información significativa y relevante desde una perspectiva tanto descriptiva como comparativa.

Tarea 2

1. Introducción

La creciente disponibilidad de grandes volúmenes de texto digitalizado ha ampliado significativamente las posibilidades de análisis del lenguaje en sus diversas dimensiones, sociales, políticas y culturales. En este contexto, la ciencia de datos proporciona herramientas que permiten transformar el texto en objetos analizables desde una perspectiva cuantitativa, facilitando tanto la exploración profunda de su contenido como su clasificación automática. Este análisis implica necesariamente un proceso de representación intermedia, que consiste en convertir el lenguaje natural (con toda su riqueza y ambigüedad inherente) en estructuras numéricas que puedan ser procesadas e interpretadas por algoritmos computacionales.

El modelado de textos en lenguaje natural mediante estructuras numéricas se convierte, por lo tanto, en un paso central para el análisis computacional del lenguaje. Esta transformación plantea desafíos conceptuales y metodológicos, ya que toda representación supone una reducción, que define qué aspectos del texto se conservan y cuáles quedan excluidos del análisis.

El presente trabajo se enmarca en este proceso de modelado. En particular, se aborda el desafío de representar textos políticos (discursos, entrevistas o debates electorales) de manera que sea posible entrenar modelos supervisados capaces de identificar al orador a partir del contenido del discurso. Esta tarea involucra decisiones técnicas y conceptuales propias del análisis de datos textuales, ya que es necesario decidir qué elementos lingüísticos incluir como características, simplificarlos mediante procesos como la lematización y la eliminación de *stopwords*, y convertirlos en representaciones numéricas a través de la vectorización. Esta reducción de dimensionalidad debe equilibrarse con la conservación de los matices contextuales esenciales para que los modelos supervisados puedan diferenciar eficazmente entre distintos oradores.

El análisis parte de un subconjunto de discursos pronunciados en el marco de las elecciones presidenciales de Estados Unidos en el año 2020, previamente procesados y explorados en la tarea anterior. A partir de esta base, se ensayan distintas estrategias para representar los textos, entre ellas el modelo de bolsa de palabras (*Bag of Words*) y la ponderación de términos mediante TF-IDF (*Term Frequency-Inverse Document Frequency*). Se analizan las propiedades, limitaciones y potencial de estas representaciones para capturar estilos discursivos diferenciables entre candidatos. Posteriormente, se entrenan modelos de clasificación supervisada, en particular *Multinomial Naive Bayes*, *Regresión Logística* y *Support Vector Machine Lineal*, que al operar sobre estas representaciones buscan aprender patrones lingüísticos característicos de cada orador.

El interés principal de este trabajo reside en comprender el proceso por el cual un conjunto de discursos puede ser transformado en vectores, evaluado mediante modelos supervisados y analizado a partir de métricas como la exactitud (*accuracy*), la matriz de confusión, la precisión (*precision*) y el *recall*; lo que permite una reflexión crítica sobre las posibilidades y limitaciones del aprendizaje automático aplicado al análisis del lenguaje.

2. Desarrollo

2.1. Limpieza de datos

Los datos empleados en este trabajo consisten en la transcripción de 269 discursos, junto con su clasificación, título, fecha y lugar de pronunciación, correspondientes a candidatos presidenciales y figuras políticas relevantes durante el proceso electoral de Estados Unidos en 2020.

En primer lugar, tras una inspección visual general del Data Frame cargado con los discursos de los candidatos, con el objetivo de observar su estructura, clasificación y orden, se examinó la primera columna *speaker* del Data Frame, correspondiente a los nombres de los candidatos presidenciales que pronunciaron discursos. Para visualizar los objetos (*inputs*) únicos registrados, así como la cantidad de apariciones de cada uno, se utilizó la función *value_counts*.

En los casos donde un mismo discurso figuraba bajo la autoría de más de un candidato (siendo la denominación *speaker* sus nombres correspondientes separados por comas), dicho discurso fue multiplicado y asignado individualmente a cada uno. Esto se realizó mediante la combinación de las funciones *str.split(',')* y *explode*. Aquellos discursos registrados bajo las etiquetas "Multiple Speakers" o "Democratic Candidates" no pudieron ser asignados a ningún candidato en particular, debido a la falta de información específica.

Tomadas estas consideraciones iniciales, se seleccionaron los discursos de los 3 candidatos con mayor frecuencia de aparición, reduciendo las entradas del Data Frame de 269 a 155, con el objetivo de facilitar el tratamiento posterior de la información y otorgar mayor claridad y relevancia a los análisis y visualizaciones subsiguientes. Los candidatos así considerados fueron: Joe Biden (81 discursos), Donald Trump (54 discursos) y Mike Pence (20 discursos).

A continuación, dado que los algoritmos de aprendizaje automático entrenados en este trabajo no operan directamente sobre texto en lenguaje natural, fue necesario aplicar un proceso de limpieza y preprocesamiento para transformar las transcripciones en datos adecuados para su representación vectorial. Esto incluyó la normalización del texto, la lematización, la eliminación de palabras vacías (*stopwords*) y otros elementos no informativos, así como la conversión a minúsculas y la remoción de signos de puntuación.

Estas etapas son fundamentales para reducir el ruido, estandarizar la información y conservar únicamente los componentes lingüísticos más relevantes, lo que permite una representación más consistente del contenido y mejora la capacidad de los modelos para identificar patrones discursivos diferenciables.

2.1.1. Normalización y estandarización

Como primer paso para el análisis de los discursos, es necesario normalizar el texto de sus transcripciones, es decir, convertir las palabras a minúsculas y eliminar signos de puntuación.

Con este fin, se utilizó la función personalizada *clean_text* para eliminar las primeras palabras hasta el primer \n (normalmente correspondientes a identificadores de la transcripción, y no al propio contenido del discurso), convertir el texto a minúsculas y eliminar signos de puntuación como puntos, comas, paréntesis, comillas, signos de exclamación e interrogación, \n, etc.

El texto "limpio" de cada discurso se almacenó en una nueva columna *CleanText* como parte del Data Frame original.

2.1.2. Eliminación de stopwords

Como paso siguiente del preprocesamiento, se optó por eliminar las *stopwords*, es decir, palabras muy frecuentes que suelen aportar poco contenido semántico por sí solas (como artículos, preposiciones o pronombres) y que tienden a generar ruido en las representaciones vectoriales del texto. Su remoción contribuye a reducir la dimensionalidad y a centrar el análisis en los términos más informativos para los modelos de clasificación.

Para esto se importó y descargó una lista de *stopwords* del idioma inglés del kit de herramientas de lenguaje natural (NLTK) de Python, que es un conjunto de bibliotecas y programas específicos para el procesamiento de lenguaje natural.

Para la eliminación de estas *stopwords* del texto, se aplicó la función personalizada *remove_stopwords* que combinó un paso de tokenización *word_tokenize* para convertir el texto transcripto en palabras individuales (tokens) que se pudieran comparar una a una con la lista de *stopwords*, y un paso de filtración de estas palabras encontradas.

2.1.3. Expansión de contracciones

Además, se elaboró manualmente una lista de las contracciones más comunes del idioma inglés con el objetivo de implementar una función que las expandiera (por ejemplo, sustituir "didn't" por "did not"). Si bien "didn't" y "did not" son expresiones fonéticamente diferentes, a la hora de contabilizar palabras utilizadas, representan el mismo significado ("did" y "not"), por lo que se consideró contabilizarlas junto con sus dos palabras constitutivas y no como una palabra propia.

Para esto se utilizó la función personalizada *expand_contractions*, que identificó las diferentes contracciones mediante la presencia de un apóstrofe gráfico y las sustituyó por su expansión correspondiente.

Este paso se aplicó posteriormente a la eliminación de *stopwords* dado que muchas de estas eran contracciones y podían ser eliminadas directamente; pero igualmente se consideró un paso necesario en caso que no todas las contracciones listadas estuvieran clasificadas como *stopword*.

2.1.4. Eliminación de números

Alcanzado este punto del preprocesamiento y analizando los vectores de palabras obtenidos mediante la *tokenización* de los textos de las transcripciones de los discursos, se encontró que cerca de 400 elementos (de aproximadamente 11.000) corresponden a números: cantidades (miles, millones) , porcentajes, años, fechas (días del mes), ordinales (1st, 2nd, 3rd), etc.

Este tipo de expresiones aparece de forma recurrente en los textos de todos los candidatos sin aportar distinciones significativas entre estilos discursivos. Además, tienden a generar ruido en el modelo, al introducir un gran número de tokens poco informativos como “2020”, “2nd” o “100”, que aumentan la dimensionalidad sin contribuir a la clasificación.

Dado que el interés del análisis se centró en las palabras y estructuras lingüísticas características de cada orador, se optó por excluir los elementos numéricos del corpus. Esto se realizó mediante la función personalizada *remove_numbers* que buscó y eliminó los números presentes en los textos junto con sus sufijos (“k”, “nd”, “s”, etc.).

2.1.5. Lematización

La lematización es el proceso de reducir una palabra a su forma base o "lema", teniendo en cuenta su significado y función gramatical. Este procedimiento se basa en reglas lingüísticas y diccionarios para identificar la forma canónica reconocida por el idioma. Aplicar la lematización a los discursos presidenciales permite normalizar el vocabulario, representando de manera unificada términos relacionados o equivalentes, y puede contribuir a mejorar el rendimiento de los modelos de clasificación al reducir la redundancia léxica.

Este proceso se realizó utilizando el lematizador de *WordNet*, en conjunto con el etiquetado gramatical (POS tagging) provisto por NLTK, que asigna a cada palabra su categoría gramatical (sustantivo, verbo, adjetivo o adverbio). Mediante la función personalizada *lematize_text*, los textos fueron, en primer lugar, tokenizados, es decir, divididos en unidades léxicas individuales, lo que permitió aplicar el lematizador y etiquetado palabra por palabra. Esta estrategia posibilitó reducir cada término a su forma base adecuada según el contexto, mejorando la precisión del proceso y evitando errores derivados de asumir una única categoría gramatical por defecto.

Resulta de interés aclarar que, dado que la eliminación de *stopwords* se realizó previo a esta etapa, en algunos casos el etiquetador POS tuvo menos contexto para determinar correctamente la categoría de ciertas palabras, lo que pudo afectar la lematización en forma puntual.

Una vez finalizado el preprocesamiento, el texto 'limpio' de cada discurso fue almacenado sobrescribiendo la columna *CleanText* del DataFrame original.

2.2. Representación numérica de textos

2.2.1. Muestreo para entrenamiento y evaluación

Con el fin de evaluar el rendimiento de distintos modelos de clasificación de forma objetiva, se dividió el conjunto de discursos de los diferentes candidatos en dos partes: un conjunto de entrenamiento, utilizado para realizar el ajuste de los parámetros del modelo, y un conjunto de prueba, reservado para evaluar la capacidad de generalización del modelo sobre datos no vistos durante el entrenamiento. Esta división es fundamental para evitar sobreajuste al conjunto de entrenamiento y estimar cómo se comportará el modelo frente a nuevos textos.

Para esto, se destinó el 30 % de los datos al conjunto de prueba y el restante 70% al de entrenamiento. Para realizar la separación se aplicó un muestreo estratificado, que garantiza que la distribución de clases (en este caso, los oradores o candidatos) se mantenga proporcional en ambos subconjuntos, lo que contribuye a una evaluación más representativa y justa.

La partición se realizó con la función *train_test_split* de la biblioteca de Python *scikit-learn*, empleando el parámetro *stratify* para mantener la proporción de oradores y fijando un valor de *random_state* para asegurar la reproducibilidad de los resultados.

De esta forma, el conjunto de 155 transcripciones de discursos de candidatos se dividió en uno de 108 textos, destinado al entrenamiento del modelo de clasificación, y uno de 47 textos, destinado a su evaluación. La Figura 1 presenta de manera visual la distribución proporcional de las clases (candidatos u oradores) en los conjuntos de entrenamiento y prueba, reflejando el muestreo estratificado aplicado durante la partición.

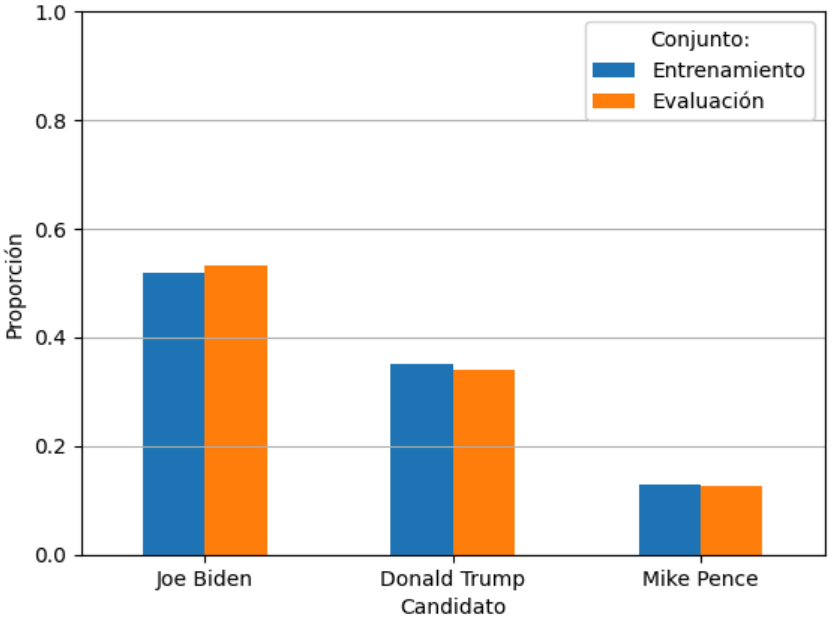

Figura 1: Proporción de discursos por candidato en los conjuntos de entrenamiento y evaluación.

2.2.2. Representación Bag of Words

A continuación, se transformaron los textos del conjunto de entrenamiento utilizando la técnica de Bag of Words (BoW). Esta representación numérica consiste en convertir cada uno de los textos de los discursos en un vector de frecuencias, donde cada dimensión del vector corresponde a una palabra distinta del vocabulario total extraído de los textos. El valor de cada componente del vector indica cuántas veces aparece esa palabra del vocabulario en el texto en cuestión.

A modo de ejemplo ilustrativo se consideran los siguientes 3 textos:

- Texto 1: "yo tengo una mascota que es un perro".
- Texto 2: "yo tengo dos mascotas una que es un perro y una que es un gato"
- Texto 3: "yo no tengo mascotas pero quiero un conejo"

Para aplicar la técnica BoW, en primer lugar se construye el vocabulario con todas las palabras únicas encontradas en los textos (ordenadas por orden alfabético): ["conejo", "dos", "es", "gato", "mascota", "mascotas", "no", "pero", "perro", "que", "quiero", "tengo", "una", "un", "yo", "y"].

Luego, cada texto se convierte en un vector que indica cuántas veces aparece cada palabra del vocabulario. Esto da lugar a la matriz de la Tabla 1, donde el número de filas corresponde a la cantidad de textos y el número de columnas a la cantidad de elementos del vocabulario.

Tabla 1: Ejemplo ilustrativo de matriz de BoW

	conejo	**dos**	**es**	**gato**	**mascota**	**mascotas**	**no**	**pero →**
Texto 1	0	0	1	0	1	0	0	0
Texto 2	0	1	1	1	0	1	0	0
Texto 3	0	0	0	0	1	1	1	0
	→ perro	**que**	**quiero**	**tengo**	**una**	**un**	**yo**	**y**
Texto 1	1	1	0	1	1	1	1	0
Texto 2	1	1	0	1	2	2	1	1
Texto 3	0	1	1	0	1	1	0	0

Aplicando el mismo procedimiento al conjunto de transcripciones de discursos de los candidatos, se utilizó la clase *CountVectorizer* de la biblioteca *scikit-learn*. Esta herramienta construye automáticamente un vocabulario a partir de todas las palabras presentes en el corpus (conjunto de 108 textos) y transforma los textos en una matriz numérica, donde cada fila representa un texto y cada columna indica la frecuencia de una palabra específica en ese texto.

A modo de ejemplo, algunas palabras del vocabulario construido son:
“ablaze”, “able”, “abnormal”, “aboard”, “abolish”, “abolition”, “abolitionist”, “abomination”, “abortion”, “abraham”, (...), “blustery”, “bo”, “board”, “boardwalk”, “boat”, “boating”, “boatload”, “bob”, “bobby”, “body”, …

Entonces, la matriz resultante tiene dimensiones (n_textos, n_palabras), donde cada fila representa un discurso y cada columna una palabra del vocabulario. En este caso, dado que el conjunto de entrenamiento contiene 108 discursos y la totalidad de los textos presenta 10.877 palabras únicas, la matriz BoW tiene una forma de (108, 10.877).

Esta matriz es dispersa (*sparse*) porque, en la mayoría de los textos analizados, solamente está presente una pequeña fracción del total de palabras del vocabulario. Es decir, la mayoría de los valores en la matriz son ceros.

Esto se visualiza claramente en el ejemplo presentado en la Tabla 1. Y respecto al análisis de las transcripciones de los discursos, de los 108x10.877 elementos de la matriz de BoW resultante, solamente 101.282 son distintos de cero (aproximadamente el 8,6%).

Una vez se reconoce la matriz como *sparse*, en vez de almacenar todos los elementos de la matriz (incluyendo los ceros), se pueden usar estructuras de datos especiales que solamente almacenan las posiciones y valores de los elementos distintos de cero, lo que no solo ahorra memoria, sino que también acelera los cálculos en muchos algoritmos de aprendizaje automático.

Por lo tanto, en tareas de procesamiento de lenguaje natural, trabajar con matrices dispersas es esencial para lograr eficiencia computacional. Sin esta optimización, incluso un conjunto de datos relativamente acotado, como el utilizado en este informe, podría implicar un uso ineficiente de la memoria RAM, dificultando su procesamiento.

2.2.3. Representación Term Frequency - Inverse Document Frequency

Una vez construida la representación inicial de los textos mediante el modelo de bolsa de palabras, se implementó una transformación adicional TF-IDF. Esta técnica permite ponderar cada palabra del vocabulario construido no solo por su frecuencia en un texto individual, sino también considerando su distribución en el conjunto total de textos.

Esta transformación toma en cuenta dos consideraciones:

- TF (Term Frequency): representa cuántas veces aparece una palabra en un texto dado. Cuanto mayor el número de apariciones, más importante se considera en ese texto.
- IDF (Inverse Document Frequency): penaliza aquellas palabras que aparecen en casi todos los textos, ya que se asume que no aportan información distintiva.

La combinación de ambos factores da lugar a una matriz numérica similar a la de BoW, pero con valores continuos (decimales) en lugar de enteros.

Esta transformación, TF-IDF, se aplicó a los textos de los discursos del conjunto de entrenamiento utilizando la clase *TfidfTransformer* de la biblioteca *scikit-learn*, que permitió conservar el mismo vocabulario de la matriz BoW, pero ajustando los valores de forma que reflejaran mejor la importancia relativa de cada término. Mediante esta transformación, los modelos de clasificación evaluados posteriormente pudieron enfocarse en aquellas palabras que efectivamente contribuyen a diferenciar entre estilos discursivos o candidatos, y no en aquellas comunes a todos los textos.

En la Figura 2 se presenta, a modo de ejemplo, la transformación TF-IDF aplicada al primer discurso del conjunto de datos: un discurso de campaña pronunciado por Joe Biden el 16 de octubre de 2020 en Michigan. La visualización adopta la forma de una nube de palabras, en la que se incluyen todos los términos con un valor de TF-IDF mayor a cero, representados con un tamaño proporcional a dicho valor. Esta representación permite identificar de manera visual qué palabras adquieren mayor relevancia en el discurso según esta ponderación.

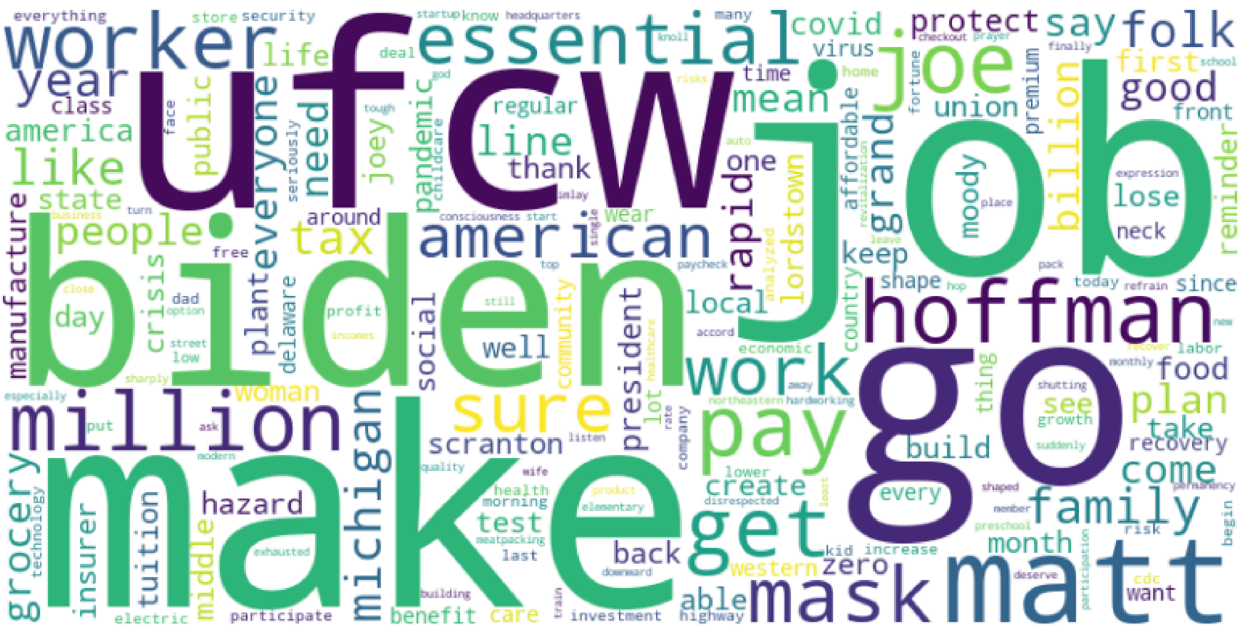

Figura 2: Nube de palabras con valor de TF-IDF mayor a cero del discurso de campaña de Joe Biden del 16 de octubre de 2020 en Michigan.

De esta forma, los términos muy frecuentes pero poco informativos (como “time”, “still” y “low”) tienen un peso menor, mientras que los términos menos comunes y más específicos para este discurso y orador (“job”, “make”, “UFCW”, “Biden”) adquieren mayor relevancia.

2.2.4. Incorporación de n-gramas

Tras aplicar las representaciones Bag of Words y TF-IDF sobre palabras individuales, cabe mencionar que una alternativa para enriquecer la caracterización de los textos consiste en incorporar n-gramas, es decir, secuencias de n palabras consecutivas. Por ejemplo, un bigrama permite captar combinaciones como “derechos humanos” o “cambio climático”, que pueden expresar ideas compuestas o estilos discursivos recurrentes que se perderían si se consideran solamente palabras aisladas.

Esta técnica puede combinarse con la transformación TF-IDF en la clase *TfIdfVectorizer* de la biblioteca *scikit-learn*. El atributo *ngram_range* permite decidir si la representación Bag of Words se hace con palabras individuales o con secuencias de palabras (n-gramas). Por ejemplo, el parámetro *ngram_range = (1,2)* indica al vectorizador que construya un vocabulario que incluya tanto unigramas (palabras individuales) como bigramas (pares de palabras consecutivas) y los pondere según su frecuencia en cada texto y su rareza en el corpus, asignando mayor peso a los elementos más representativos de cada texto. De este modo, se mejora la capacidad de capturar expresiones características y se refuerza el valor discriminativo de la representación.

Si bien hasta este punto del desarrollo sólo se trabajó con palabras individuales (unigramas), la incorporación de n-gramas, como los bigramas, fue considerada posteriormente al analizar el impacto de distintos hiperparámetros en el desempeño de los modelos entrenados.

Además, en el caso del Análisis de Componentes Principales (PCA), se estudió la influencia de los bigramas como caso comparativo.

2.2.5. Técnicas alternativas de representación de texto

Una técnica ampliamente utilizada como alternativa a TF-IDF es Word2Vec. A diferencia de las representaciones basadas en conteo, como Bag of Words o TF-IDF, que tratan a las palabras como unidades independientes sin contexto, Word2Vec genera vectores densos y continuos que capturan relaciones semánticas y contextuales entre palabras. Esto se logra a partir de un entrenamiento no supervisado que toma grandes cantidades de texto y utiliza redes neuronales simples para aprender patrones de coocurrencia.

El modelo puede entrenarse con dos enfoques principales: Skip-Gram, que predice el contexto de una palabra dada, y CBOW (Continuous Bag of Words), que predice una palabra a partir de su contexto. En ambos casos, el objetivo es que palabras que aparecen en contextos similares (como “presidente” y “gobernador”) resulten representadas por vectores numéricos cercanos entre sí, incluso si no comparten raíces ni frecuencia de aparición. De este modo, Word2Vec capta el significado aproximado de una palabra en función de cómo se utiliza, algo que técnicas como TF-IDF y BoW no pueden hacer.

Una ventaja significativa de Word2Vec es su capacidad para reflejar similitudes semánticas y sintácticas, lo que permite al modelo generalizar mejor cuando se enfrenta a vocabularios compartidos entre clases o a expresiones con estructuras similares.

Por lo tanto, si se aplicara esta técnica al problema tratado en este trabajo, se esperaría una mayor sensibilidad a los matices del lenguaje usados por cada candidato, lo cual podría mejorar el desempeño del clasificador, especialmente en clases con vocabulario más sutil o similar al de otros oradores. No obstante, Word2Vec también requiere más datos y mayor tiempo de entrenamiento, y su implementación implica una mayor complejidad técnica en comparación con las transformaciones utilizadas previamente.

2.3. Análisis de Componentes Principales

Para explorar la distribución de los discursos según sus características textuales, se aplicó un Análisis de Componentes Principales (PCA) sobre los vectores generados previamente mediante la representación TF-IDF.

PCA es una técnica no supervisada de reducción de dimensionalidad. Esto significa que no utiliza las etiquetas de clase (en este caso, el orador del discurso) durante su cálculo, sino que únicamente considera las características numéricas de los datos.

El método tiene como objetivo proyectar los datos en un espacio de menor dimensión, conservando la mayor cantidad posible de información, entendida como la varianza de los datos. Para ello, se buscan nuevas variables no correlacionadas entre sí, denominadas componentes principales, que corresponden a combinaciones lineales de las variables originales.

Cada componente captura una dirección en la que la variabilidad de los datos es máxima, permitiendo así representar la estructura del conjunto de datos de forma más compacta y eficiente:

- La primera componente principal representa la dirección en la que los datos presentan mayor varianza.
- La segunda componente principal es ortogonal a la primera y representa la segunda mayor fuente de variación, y así sucesivamente.

Esta transformación se llevó a cabo utilizando la clase *pca* del módulo *decomposition* de la biblioteca *scikit-learn*, especificando el parámetro *n_components* = 2, lo cual indica que se desea conservar únicamente las dos primeras componentes principales del conjunto de datos. Estas dos componentes corresponden a las direcciones en el espacio vectorial original que explican la mayor cantidad de variación en los datos, y permiten representar los discursos en un plano bidimensional.

Para aplicar la transformación, se utilizó el método *fit_transform*, que cumple la doble función de ajustar el modelo PCA a los datos, es decir, calcular las componentes principales a partir de los vectores TF-IDF; y transformar los datos originales proyectándolos sobre estas nuevas direcciones. Dado que los vectores TF-IDF se encuentran en formato de matriz dispersa, fue necesario convertirlos previamente a un arreglo denso para que pudieran ser procesados por PCA por medio de la función *toarray*.

Los resultados de la reducción de dimensionalidad se presentan en la Figura 3, donde se visualiza el conjunto de entrenamiento proyectado sobre las dos primeras componentes principales obtenidas a partir de los vectores TF-IDF.

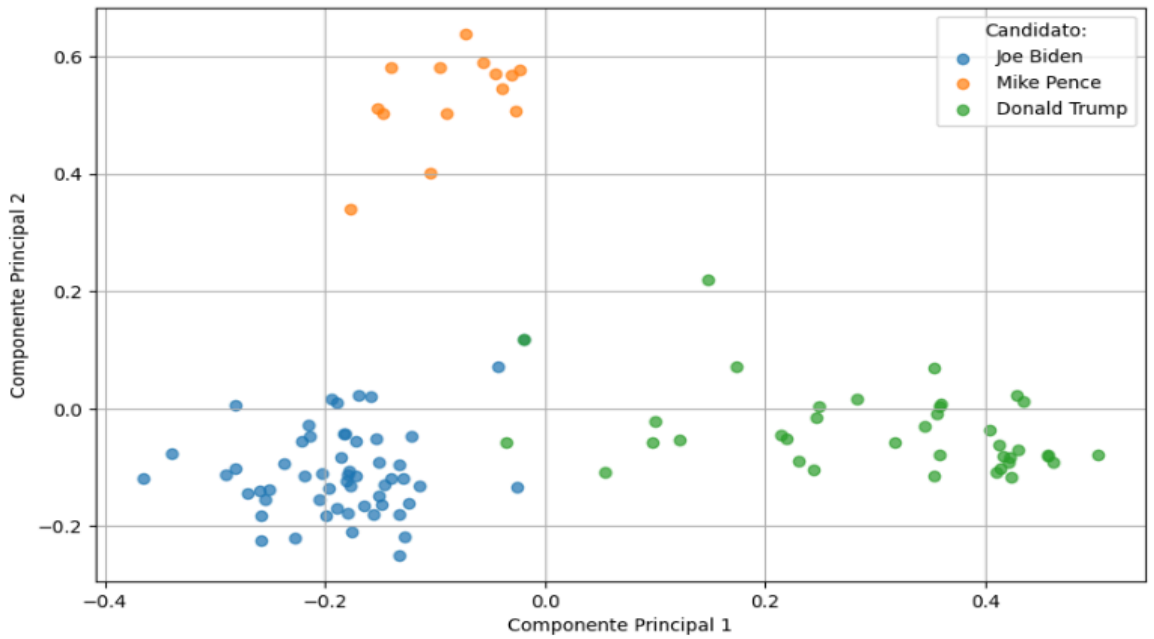

Figura 3: Proyección del conjunto de entrenamiento sobre las dos primeras componentes principales obtenidas mediante PCA, aplicadas a los vectores TF-IDF.

En la figura se observan agrupamientos relativamente definidos en algunas configuraciones, lo que sugiere que existen diferencias estilísticas o léxicas que permiten cierta separación entre oradores. No obstante, también se evidencian zonas de superposición, lo que indica que algunos discursos comparten características comunes.

Por otra parte, se consideró de particular interés analizar cómo se componía la primera componente principal a partir de los términos del vocabulario.

Para ello, se utilizó el atributo *.components_* del objeto PCA, el cual proporciona las combinaciones lineales de las variables originales (en este caso, los unigramas vectorizados) que definen cada componente principal. Esto, en combinación con la función *argsort* de la biblioteca *numpy*, permitió identificar qué palabras tenían mayor peso en la construcción de esta componente, ofreciendo una interpretación más concreta de las dimensiones latentes obtenidas mediante la reducción de dimensionalidad.

Los resultados pueden visualizarse mediante la nube de palabras presentada en la Figura 4, la cual representa gráficamente los términos más influyentes en la primera componente principal. El tamaño de cada palabra está jerarquizado en función de su peso relativo dentro de la componente correspondiente, lo que permite interpretar de manera intuitiva qué términos contribuyen en mayor medida a la variación explicada por el modelo.

Figura 4: Nube de palabras de los términos más influyentes en la primera componente principal del análisis PCA.

Adicionalmente, definiendo la varianza explicada como la proporción de la variabilidad total en los datos que es capturada por cada componente principal (es decir, cuánta información de los datos originales representa cada componente), se obtuvo que las dos primeras componentes explicaron en conjunto aproximadamente un 17 % de la varianza total. Si bien este valor es relativamente bajo, la proyección en dos dimensiones permite una visualización exploratoria de la estructura general del conjunto de datos y de posibles agrupamientos entre discursos según el autor.

Complementariamente, se generó un gráfico de la varianza explicada acumulada en función de la cantidad de componentes, con fines exclusivamente exploratorios, el cual se presenta en la Figura 5.

Si bien no existe un umbral único y estricto, se considera razonable conservar un número de componentes que expliquen al menos entre el 70 % y el 90 % de la varianza total. En este caso esto se correspondería con la consideración de más de 50 componentes.

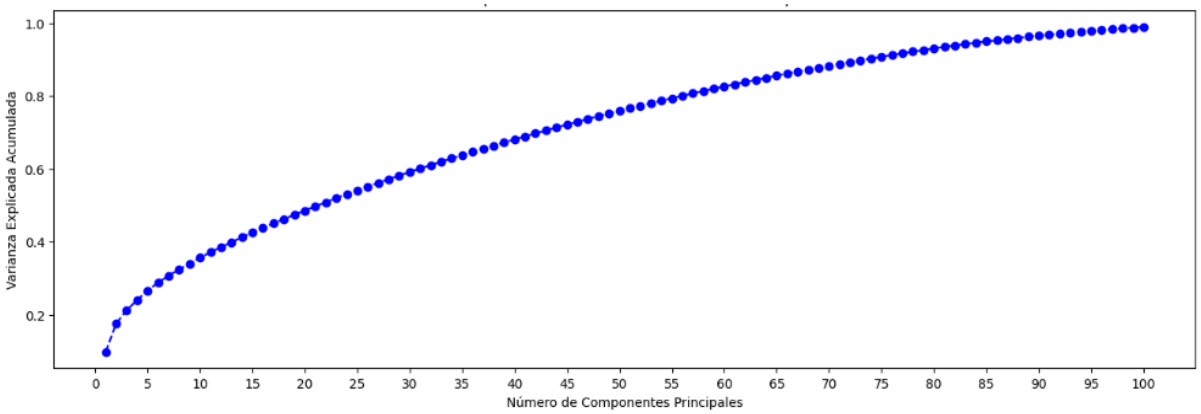

Figura 5: Varianza explicada acumulada en función de la cantidad de componentes principales.

En este caso, la observación fue utilizada únicamente como referencia para comprender la complejidad del espacio vectorial generado a partir del modelo TF-IDF, sin que derivara en análisis adicionales.

Paralelamente, y como complemento del desarrollo anterior, se exploraron variantes del preprocesamiento del texto con el objetivo de refinar el análisis. En particular, se evaluaron los resultados del PCA en tres configuraciones alternativas: (a) sin eliminar las *stopwords* del idioma inglés (Figura 6.a), (b) sin aplicar ningún tipo de limpieza previa, es decir, conservando signos de puntuación, contracciones y sin realizar lematización, (Figura 6.b), e (c) incorporando bigramas en la vectorización, dado que por defecto solo se consideran unigramas (Figura 6.c).

Para estos análisis, se utilizó el parámetro *use_idf* = True en el transformador TF-IDF, lo cual permite ponderar los términos según su frecuencia inversa en los documentos: esto penaliza las palabras comunes en todos los textos y destaca aquellas más específicas de ciertos discursos. Asimismo, se ajustó el parámetro ngram_range, definiéndose como (1,1) para trabajar únicamente con unigramas, o como (1,2) para incluir también a los bigramas en la representación vectorial.

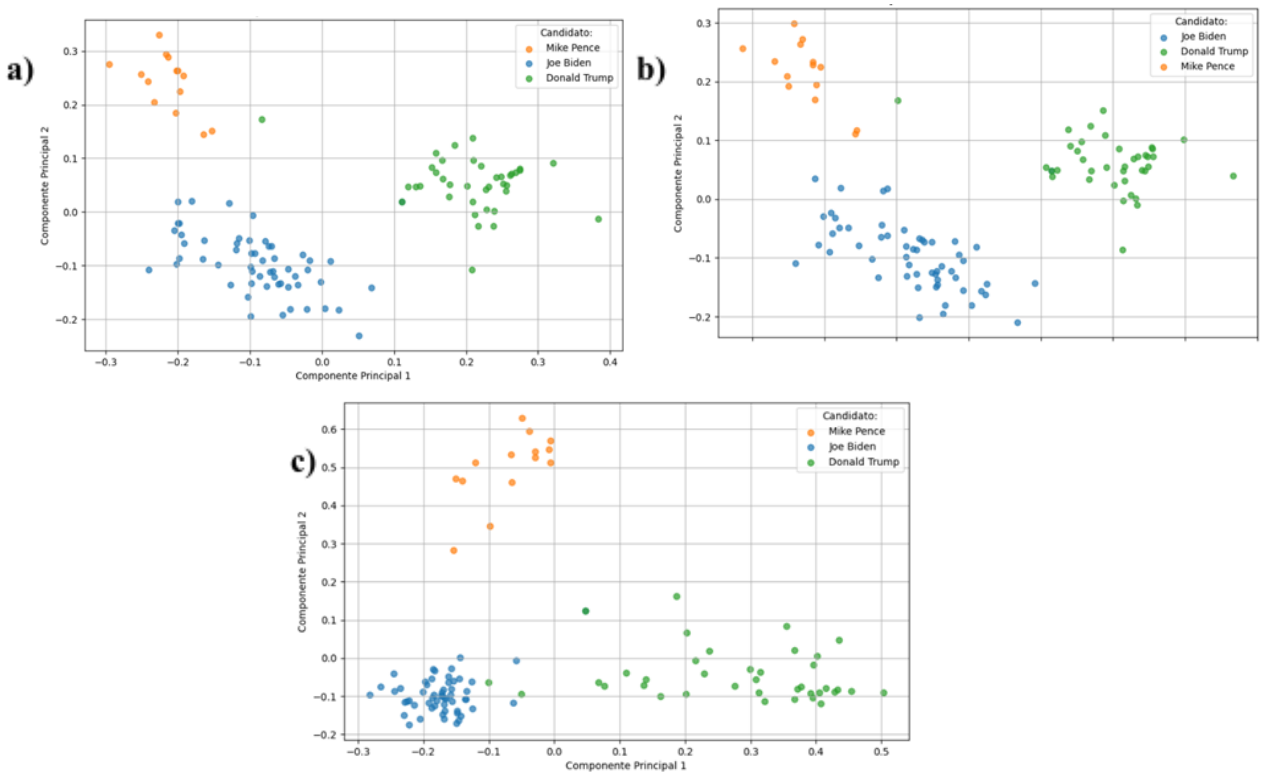

Figura 6: Proyección del conjunto de entrenamiento sobre las dos primeras componentes principales obtenidas mediante PCA, aplicadas a los vectores TF-IDF.
Figura 6.a Sin eliminación de *stopwords*, *use_idf* = *True*, *ngram_range* = (1,1). **Figura 6.b** Sin ninguna limpieza previa, *use_idf* = *True*, *ngram_range* = (1,1). **Figura 6.c** Todos los pasos de limpieza previos, *use_idf* = *True*, *ngram_range* = (1,2).

En principio, estas condiciones alternativas de preprocesamiento no modificaron de manera significativa la visualización resultante de la reducción de dimensionalidad mediante PCA. En todos los casos se observaron tanto agrupamientos relativamente definidos como zonas de superposición entre clases, sugiriendo que existen diferencias léxicas o estilísticas que permiten cierta separación entre oradores, aunque algunos discursos comparten características comunes.

Las diferencias más notables se relacionaron con la dispersión de los puntos correspondientes a cada candidato. Al omitir ciertos pasos de limpieza del texto, los discursos de Joe Biden tendieron a distribuirse de manera más dispersa, mientras que los de Donald Trump se agruparon de forma más compacta. Además, la inclusión de *stopwords* como parte del texto alteró la composición de los términos con mayor peso en las componentes principales, lo cual sugiere que estas palabras podrían tener un rol relevante en el estilo discursivo característico de cada orador.

Dado que los resultados generales del PCA no se vieron sustancialmente afectados, se decidió continuar el análisis adoptando la estrategia de preprocesamiento más completa (incluyendo lematización, eliminación de puntuación y stopwords) y utilizando los siguientes parámetros para la vectorización: *use_idf* = True y *ngram_range* = (1,1).

No obstante, las configuraciones alternativas también fueron consideradas al momento de explorar distintos conjuntos de hiperparámetros y al comparar el rendimiento de los modelos entrenados, a modo de análisis complementario.

2.4. Entrenamiento y evaluación de modelos

2.4.1. Modelo Multinomial Naive Bayes

Con el objetivo de clasificar los discursos según el candidato o candidata correspondiente, se entrenó el modelo Multinomial Naive Bayes utilizando el conjunto de desarrollo previamente vectorizado mediante TF-IDF.

El modelo Multinomial Naive Bayes es un clasificador probabilístico basado en el teorema de Bayes, especialmente diseñado para datos discretos como conteos de palabras en textos. Asume que las características (por ejemplo, la presencia o frecuencia de términos en un texto) son condicionalmente independientes entre sí dado el valor de la clase.

Esta suposición de independencia es lo que le da el nombre de "*naive*" (ingenuo), ya que en la práctica las características (como palabras de un texto) suelen estar correlacionadas. Aún así, esta simplificación permite realizar los cálculos de forma eficiente y, a pesar de su naturaleza poco realista, el modelo suele ofrecer un buen rendimiento en tareas de clasificación de texto.

En este contexto, el modelo estima la probabilidad de que un texto pertenezca a una clase determinada (en este caso, un orador específico) en función de su distribución de palabras, y se adapta correctamente a representaciones como bag of words o TF-IDF.

Para el entrenamiento de este modelo, se aplicó el método *fit* sobre la matriz TF-IDF de los discursos de los oradores del conjunto de desarrollo, junto con sus etiquetas reales ("Joe Biden", "Donald Trump" o "Mike Pence").

Una vez entrenado, el modelo fue utilizado para predecir las etiquetas del conjunto de prueba. Para esto, los textos fueron primero transformados al espacio de bag of words mediante el vectorizador ya ajustado, aplicando la función *vectorizer.transform*; y luego convertidos a su representación TF-IDF con el objeto *tfidf_transformer*, a través de la función *transform*.

La evaluación del modelo se realizó mediante el cálculo de la exactitud (*accuracy*), definida como la proporción de predicciones correctas sobre el total de muestras. En este caso, se obtuvo un valor de *accuracy* igual a 0.83, lo que indica un buen rendimiento general del clasificador en la tarea de distinguir entre los discursos de los distintos candidatos.

Además, se construyó la matriz de confusión utilizando el método *from_predictions* de la clase *ConfusionMatrixDisplay*, lo cual permitió visualizar gráficamente la relación entre las etiquetas predichas y las reales. Esta matriz se presenta en la Figura 7, donde pueden observarse tanto los verdaderos positivos (elementos correctamente clasificados) como los errores de clasificación cometidos por el modelo para cada candidato.

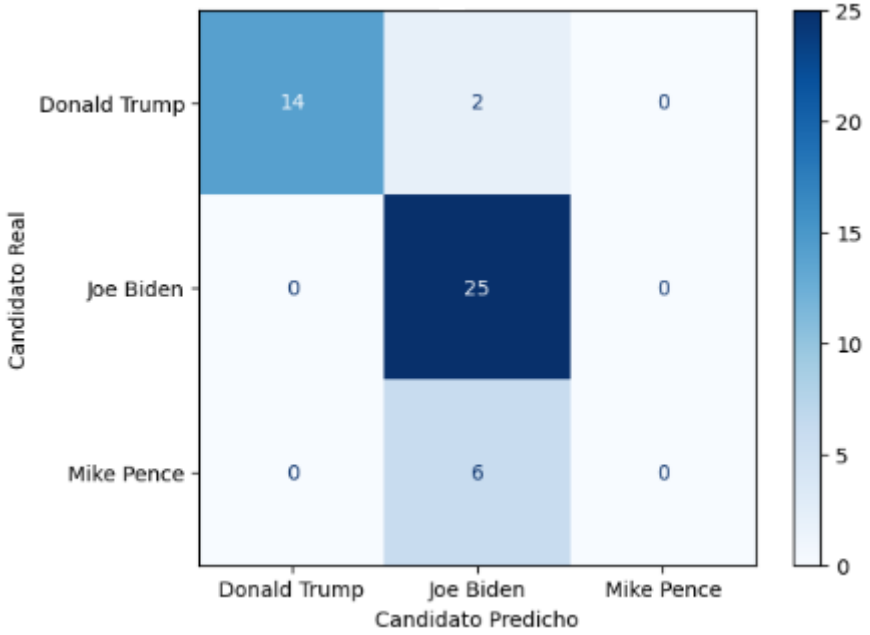

Figura 7: Matriz de confusión del modelo Multinomial Naive Bayes sobre el conjunto de prueba (*test*).

Una observación relevante a partir de la Figura 7 es que el modelo tiende a predecir con mayor frecuencia discursos de Joe Biden, incluso cuando el discurso pertenecía a otro candidato. Esto puede deberse a que la clase correspondiente a Biden está sobrerrepresentada

en el conjunto de entrenamiento, lo que lleva al modelo a inclinarse por esta etiqueta al tomar decisiones.

Además, se destaca que la clase correspondiente a Mike Pence no fue predicha en ningún caso, lo que indica que el modelo no logró identificar sus discursos. Sin embargo, a pesar de esta omisión, la *accuracy* global del modelo se mantiene alta (0.83), lo cual refleja que la clase de Pence está subrepresentada y, por tanto, su impacto en la métrica de exactitud es reducido.

Con este resultado queda en evidencia que considerar únicamente el valor de *accuracy* como forma de evaluación del modelo puede ser problemático, especialmente en contextos con desbalance de clases, ya que si una/s clase/s está/n sobrerrepresentada/s, el modelo puede obtener un valor alto simplemente prediciendo siempre la/s clase/s mayoritaria/s, sin realmente aprender a distinguir entre las otras clases.

Si el desbalance fuera aún mayor, el modelo podría obtener una *accuracy* incluso más alta prediciendo sistemáticamente la clase más frecuente, mientras que las clases minoritarias quedarían completamente ignoradas. En consecuencia, el modelo parecería funcionar bien según esta métrica, cuando en realidad no estaría capturando la diversidad del problema ni ofreciendo una clasificación útil para todas las clases.

Por eso, en estos casos es fundamental complementar la evaluación con métricas como *precision* y *recall* por clase, así como analizar la matriz de confusión.

Por lo tanto, también se calcularon las métricas de precisión (*precision*) y recuperación (*recall*) para cada clase utilizando la función *classification_report*. Estas métricas permiten un análisis más detallado del rendimiento del modelo por candidato. En particular:

- La precisión indica qué proporción de los discursos que el modelo asignó a un candidato realmente pertenecían a dicho candidato.
- El *recall* indica qué proporción de los discursos de un candidato fueron correctamente identificados por el modelo.

Los valores obtenidos se presentan en la Tabla 2 y muestran que el modelo logró un desempeño razonable, aunque no completamente balanceado entre los distintos candidatos. Las métricas reflejan diferencias que pueden atribuirse tanto a la distribución desigual de clases en el conjunto de entrenamiento como a características léxicas propias de cada orador. Estas métricas se relacionan directamente con la matriz de confusión (Figura 7), ya que se calculan a partir de los valores de verdaderos positivos, falsos positivos y falsos negativos observados para cada clase.

Tabla 2: Métricas de desempeño por clase del modelo Multinomial Naive Bayes sobre el conjunto de prueba.

Candidato	Precision	Recall
Joe Biden	0.76	1.00
Donald Trump	1.00	0.88
Mike Pence	0.00	0.00

En particular, en la tabla se observa que el modelo identificó correctamente todos los discursos de Joe Biden (*recall* = 1.00), aunque con una precisión más moderada (*precision* = 0.76), lo cual indica que algunos discursos de otros candidatos fueron clasificados como suyos. Para Donald Trump, la precisión fue perfecta (1.00), pero su *recall* algo menor (0.88), lo que sugiere que no todos sus discursos fueron reconocidos. Finalmente, el modelo no logró identificar correctamente ningún discurso de Mike Pence (*precision* y *recall* de 0.00), lo que podría deberse a su escasa representación en el conjunto de entrenamiento.

Complementariamente al desarrollo anterior, y previo a la búsqueda y optimización de hiperparámetros del modelo, se evaluó la *accuracy* en los casos de pretratamiento de los textos planteados en la sección 2.3 y cuya reducción de dimensionalidad por PCA se mostró en la Figura 6. Los resultados se presentan en la Tabla 3.

Tabla 3: *Accuracy* del modelo Multinomial Naive Bayes para distintas estrategias de pretratamiento de texto.

	Accuracy
a) Sin eliminación de stopwords	0.53
b) Sin ninguna limpieza previa	0.53
c) Consideración de unigramas y bigramas	0.85

A modo de comentario, se observó que en los casos (a) y (b) el modelo asignaba todos los discursos al candidato Joe Biden (*recall* = 1 y *precision* = 0.53).

Esto puede explicarse porque las *stopwords* (palabras muy frecuentes pero poco informativas) aparecen en todos los discursos sin distinción entre candidatos, pero al estar presentes en mayor cantidad en los discursos de Biden (quien cuenta con más textos del conjunto de entrenamiento), el modelo las asoció con su clase. Como resultado, estas palabras comunes dominaron el aprendizaje del modelo y redujeron su capacidad de discriminar entre candidatos, favoreciendo sistemáticamente la clase más representada (Joe Biden).

Este resultado puso en evidencia la necesidad de aplicar una limpieza adecuada al texto antes del entrenamiento del modelo. Eliminar *stopwords* y normalizar los datos (convertir a minúsculas, eliminar puntuación y caracteres irrelevantes) permite reducir el ruido y resaltar las palabras discriminatorias entre las clases, mejorando la capacidad del modelo para distinguir entre candidatos.

Por otra parte, en el caso (c), el valor de accuracy (0.85) apenas mostró una mejora respecto al caso base (0.83), y la matriz de confusión se mantuvo sin cambios. Por esto, el hiperparámetro *ngram_range* del vectorizador fue incorporado en el conjunto de parámetros considerados durante la búsqueda subsiguiente de hiperparámetros óptimos del modelo.

2.4.2. Optimización de hiperparámetros

Una forma complementaria de evaluar el rendimiento de un modelo es mediante la validación cruzada, una técnica que permite estimar la capacidad de generalización y reducir la dependencia de una única partición de los datos (de entrenamiento y prueba). Para este conjunto de datos, la validación cruzada se aplicó exclusivamente sobre el conjunto de entrenamiento, el cual se dividió en 5 subconjuntos o *folds*.

En cada iteración, uno de los subconjuntos se utilizó como conjunto de validación, mientras que los restantes sirvieron para entrenar el modelo. Este procedimiento se repitió alternando el subconjunto de validación en cada ciclo, y finalmente se calcularon los promedios de las métricas obtenidas (por ejemplo, *accuracy*) para obtener una estimación más robusta del desempeño durante el entrenamiento.

Complementario a lo anterior, cada modelo posee hiperparámetros, los cuales no se aprenden directamente a partir de los datos, sino que deben ser definidos antes del entrenamiento y posteriormente evaluados. La selección adecuada de estos valores es fundamental, ya que puede tener un impacto significativo en el desempeño final del modelo. En este trabajo, la evaluación de los hiperparámetros se realizó mediante el esquema de validación cruzada previamente descrito. Una vez definidos los valores óptimos, el modelo final fue reentrenado y evaluado sobre un conjunto de prueba independiente, no utilizado durante la validación cruzada, con el objetivo de obtener una estimación objetiva y realista de su rendimiento.

En este trabajo se analizaron los hiperparámetros del *TfidfVectorizer* y del clasificador *MultinomialNB*. A continuación, se presentan los principales hiperparámetros considerados durante el proceso de ajuste del modelo, acompañados de una breve descripción funcional:

- *ngram_range:* Especifica el rango de n-gramas que se utilizarán como características del modelo. Por ejemplo, el rango (1, 1), consideran únicamente unigramas (secuencias de una sola palabra), el rango (1, 2) unigramas y bigramas (secuencias de dos palabras), etc.
- *min_df:* Establece el umbral mínimo de frecuencia documental; una palabra debe aparecer en al menos esa cantidad de documentos para ser incluida como característica.

- *max_df:* Define el umbral máximo de frecuencia documental; si una palabra aparece en un porcentaje superior de documentos, se descarta por considerarse poco informativa o demasiado común.
- *alpha:* Parámetro de suavizado de Laplace utilizado para evitar problemas con palabras que no aparecen en ciertas clases durante el entrenamiento. Sin este suavizado, el modelo asignaría una probabilidad cero a esas palabras, lo que puede provocar que todo un conjunto de características tenga probabilidad cero y afecte negativamente la predicción. Un valor de alpha mayor a cero distribuye una pequeña probabilidad a todas las características, mejorando la robustez y la capacidad del modelo para generalizar a datos nuevos.
- *fit_prior:* Controla si el modelo considera las probabilidades previas reales de las clases, basadas en la distribución de los datos de entrenamiento. Cuando se establece en True, el modelo aprende y utiliza estas probabilidades, lo que es útil cuando las clases están desbalanceadas. Si se define como False, el modelo asume que todas las clases son igualmente probables, lo que puede ser beneficioso si se desea evitar que clases mayoritarias dominen la predicción o cuando no se confía en la distribución observada en los datos.

La búsqueda de hiperparámetros se realizó utilizando la herramienta *GridSearchCV* de *scikit-learn*, la cual explora exhaustivamente combinaciones definidas de parámetros para optimizar el rendimiento del modelo. Se construyó un *pipeline* que integra el vectorizador *TfidfVectorizer* y el clasificador *MultinomialNB*, y se definió una grilla con valores posibles para los hiperparámetros más relevantes de ambos componentes, incluyendo el rango de n-gramas ((1,1) y (1,2)), los umbrales de frecuencia mínima (1, 3 y 5) y máxima (0.9 y 1.0) de palabras, el parámetro de suavizado alpha (0.01, 0.1 y 1.0), y la consideración de probabilidades previas en el clasificador (True o Flase). La evaluación se realizó mediante validación cruzada con 5 particiones. Como métrica para seleccionar el mejor conjunto de hiperparámetros se utilizó la *accuracy* promedio entre particiones.

Los valores óptimos encontrados para los hiperparámetros considerados en la búsqueda se resumen en la Tabla 4.

Tabla 4: Hiperparámetros óptimos seleccionados para el modelo Multinomial Naive Bayes.

Hiperparametro	Valor
tfidf__ngram_range	(1,1)
tfidf__min_df	5
tfidf__max_df	0,9
alpha	0,01
fit_prior	False

Adicionalmente, si bien la optimización se realizó variando simultáneamente todos los hiperparámetros mencionados, para facilitar la visualización de los resultados de *accuracy*, en la Figura 8 se presenta únicamente la variación de esta métrica en función de los valores de *alpha* y *fit_prior*, manteniendo constantes los valores del resto de los hiperparámetros en aquellos que resultaron en un mayor rendimiento (*ngram_range* = (1,1), *min_df* = 5 y *max_df* = 0.9).

La Figura 8 corresponde a un diagrama de violín, que combina un boxplot con una estimación de la densidad de probabilidad de los datos. En este tipo de gráfico, el ancho del "violín" representa la densidad de observaciones para diferentes valores de *accuracy*, permitiendo observar la distribución completa de los resultados. La línea central indica la mediana, mientras que las líneas internas muestran los cuartiles. De esta forma, es posible identificar no solo el valor central sino también la variabilidad y la forma de la distribución de la métrica bajo diferentes configuraciones de los hiperparámetros.

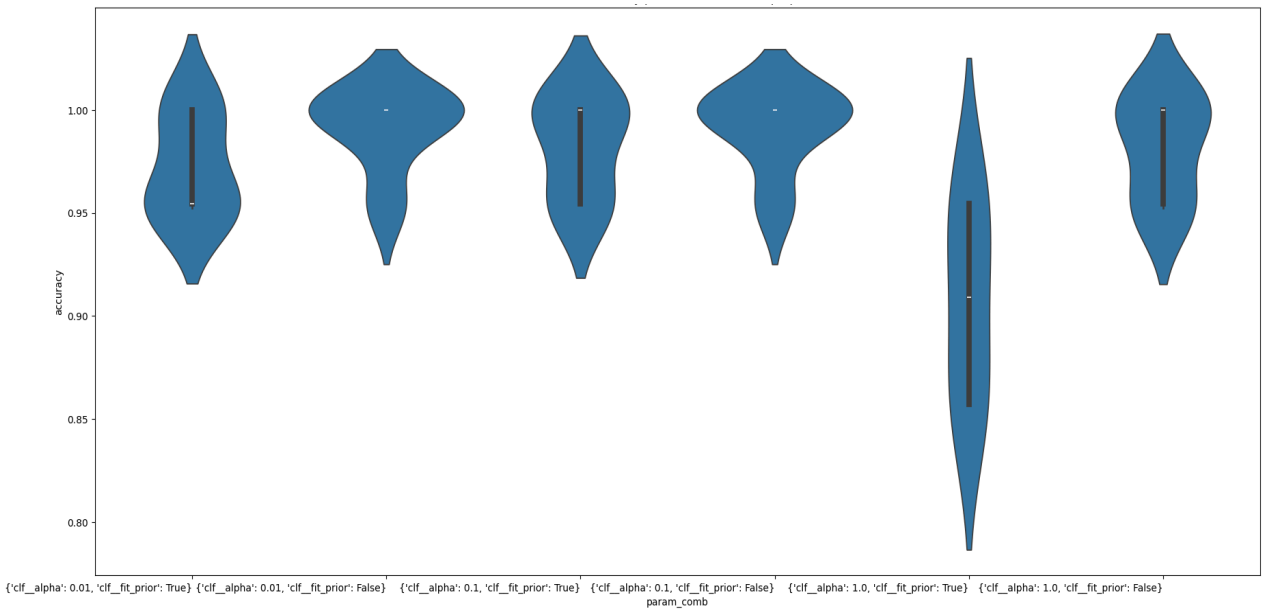

Figura 8: Rendimiento del modelo de acuerdo a la variación de hiperparametros (*alpha* y *fit_prior*).

La evaluación mediante validación cruzada reflejó un valor de *accuracy* promedio de 0.99 utilizando los valores óptimos de los hiperparámetros. Esto refleja una mejora considerable respecto al resultado obtenido previamente, que se correspondió con un *accuracy* de 0.83.

Una vez determinados los hiperparámetros que generan el mejor desempeño del modelo, se procedió a reentrenar utilizando el conjunto de entrenamiento original (obtenido mediante muestreo estratificado), y posteriormente se evaluó el modelo en el conjunto de prueba ya separado. Mientras que la búsqueda de hiperparámetros se basó en la validación cruzada, que calcula métricas promedio sobre particiones del conjunto de entrenamiento, este paso permite obtener las métricas finales de evaluación y la matriz de confusión (Figura 9).

En esta matriz se observa que tanto la precisión (*precision*) como la sensibilidad (*recall*) alcanzan valores del 100 %, lo que indica que el modelo clasifica correctamente la totalidad de los discursos analizados. Esto refleja que, con los hiperparámetros ajustados y el conjunto de datos utilizado, el modelo logró una capacidad de generalización excelente para el problema planteado.

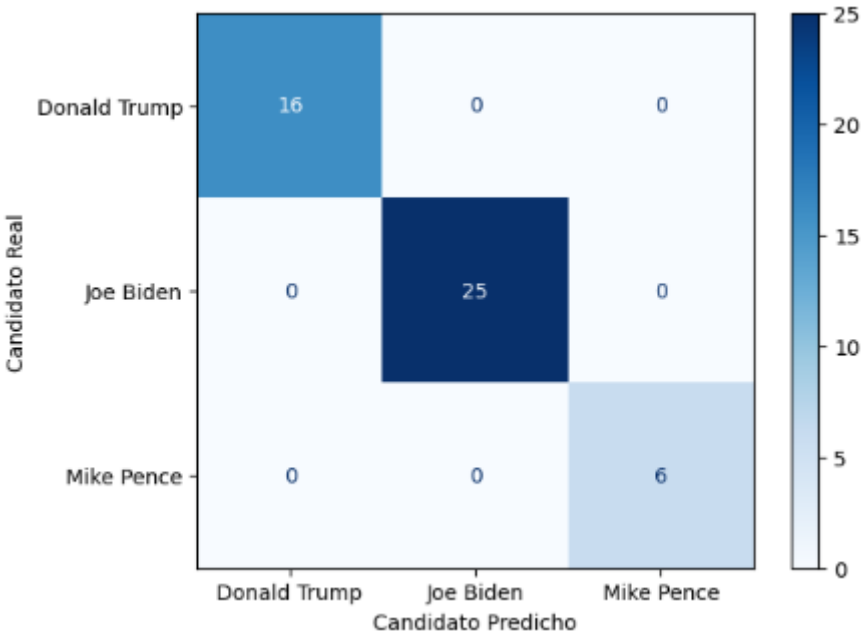

Figura 9: Matriz de confusión resultante del modelo Multinomial Naive Bayes con los hiperparametros óptimos.

Por otra parte, si bien el modelo obtenido mostró un rendimiento elevado, es importante reconocer las limitaciones asociadas al uso de representaciones basadas en bag-of-words o TF-IDF. Entre ellas se destaca la pérdida de contexto y del orden de las palabras, lo cual afecta la interpretación en casos de polisemia, así como la incapacidad de captar relaciones semánticas entre términos con significados similares. Además, la transformación de cada palabra en una dimensión del vector genera representaciones de alta dimensionalidad y baja densidad, lo que puede incrementar la complejidad y favorecer el sobreajuste.

Para mitigar estas limitaciones, se aplican estrategias como el uso de n-gramas, que permiten capturar parte del contexto. No obstante, en este caso se optó por unigramas, ya que ofrecieron el mejor desempeño al ajustar el hiperparámetro *ngram_range*. Dado que el objetivo era predecir el autor del discurso, los términos correspondientes a palabras individuales resultaron suficientes.

Asimismo, se implementó la lematización para reducir la variabilidad léxica y compactar el vocabulario, mejorando la representación semántica. Sin embargo, este proceso podría introducir cierto sesgo, al unificar términos que podrían ser distintivos entre candidatos, afectando así la capacidad del modelo para capturar diferencias sutiles en el lenguaje empleado.

2.4.3. Modelos de clasificación alternativos

Luego de haber entrenado y optimizado un modelo Multinomial Naive Bayes, cuyo desempeño fue altamente satisfactorio, se procedió a explorar otros algoritmos de clasificación para comparar su rendimiento sobre el mismo conjunto de datos. Esto se debe a que distintos modelos pueden comportarse de manera diferente según las características del problema, y es útil contrastar sus resultados para obtener una visión más completa. En este contexto, se evaluaron dos modelos adicionales: Logistic Regression (regresión logística) y Linear Support Vector Classifier (LinearSVC).

El modelo de Logistic Regression es un clasificador lineal que estima la probabilidad de pertenencia a una clase en función de una combinación lineal de las variables de entrada, a través de una función logística (sigmoidea). Esta característica lo hace interpretable y eficiente. En problemas multiclase como el presente, utiliza estrategias como *one-vs-rest*, entrenando un clasificador binario por cada clase.

En una primera etapa, el modelo de regresión logística se entrenó directamente a partir de las representaciones TF-IDF previamente obtenidas, siguiendo un procedimiento análogo al aplicado para el modelo Multinomial Naive Bayes. Los resultados iniciales de este entrenamiento se presentan en la Tabla 5, y reflejan un rendimiento razonable, aunque inferior al observado para el modelo optimizado de Naive Bayes.

A continuación, se exploró la posibilidad de mejorar su desempeño mediante una búsqueda conjunta de hiperparámetros, utilizando *GridSearchCV* y un *pipeline* que integraba tanto el vectorizador *TfidfVectorizer* como el clasificador Logistic Regression.

Nuevamente, los parámetros evaluados para el vectorizador fueron *ngram_range*, *min_df* y *max_df*. Respecto al clasificador, se exploraron los siguientes hiperparámetros:

- *C*: controla la fuerza de regularización, es decir, cuánto se penalizan los coeficientes grandes del modelo para evitar el sobreajuste al conjunto de entrenamiento. Valores más bajos implican mayor regularización.
- *penalty*: define el tipo de regularización aplicada. Se consideraron las penalizaciones L1, que puede llevar algunos coeficientes exactamente a cero favoreciendo modelos más simples, y L2, que distribuye la penalización de manera más uniforme.
- *solver*: especifica el algoritmo de optimización utilizado para ajustar el modelo; su elección depende del tipo de penalización y del tamaño del conjunto de datos.

Se eligió el *solver 'liblinear'*, que es compatible con ambas penalizaciones L1 y L2.

Este proceso permitió evaluar distintas combinaciones de parámetros y seleccionar aquellos que maximizaban el valor de *accuracy* promedio. Los valores óptimos de los hiperparámetros se presentan en la Tabla 5.

Los resultados finales de reentrenamiento del modelo en estas condiciones se presentan en la Tabla 6 y la Figura 10 (matriz de confusión).

Tabla 5: Hiperparámetros óptimos seleccionados para el modelo Logistic Regression.

Hiperparametro	Valor
tfidf__ngram_range	(1,1)
tfidf__min_df	3
tfidf__max_df	1,0
C	10
penalty	L2

Tabla 6: Métricas de desempeño por clase del modelo Logistic Regression sobre el conjunto de prueba.

	Previo a optimización de hiperparámtros		**Posterior a optimización de hiperparámtros**	
	Precision	**Recall**	**Precision**	**Recall**
Donald Trump	1,00	0,94	1,00	1,00
Joe Biden	0,93	1,00	0,96	1,00
Mike Pence	1,00	0,83	1,00	0,83
Accuracy	0,96		0,98	

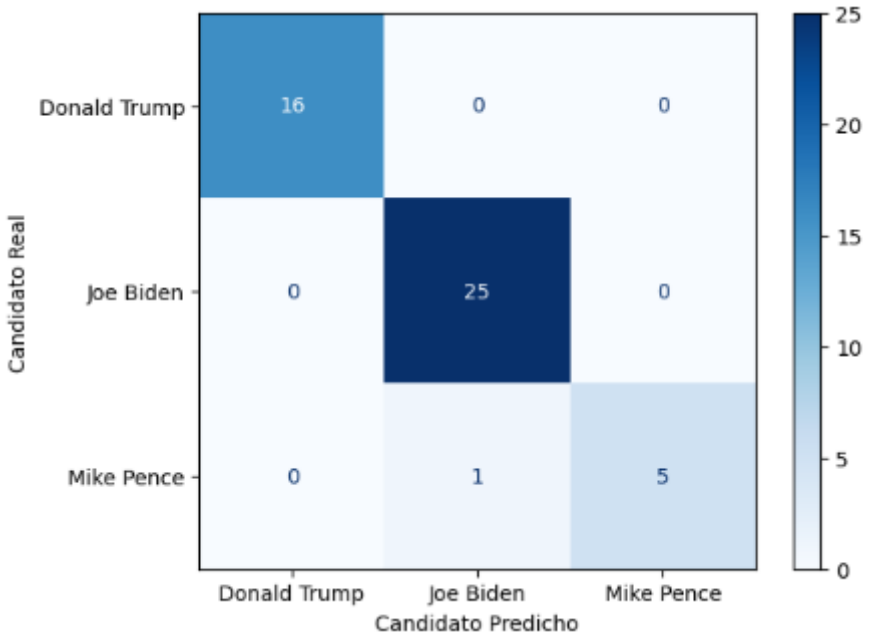

Figura 10: Matriz de confusión del modelo Logistic Regression sobre el nuevo conjunto de prueba y optimizando hiperparámetros relevantes.

El modelo de Logistic Regression alcanzó un desempeño muy alto tras la búsqueda de hiperparámetros, con una *accuracy* del 98 % y valores de *precision* y *recall* también elevados en las tres clases analizadas. En particular, se observa un excelente rendimiento en la clasificación de discursos de Donald Trump y Joe Biden (*recall* del 100 % en ambos casos). En cambio, el modelo mostró una leve caída para Mike Pence (*recall* = 0.83), lo que indica que un discurso suyo fue erróneamente clasificado como perteneciente a otra clase (en este caso a Joe Biden).

Si bien este resultado sigue siendo competitivo, representa una ligera disminución respecto al desempeño perfecto alcanzado por el modelo Multinomial Naive Bayes, el cual clasificó correctamente todos los discursos del conjunto de prueba. Por lo tanto, aunque Logistic Regression constituye una alternativa robusta y generalizable, Naive Bayes constituyó un modelo más preciso para este conjunto de datos.

Por su parte, el modelo LinearSVC forma parte de la familia de los Support Vector Machines (SVM), y tiene como objetivo encontrar un hiperplano que separe las clases de forma lineal maximizando la distancia entre los puntos de cada clase más cercanos a ese hiperplano. Esta propiedad lo vuelve eficaz en problemas de alta dimensionalidad, como los que surgen en tareas de clasificación de texto, donde cada palabra o n-grama puede ser una dimensión.

Al aplicarlo al conjunto de discursos, de manera completamente análoga a los modelos anteriores, este clasificador alcanzó un rendimiento idéntico al del modelo Multinomial Naive Bayes ajustado y optimizado (mejores hiperparámetros). Se obtuvo un valor de *accuracy* del 100 % sobre el conjunto de prueba, y tanto la *precision* como el *recall* alcanzaron valores perfectos para las tres clases. La correspondiente matriz de confusión (Figura 9) confirma que todos los discursos fueron correctamente clasificados, sin errores. Estos resultados posicionan al modelo LinearSVC como una alternativa más eficaz que Multinomial Naive Bayes, ya que permite alcanzar un rendimiento excelente sin necesidad de ajustar los hiperparámetros propios del modelo, ya que los valores por defecto fueron suficientes para lograr una clasificación óptima.

En tareas de clasificación de texto con representaciones TF-IDF, LinearSVC suele superar a MultinomialNB, especialmente cuando el vocabulario es extenso y existen correlaciones entre términos relevantes, como se observó en este trabajo. No obstante, Naive Bayes continúa siendo una alternativa útil debido a su simplicidad y velocidad de entrenamiento. Por otro lado, aunque Logistic Regression permite mayor flexibilidad mediante el ajuste de hiperparámetros, en este caso no logró superar en rendimiento a los otros dos modelos.

2.5. Análisis del desbalance de clases y su impacto en el modelo

Con el objetivo de explorar el impacto del desbalance de clases sobre el rendimiento del modelo, se modificó la composición del conjunto de datos original. Inicialmente, la distribución de discursos por candidato ya presentaba cierta desigualdad (Joe Biden contaba con 81 discursos, Donald Trump con 54, y Mike Pence con solo 20).

Para profundizar en el análisis, se decidió sustituir al candidato Mike Pence por Barack Obama, con una cantidad aún más reducida de discursos (únicamente 4). Este nuevo escenario acentuó el desbalance entre clases, permitiendo evaluar cómo este afecta al comportamiento del modelo y qué limitaciones puede presentar una métrica global como el valor de *accuracy* en contextos de alta desproporción. Además, se consideraron indicios visibles en el mapeo con PCA.

Habiéndose aplicado el mismo procedimiento de pretratamiento y limpieza que en los casos anteriores, el set de datos se reduce a 139 discursos: 81 de Joe Biden, 54 de Donald Trump y 4 de Barack Obama.

Como primer paso, se aplicó un muestreo estratificado, siguiendo el mismo criterio previamente utilizado para el conjunto de datos original. De esta manera, se obtuvieron 97 discursos para el conjunto de entrenamiento (incluyendo 3 de Barack Obama) y 42 discursos para el conjunto de evaluación (con solo 1 de Obama).

El procedimiento de vectorización del texto mediante Bag of Words (BoW) y la posterior transformación con TF-IDF se mantuvo sin cambios respecto al aplicado en las etapas anteriores.

Sin embargo, dada la nueva composición del conjunto de datos, se observó un cambio en las palabras más relevantes, o sea aquellas con valores de TF-IDF mayores que cero, para el mismo discurso (Joe Biden el 16 de octubre de 2020 en Michigan). Esta variación puede deberse a la inclusión de un nuevo orador con un estilo lingüístico distinto y a la marcada reducción en la cantidad de discursos disponibles para algunas clases, lo que afecta el cálculo de las frecuencias relativas en el corpus. Estos resultados se presentan en la Figura 11 como nube de palabras.

Figura 11: Nube de palabras con valor de TF-IDF mayor a cero del discurso de campaña de Joe Biden del 16 de octubre de 2020 en Michigan.

Asimismo, la representación de los discursos en el espacio reducido mediante PCA también se vio afectada por la nueva composición del conjunto (Figura 12). En particular, los discursos de Barack Obama se solaparon con los de ambos candidatos, lo que dificultó la separación visual entre clases.

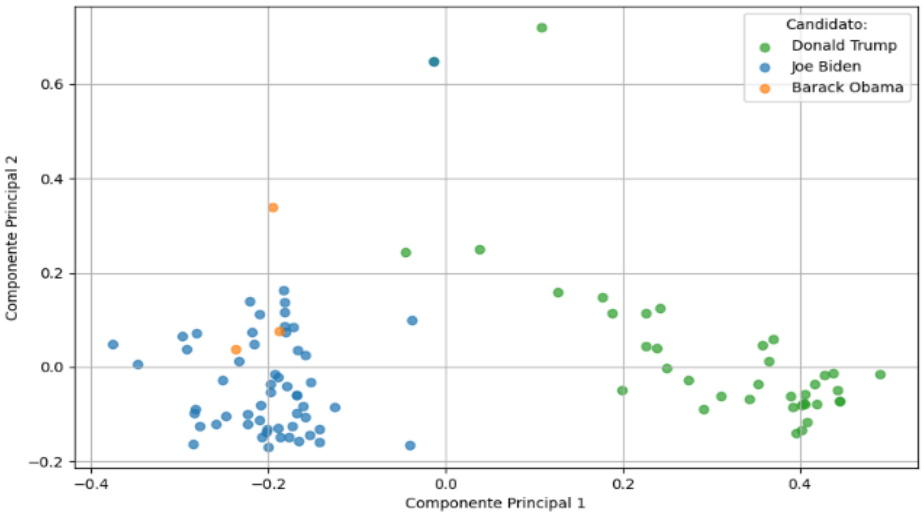

Figura 12: Proyección del conjunto de entrenamiento sobre las dos primeras componentes principales obtenidas mediante PCA, aplicadas a los vectores TF-IDF.

La varianza explicada acumulada debida a la consideración de dos componentes principales se mantuvo prácticamente incambiada respecto al caso original, en aproximadamente 16%.

A continuación, en base a las transformaciones del texto ya mencionadas, el entrenamiento del modelo se realizó de forma análoga a los casos anteriores. En este nuevo escenario, el modelo alcanzó un *accuracy* de 0.93 (previo a la optimización de hiperparámetros) sobre el conjunto de evaluación. La correspondiente matriz de confusión se muestra en la Figura 13.

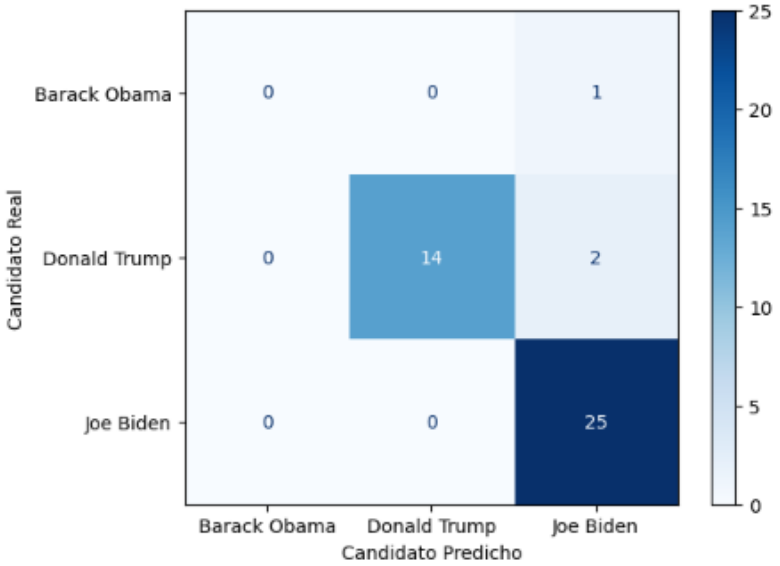

Figura 13: Matriz de confusión del modelo Multinomial Naive Bayes sobre el nuevo conjunto de prueba.

A partir de esta matriz de confusión, se observa que el modelo nunca predice la clase correspondiente a Barack Obama, asignando sus discursos a Joe Biden en todos los casos. A pesar de esta falla, el valor de *accuracy* se mantiene relativamente alto (0.93), lo que pone en evidencia una vez más las limitaciones de esta métrica en contextos de clases desbalanceadas.

Esta situación indica que el modelo no logra aprender patrones representativos de la clase Barack Obama, probablemente debido a la escasa cantidad de ejemplos disponibles (solo 4 discursos en el total del set de datos). Además, se confirma una tendencia a confundir discursos de oradores demócratas entre sí, mientras que los de Donald Trump son mejor diferenciados, posiblemente por su vocabulario más distintivo.

Finalmente, se realizó una búsqueda de hiperparámetros utilizando el mismo procedimiento aplicado para el set de datos original. Se consideraron los mismos hiperparámetros del vectorizador (*ngram_range*, *min_df* y *max_df)* y del modelo (*alpha* y *fit_prior*), con el objetivo de encontrar la combinación que optimice el rendimiento.

Cabe aclarar que, debido a la escasa cantidad de discursos disponibles para Barack Obama en el conjunto de entrenamiento, la validación cruzada se realizó con solo 3 pliegues (*folds*) para asegurar que cada clase estuviera representada en todas las particiones.

Los resultados de esta búsqueda se presentan en la Tabla 7, donde se detallan los valores óptimos seleccionados. Con estos parámetros, el modelo fue reentrenado y evaluado nuevamente sobre el conjunto de prueba, alcanzando un accuracy de 0.96. La nueva matriz de confusión se muestra en la Figura 14.

Tabla 7: Hiperparámetros óptimos seleccionados para el modelo Multinomial Naive Bayes.

Hiperparámetro	Valor
ngram_range	(1,2)
min_df	1
max_df	0,9
alpha	0,01
fit_prior	True

A partir de esta tabla se observa que los hiperparámetros *ngram_range*, *min_df* y *fit_prior* presentan valores óptimos diferentes a los hallados para el conjunto de datos anterior. Es particularmente interesante remarcar la reducción en el valor de *min_df*, que define la cantidad mínima de discursos en los que una palabra debe aparecer para no ser descartada como irrelevante. Esta disminución es consecuencia directa de la escasa cantidad de discursos disponibles para Barack Obama, lo que hace necesario conservar palabras menos frecuentes para no perder información potencialmente útil para la clasificación de esta clase.

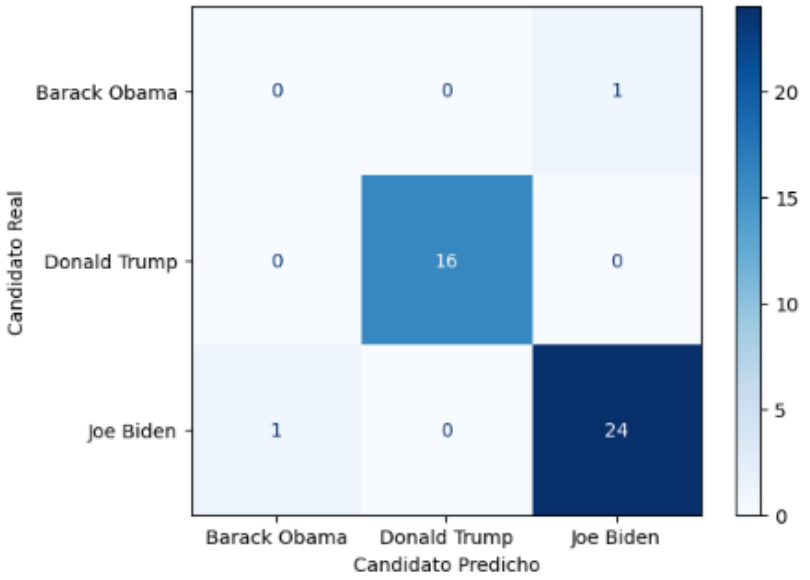

Figura 14: Matriz de confusión del modelo Multinomial Naive Bayes sobre el nuevo conjunto de prueba y optimizando hiperparámetros relevantes.

La matriz de confusión muestra que la clase correspondiente a Barack Obama continúa siendo clasificada de forma problemática: de los dos discursos asociados a esta clase en el conjunto de evaluación, uno fue erróneamente clasificado como de Joe Biden (perteneciente realmente a Barack Obama), y el otro fue el único caso en que el modelo predijo la clase Barack Obama, aunque en realidad correspondía a un discurso de Joe Biden.

Este comportamiento puede deberse a la escasa representación de Obama en el conjunto de entrenamiento, así como al hecho de que uno de los cuatro discursos está compartido por ambos oradores (ambos participan del mismo y parte del preprocesamiento incluía la mutua asignación de los discursos compartidos), lo que puede haber introducido ambigüedad en los datos. Además, dado que ambos pertenecen al mismo partido político, es esperable cierta similitud léxica y temática entre sus discursos, lo que también podría contribuir a la confusión del modelo.

En suma, este análisis permitió evidenciar cómo una desproporción aún mayor en la cantidad de discursos por clase afecta negativamente la capacidad del modelo para identificar correctamente oradores con muy poca representación, incluso cuando el valor de *accuracy* se mantiene elevado.

Esto refuerza la importancia de contar con conjuntos de datos balanceados y representativos para una clasificación robusta. En este contexto, podría resultar útil considerar técnicas como el sobre-muestreo de clases minoritarias o el submuestreo de clases mayoritarias para mitigar los efectos del desbalance. Estas estrategias permiten equilibrar la representación de clases durante el entrenamiento y podrían mejorar la capacidad del modelo para reconocer categorías subrepresentadas como la de Barack Obama, sin necesidad de modificar la arquitectura del modelo ni las transformaciones aplicadas al texto.

4. Conclusión

En este trabajo se implementó el aprendizaje automático supervisado con el fin de clasificar discursos de candidatos presidenciales. A lo largo del desarrollo, se aplicaron distintas etapas: preprocesamiento del texto, vectorización mediante Bag of Words y TF-IDF, y entrenamiento y evaluación de modelos. El modelo final alcanzó una exactitud del 100 %, un valor poco frecuente en contextos reales, posiblemente influido por el tamaño acotado del conjunto de datos o la similitud entre los subconjuntos de entrenamiento y prueba. Además, si bien el análisis mediante PCA evidenció una separación general entre clases, también mostró que algunos individuos no se encuentran claramente diferenciados en el espacio.

Más allá de los resultados finales obtenidos, el trabajo permitió identificar aspectos clave que enriquecen el proceso de aprendizaje y son relevantes para futuros desarrollos. En primer lugar, se evidenció el impacto del preprocesamiento del texto sobre el rendimiento del modelo. Aunque el análisis exploratorio mediante Análisis de Componentes Principales (PCA) no mostró diferencias marcadas entre los distintos niveles de limpieza, el desempeño del modelo sí se vio afectado por la presencia de *stopwords*. En particular, su inclusión redujo el *accuracy* del modelo Multinomial Naive Bayes. Esto puso de manifiesto la importancia de eliminar *stopwords* y normalizar el texto (uso de minúsculas, eliminación de puntuación y caracteres irrelevantes) para reducir el ruido y resaltar las palabras verdaderamente discriminativas entre clases.

Asimismo, se destacó la influencia del desbalance de clases en el desempeño del modelo. En dos casos particulares (Mike Pence en el primer conjunto de datos y Barack Obama en el segundo) se observó que las clases subrepresentadas no fueron clasificadas por el modelo Multinomial NB, a pesar de que el *accuracy* fue bueno (0,83 y 0,93, respectivamente). Este resultado pone de manifiesto la importancia de no limitar la evaluación al valor global de *accuracy*, ya que esta métrica puede ocultar deficiencias en la predicción de clases minoritarias. En su lugar, es fundamental analizar la matriz de confusión y los reportes detallados de *precision* y *recall* por clase. De todos modos, tras la optimización de hiperparámetros, se logró mejorar significativamente la capacidad del modelo para identificar estas clases. Esta experiencia refuerza la relevancia de un ajuste adecuado de los hiperparámetros, tanto del modelo como del método de vectorización, para mejorar la capacidad predictiva y garantizar un desempeño más equilibrado.

Cabe destacar que, en estos casos, el Análisis de Componentes Principales (PCA) resulta especialmente útil, ya que permite visualizar la distribución de las clases en el espacio de características y obtener información relevante sobre la estructura subyacente del conjunto de datos. En particular, facilita evaluar si una clase subrepresentada posee características lo suficientemente distintivas como para ser correctamente clasificada, o si su baja representación la hace quedar solapada dentro del agrupamiento de otra clase.

Finalmente, al analizar diferentes modelos se identificó que el modelo LinearSVC superó al MultinomialNB, en línea con lo esperado ya que el trabajo es sobre clasificación de texto con representaciones TF-IDF. No obstante, Naive Bayes continúa siendo una alternativa útil debido a su simplicidad y velocidad de entrenamiento. Adicionalmente, al entrenarlo con los hiperparametros se obtuvo un modelo muy satisfactorio. Por otro lado, aunque Logistic Regression permite mayor flexibilidad mediante el ajuste de hiperparámetros, en este caso no logró superar en rendimiento a los otros dos modelos.

En conclusión, este trabajo no solo permitió alcanzar un alto desempeño en la clasificación de discursos, sino que también brindó valiosas lecciones sobre la importancia del preprocesamiento, la selección de métricas adecuadas y la optimización de modelos. Estos aprendizajes serán fundamentales para enfrentar desafíos más complejos en el análisis del lenguaje natural.