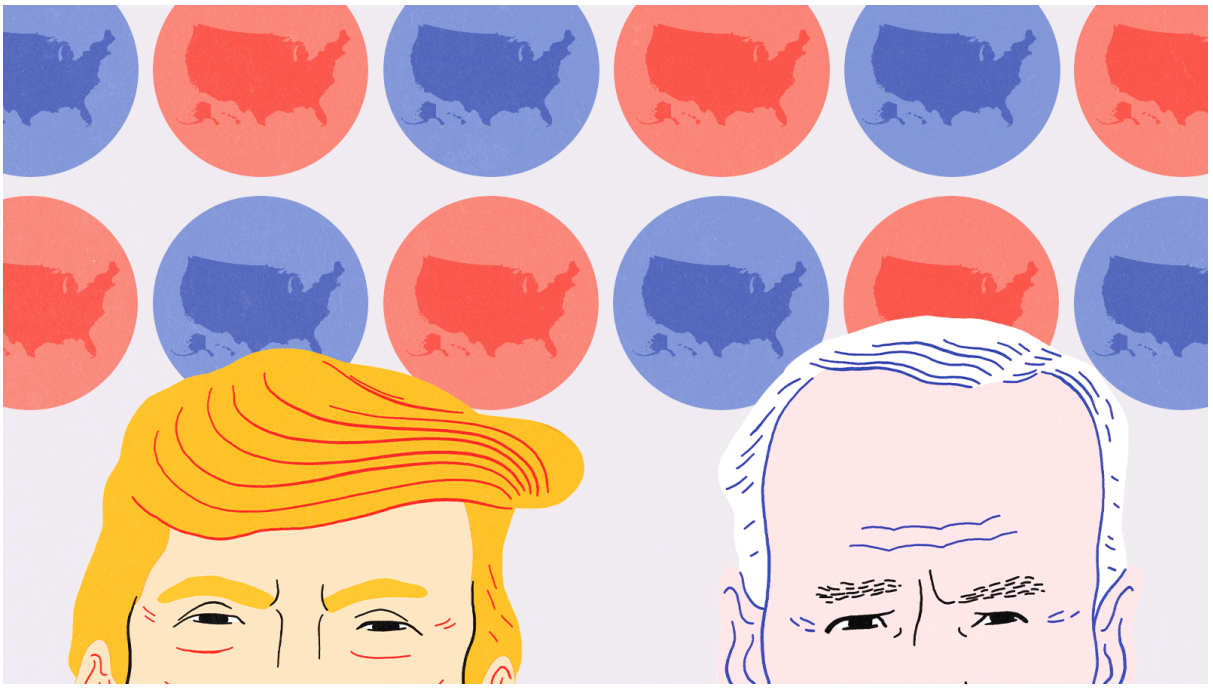
FACULTAD DE INGENIERÍA | UNIVERSIDAD DE LA REPÚBLICA URUGUAY

INTRODUCCIÓN A LA CIENCIA DE DATOS

TAREA 2



Fuente imagen: abcNEWS

GRUPO 23 - Valentina Cornelius, Santiago Quinteros

JUNIO 2025

1. Introducción

La Tarea 1 se centra en el estudio de una base de datos abierta con discursos de políticos en el marco de las elecciones de Estados Unidos 2020. En la Tarea 2 se profundiza el análisis de la base de datos.

2. Objetivos

Tarea 1: Tratamiento de datos de una base de datos abierta. Estudio de discursos en el tiempo, conteo de palabras y menciones cruzadas entre candidatos, para los 5 candidatos con más discursos en la base de datos.

Tarea 2: Uso de técnicas de procesamiento de lenguaje natural. Entrenamiento y evaluación de modelos.

3. Datos

Los datos se extraen del archivo “us_2020_election_speeches.csv”, generando un DataFrame (df) con 269 filas (cantidad de discursos) y 6 columnas (atributos). De estas columnas, las que destacan para alcanzar los objetivos establecidos son: “speaker” (candidato/a, por practicidad se utilizará el término “candidato” aunque también hay candidatas), “date” (fecha) y “text” (texto del discurso).

En una primera explotación de los datos se detecta que hay 3 discursos (1.1 %) donde no hay speaker (datos faltantes en columna “speaker”). Otros problemas detectados en los datos sobre los nombres de los candidatos son los siguientes:

- 14 discursos (5.2 %) tienen candidatos indefinidos: ‘???’, ‘Multiple Speakers’, ‘Democratic Candidates’;
- 14 discursos (5.2 %) tienen candidatos combinados, por ejemplo: ‘Joe Biden, Kamala Harris’.

Para el procesamiento de los datos se decide no considerar los discursos donde no se definen los nombres de los candidatos (datos faltantes e indefinidos, que representan un 6.3 % del total). Los otros casos mencionados se procesan, replicando los discursos y dejando un sólo nombre. El contenido de los discursos se corrige para que coincida con el candidato según se explicará a continuación.

Explorando los discursos (columna “text”), se observan los siguientes problemas:

- los datos no contienen solamente el texto de un discurso, sino que incluyen, para cada intervención, el nombre de la persona (o una referencia genérica) y el minuto en el que se expresa;
- el texto del discurso asignado a un candidato particular puede presentar intervenciones de otras personas.

Que se repita el nombre del candidato no es un problema si se tiene en cuenta al interpretar los resultados del conteo de palabras. El problema es que aparezcan discursos de otras personas: se considerarían palabras que no corresponden. Para solucionarlo, se filtran los diálogos manteniendo solamente los que se corresponden con el “speaker”. Algo a destacar

en este punto, es que los candidatos a veces aparecen con otros nombres en los textos de los discursos. Por lo tanto, antes del filtrado, se uniformiza la forma en la que se hace referencia al “speaker” en los discursos, identificando “sinónimos”. La Tabla 1 presenta las alternativas consideradas para los candidatos con más discursos (ver Top 5).

Tabla 1. Alternativas de denominación de candidatos/as

Nombre de candidato/a	Alternativas al nombre
Donald Trump	President Trump, President Donald J. Trump, Donald J. Trump
Joe Biden	Vice President Biden
Kamala Harris	Senator Harris
Mike Pence	Vice President Mike Pence

Luego de aplicado el tratamiento de los datos descrito, resulta una base final de 284 filas (discursos), con una nueva columna “Phrases”, en la cual solamente hay discursos del candidato identificado en dicha fila.

4. Análisis Descriptivo

Se realiza un análisis descriptivo de la base de datos, considerando los 5 candidatos con más cantidad de discursos.

4.1. Top 5 candidatos

La Figura 1 muestra los 5 candidatos con más discursos.

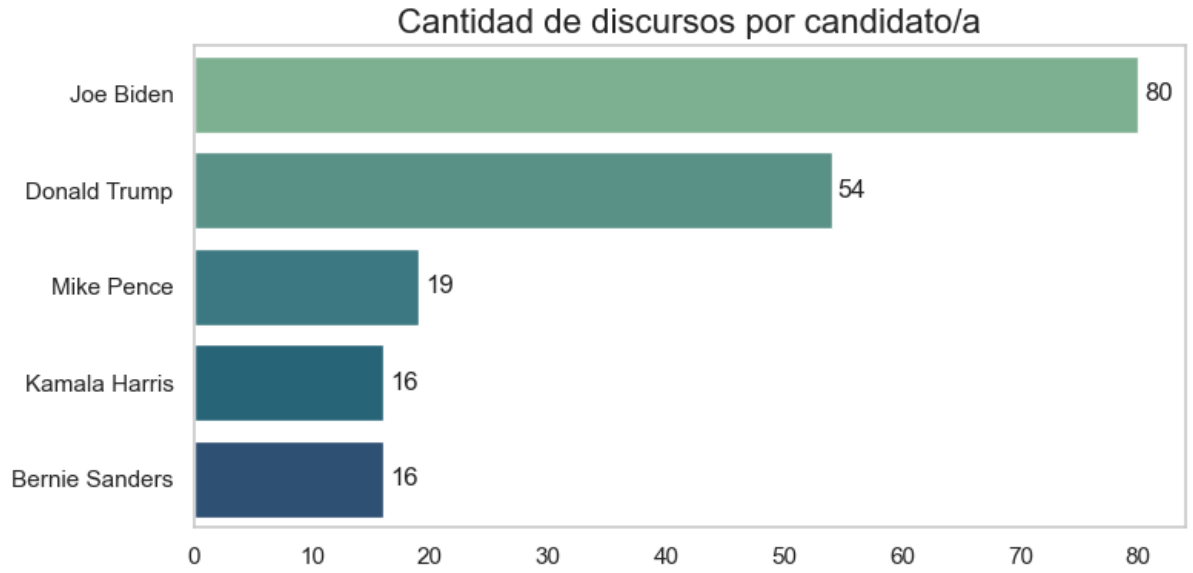


Figura 1. Top 5 de candidatos con más discursos en la base de datos analizada

Los dos candidatos con mayor cantidad de discursos son Joe Biden y Donald Trump, ambos candidatos por la presidencia de Estados Unidos en el año 2020. Luego, le siguen Mike Pence, candidato a vicepresidencia por Trump, y Kamala Harris, candidata a la vicepresidencia por Biden. Finalmente, Bernie Sanders es otro de los 5 candidatos con más discursos, quien también participó por la presidencia en la parte temprana de las elecciones por el partido demócrata.

4.2. Discursos en el tiempo

Para poder estudiar cómo se distribuyen los discursos en el tiempo, se realizó la conversión de los datos de fecha (columna “date”) a un formato adecuado.

La Figura 2 muestra la distribución de los discursos de los candidatos a lo largo del tiempo. Se observa que todos los discursos se dan en el año 2020. La mayoría de los discursos se concentran hacia los meses de setiembre y octubre, que corresponden a los meses de debates presidenciales. A excepción del candidato Berni Sanders, cuyos discursos se concentran hacia marzo. Los resultados son coherentes, ya que Sanders retiró su candidatura en el mes de abril[1].

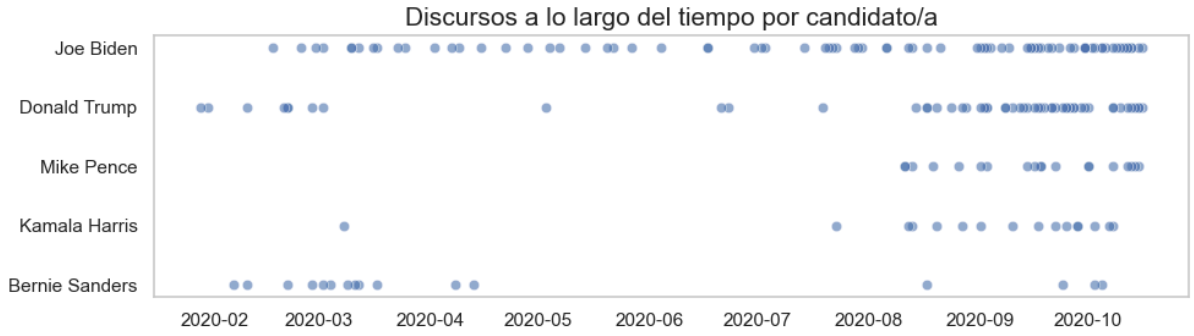


Figura 2. Distribución de los discursos de los candidatos a lo largo del tiempo

La Figura 3 muestra la cantidad de discursos por candidato por mes. Se observan picos claros en el mes de setiembre. Esto es coherente ya que ese mes se realizó el primer debate de las elecciones generales[2]. A esa altura de la campaña, Biden y Trump eran los candidatos a la presidencia, por lo que es razonable que sean los candidatos con picos más altos en cantidad de discursos ese mes. Los siguen Harris y Pence, quienes eran candidatos a la vicepresidencia.

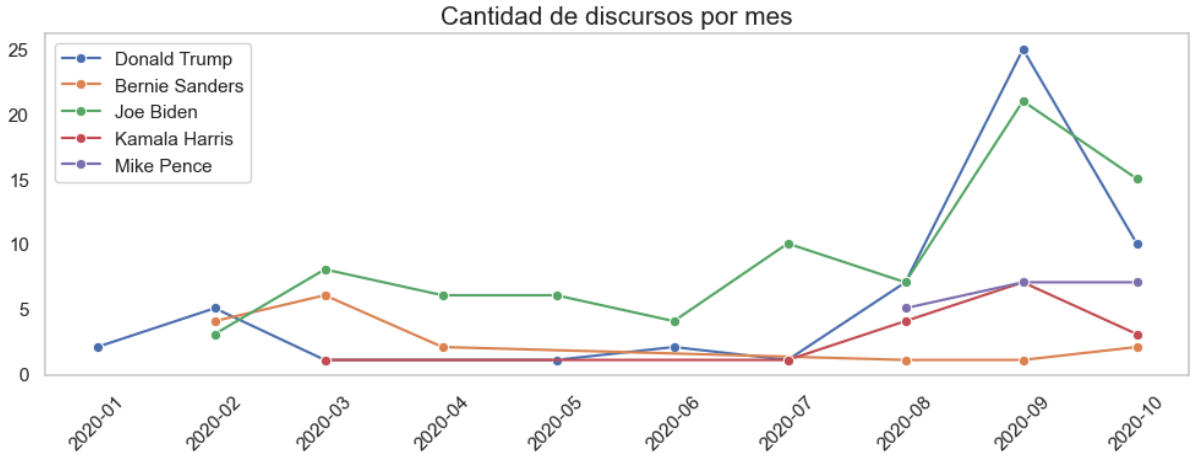


Figura 3. Variación de la cantidad de discursos por mes por candidato/a

[1] Fuente: https://edition.cnn.com/election/2020/results/president
[2] Fuente: https://apps.npr.org/elections20-primaries/

Los discursos finalizan en octubre porque las elecciones fueron el 3 de noviembre. En este gráfico se ve cómo Joe Biden fue el candidato más constante dando discursos, mientras que Donald Trump centró su estrategia en tener un pico máximo de discursos en setiembre.

4.3. Conteo de palabras

Para poder estudiar la cantidad de veces que aparece cada palabra en los discursos, es necesario normalizar el texto y eliminar los signos de puntuación.

Los cambios realizados fueron:

- Conversión del texto a minúsculas.
- Eliminación de signos de puntuación: ".", ",", “;”, ":", "?", “!” , "(", ")", "[", "]", "...", "“", "”"
- Eliminación de otros elementos: "\n", "1","2","3","4","5","6","7","8","9","0".
- Eliminación de “stop words”, por ejemplo “we’re”, “it’s”, “i’m”.

La denominación “stop words” hace referencia a palabras que se usan frecuentemente pero que no ofrecen información significativa por sí solas. Existen recursos que permiten extraer estas palabras según el idioma. En Anexo se presenta una lista de las palabras eliminadas.

Estas transformaciones se realizaron para poder obtener conclusiones más ricas de las palabras más mencionadas por los candidatos. Por ejemplo, que la palabra más repetida de un candidato sea “the” es entendible, porque es una palabra que se usa mucho, pero ese resultado no permite concluir nada en el contexto de las elecciones.

Una vez generado el listado “limpio” de palabras por candidato/a, se realizó el recuento de las diferentes palabras. La Figura 4 muestra las palabras más mencionadas por candidato/a.

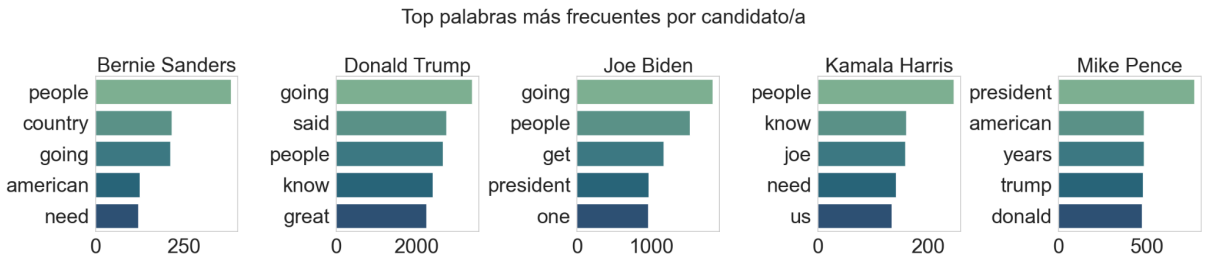


Figura 4. Palabras más mencionadas por cada candidato/a

Se observa como hay palabras comunes que repiten varios candidatos, como es el caso de “people”, “going” y “president”. Viendo el top 5 de palabras no se nota ninguna que permita identificar claramente un estilo o ideología marcada en el discurso. Tal vez los candidatos Sanders y Pence, son los que parecen tener un discurso más centrado en los estadounidenses al mencionar “american” (y “country” en caso de Sanders).

Algunas ideas para modificar esta visualización con el fin de encontrar diferencias entre partidos políticos, fechas o lugares serían:

- Agrupar los discursos según el partido político del candidato y ver las palabras más mencionadas según ese criterio.

- Agrupar los discursos según el mes, y ver las palabras que más usadas en cada uno. Se podría hacer una visualización global, o discriminando también entre partidos, para ver si hay palabras que empiezan a aparecer a partir de cierto momento en uno o ambos grupos.
- Asociar una geolocalización al discurso según el lugar donde fue realizado (en caso de ser posible, ya que hay discursos virtuales). Se podrían ver las palabras más usadas en diferentes regiones o Estados. Estas palabras podrían representarse en un mapa, o generar nubes de palabras con la forma de algún estado en particular.

En cuanto al total de palabras, Donald Trump no fue el candidato con más discursos, pero fue el que más palabras dijo, como se muestra en la Figura 5. Se entiende que es razonable que los dos candidatos a la presidencia sean los que más hayan hablado.

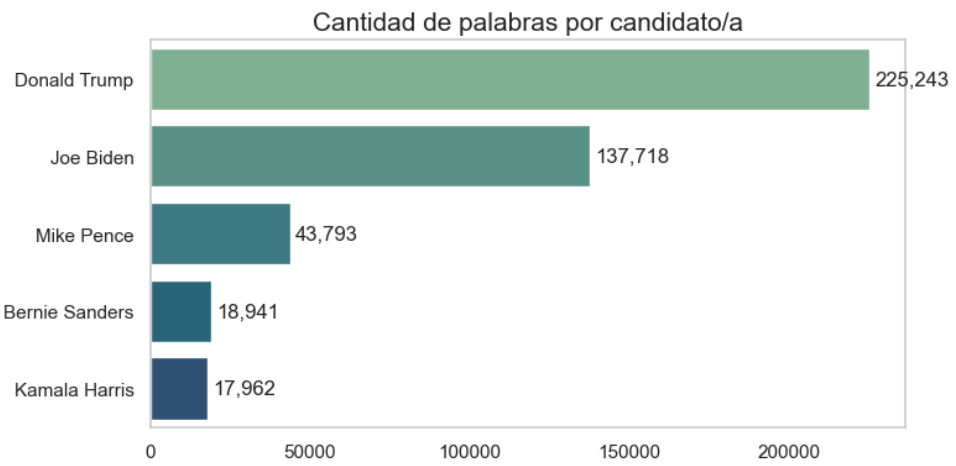


Figura 5. Cantidad de palabras por candidato/a

Se observa que, si bien cualitativamente se espera obtener resultados similares a los de otros grupos, el número de palabras obtenido seguramente difiera dadas las hipótesis asumidas al procesar los datos.

4.4. Menciones cruzadas

Para analizar las menciones cruzadas se propone la Figura 6, que presenta un mapa de calor, donde figura la cantidad de veces que se mencionaron entre candidatos.

La metodología utilizada para llegar al conteo de veces que un candidato fue nombrado, se basó en el siguiente cálculo:

- Las veces que un candidato nombra a otro por su nombre (p. ej., “Joe”), más las veces que lo nombra por su apellido (p. ej., “Biden”), menos las veces que lo nombra por su nombre completo (p. ej., “Joe Biden”).

Esta nos pareció la mejor aproximación a las menciones, debido a que dentro de los discursos hay varias formas de nombrar a otro candidato.

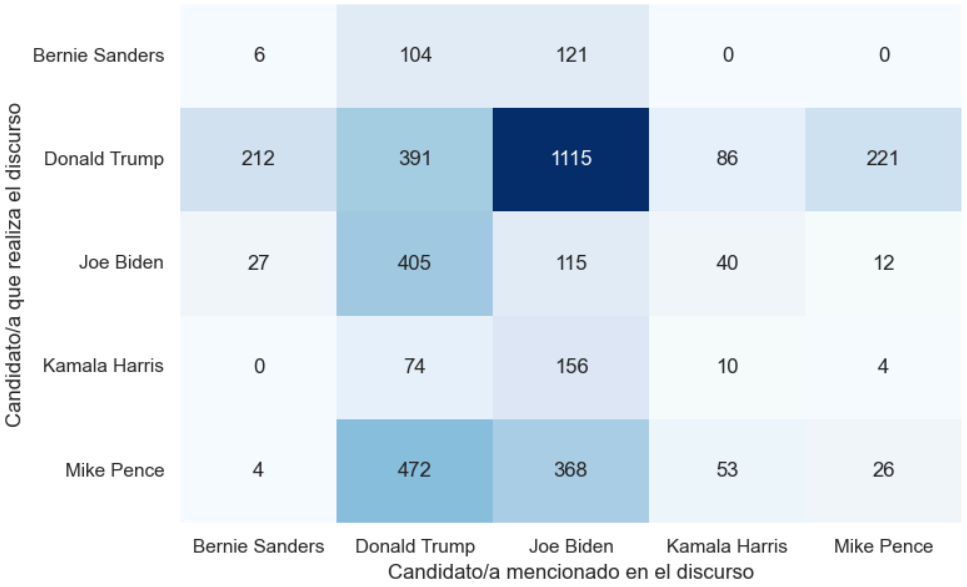


Figura 6. Cantidad de menciones cruzadas entre candidatos

Revisando los candidatos que se mencionaron entre sí, el que más menciono al resto fue Donald Trump, con un máximo de 1115 menciona a Joe Biden. Por otro lado, el candidato más mencionado fue Joe Biden, que tanto en su oposición, como miembros del partido demócrata fue uno de los más mencionados. Ambos vicepresidentes mencionaron principalmente al candidato a la presidencia de su partido, con menciones significativas a la oposición también. Bernie Sanders, solamente se mencionó al mismo, y a los candidatos a la presidencia, sin nombrar nunca a los candidatos a la vicepresidencia. Por último, Kamala Harris fue la candidata con menos menciones, siendo más nombrada por la oposición, Donald Trump, que su propio partido.

5. Procesamiento de lenguaje natural

5.1. Consideraciones respecto a los datos

En esta parte del trabajo, se considera la base de datos procesada de forma similar que para las partes anteriores, con algunos cambios.

Se mantienen las siguientes modificaciones:

- Conversión del texto a minúsculas.
- Eliminación de signos de puntuación: ".", ",", “;”, ":", "?", “!” , "(", ")", "[", "]", "...", "“", "”"
- Eliminación de otros elementos: "\n", "1","2","3","4","5","6","7","8","9","0".

Los cambios principales son:

- No se eliminan las “stop words” (p. ej., “we’re”, “the”, “of”).
- Se expanden las contracciones (p. ej., “we’re” se convierte en “we are”)

Además, se trabaja con los 3 candidatos con más discursos: Joe Biden, Donald Trump y Mike Pence (ver Figura 1).

5.2. Conjuntos de entrenamiento y testeo

Se particiona la base en un conjunto de test (testeo) del 30 % del total y otro de dev (desarrollo) utilizando muestreo estratificado. Esta técnica de desarrollar modelos únicamente con datos de entrenamiento es crucial para detectar overfitting, y evaluar en base a una simulación real, donde el modelo se enfrenta a datos nuevos que no ha visto antes. En la Figura 7 se verifica que el balance de discursos de cada candidato es similar en los conjuntos dev y test.

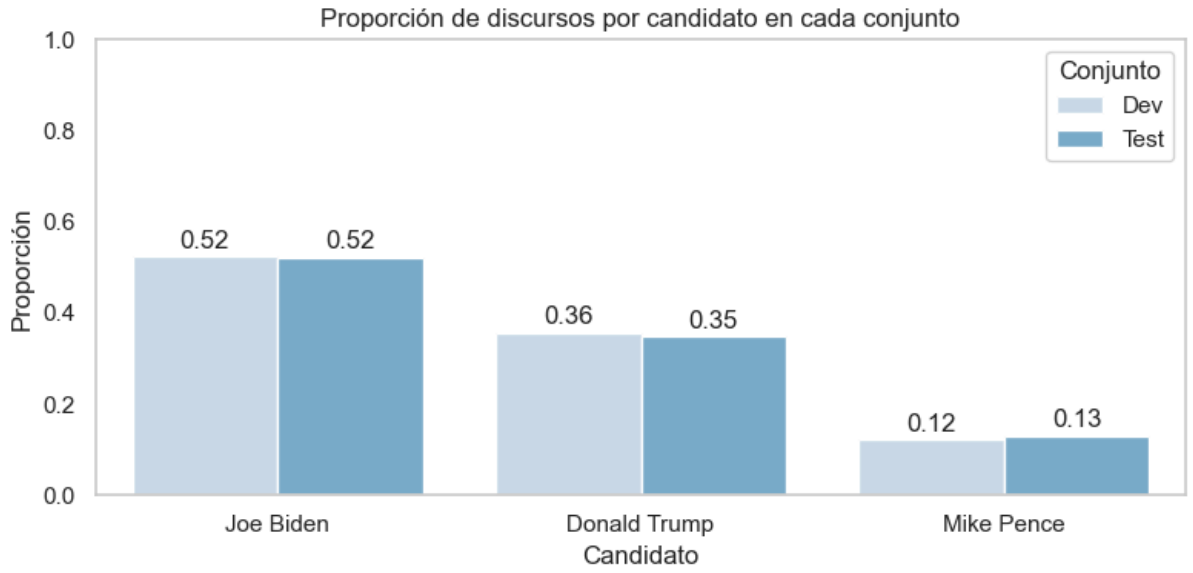


Figura 7. Porcentaje de datos por candidato para los conjuntos de dev y test

5.3. Representación numérica de texto

El texto de los discursos de cada candidato, contenidos en los conjuntos de test y dev, se transforman a la representación numérica (features) de conteo de palabras o “bag of words”. Este es un paso crucial para la tokenización de palabras y posterior entrenamiento de modelos de machine learning que aprenden a partir de matrices numéricas.

Bag of Words (BoW) es una técnica de procesamiento de texto que se utiliza en el procesamiento de lenguaje natural (NLP) y minería de texto. Consiste en representar un documento de texto como un conjunto (bag) de palabras, ignorando el orden y la estructura gramatical de las palabras en el texto.[3]

Por ejemplo, si se aplicara al párrafo anterior, el vocabulario generado contendría las palabras “procesamiento”, “lenguaje” y “texto”, con frecuencias de 2, 1 y 4 respectivamente. Si se borraran todas las palabras (del vocabulario) salvo esas 3, el vector de representación BoW sería [2,1,4].

Si el vocabulario se genera a partir de varios documentos (o párrafos, como el ejemplo anterior), la matriz resultante tendrá una columna por cada palabra única encontrada, y una fila por cada documento analizado. Luego, se vectoriza cada documento identificando las

[3] Fuente: https://keytrends.ai/es/academy/glosario/inteligencia-artificial/bag-of-words

veces que cada una de esas palabras aparece en él. Es común que palabras de algunos documentos no estén en otros; en esos casos el vector adopta valores nulos. Por eso la matriz es dispersa (sparse), es decir, es de gran tamaño y tiene pocos elementos no nulos, esto reduce el peso de almacenamiento de la matriz y el consumo de RAM al procesarla.

Aplicando esta técnica a los conjuntos de entrenamiento y testeo resultan las siguientes matrices:

- dev: 107 filas y 11620 columnas
- test: 46 filas y 11620 columnas

Eso quiere decir que el vocabulario se compone de 11620 palabras, el conjunto dev tiene 107 discursos y el test 46 discursos.

Como se mencionó anteriormente, la técnica BoW ignora el orden de las palabras en el texto. Esto puede afectar las conclusiones que puedan obtenerse del procesamiento. Existen algunas técnicas complementarias/alternativas que permiten tener en cuenta el contexto de una palabra. Una de ellas es el uso de n-gramas.

Un **n-grama** es un conjunto de n elementos consecutivos en un documento de texto, que puede contener palabras, números o símbolos (incluidos los símbolos de puntuación).[4] El modelado de n-gramas puede ser útil para enriquecer el análisis, por ejemplo, de sentimiento. No es lo mismo considerar “muy lindo” todo junto, que “muy” y “lindo” por separado, si se quiere concluir sobre sentimientos/opiniones de textos.

Por otro lado, es interesante profundizar en la relevancia de los distintos términos (sean palabras o n-gramas), ya que una palabra puede repetirse mucho pero no aportar demasiado al análisis (como ocurre con las stop words).

Term Frequency - Inverse Document Frequency (TF-IDF) es un método estadístico que también se utiliza ampliamente en el NLP, y en la recuperación de información. Mide la importancia que tiene un término dentro de un documento en relación con una colección de documentos. TF-IDF vectoriza/puntúa un término multiplicando su frecuencia (TF) por la frecuencia inversa del documento (IDF).[5]

Los cálculos realizados se resumen a continuación:

$$TF = \frac{número\ de\ veces\ que\ aparece\ e\ término\ en\ el\ documento}{número\ total\ de\ términos\ en\ el\ documento}$$

$$IDF = log\left(\frac{número\ total\ de\ documentos}{número\ de\ documentos\ que\ contienen\ el\ término}\right)$$

$$TF - IDF = TF \cdot IDF$$

[4] Fuente: https://la.mathworks.com/discovery/ngram.html
[5] Fuente: https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/

El método TF-IDF se aplica a los conjuntos dev y test para destacar las palabras más relevantes de los discursos. Las matrices mantienen las dimensiones resultantes de la aplicación de BoW.

6. Análisis exploratorio de los datos procesados

6.1. Análisis de Componentes Principales (PCA)

El **Principal Component Analysis (PCA)** es un método de reducción de dimensionalidad ampliamente utilizado en aprendizaje automático y estadística. Permite transformar un conjunto de datos compuesto por variables potencialmente correlacionadas en un conjunto de variables linealmente no correlacionadas, conocidas como componentes principales.[6]
Este modelo permite realizar un análisis exploratorio de los discursos presidenciales que previamente representamos numéricamente.

La Figura 8 muestra la distribución del conjunto de entrenamiento, considerando las 2 primeras componentes PCA sobre los vectores TF-IDF.

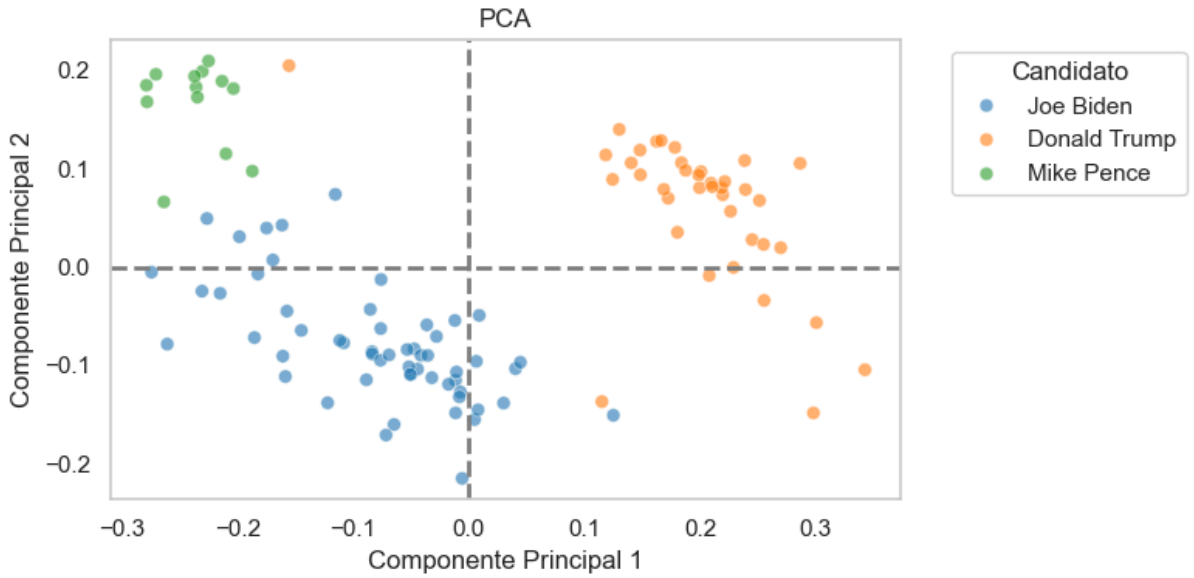


Figura 8. PCA sobre vectores TF-IDF del conjunto de entrenamiento.

Se observa que los datos se agrupan, con una separación clara de los discursos de Donald Trump respecto a los otros candidatos; los textos de Trump se ven bien representados en los valores positivos del primer componente principal (especialmente en los valores positivos de ambos componentes principales). Sin embargo, la distancia entre los discursos de Joe Biden y Mike Pence no es tan notoria: Mike Pence se ubica en valores negativos del primer componente y positivos del segundo componente, pero Biden también tiene algunos discursos en la misma ubicación que Mike. Igualmente, en términos generales, Biden concentra mayormente sus discursos en los valores negativos de los componentes, lo que se opone completamente con los discursos de Trump en ambas dimensiones; esto permite notar la clara oposición en los discursos de estos candidatos (Trump y Biden). Finalmente,

[6] Fuente: https://www.opentrain.ai/glossary/principal-component-analysis-pca

Mike parece compartir y oponerse en una de las dimensiones con ambos candidatos, dando una noción de que tiene puntos encontrados y diferentes en su discurso con Trump y Biden.

Los datos utilizados en la Figura 8 incluyen las stop words y no consideran n-gramas. Si se utiliza el filtrado de stop words (para idioma inglés), el parámetro use_idf = True (habilita la ponderación de frecuencia del documento inverso) y se consideran unigramas y bigramas (ngram_range=(1,2)), se obtiene la distribución de la Figura 9.

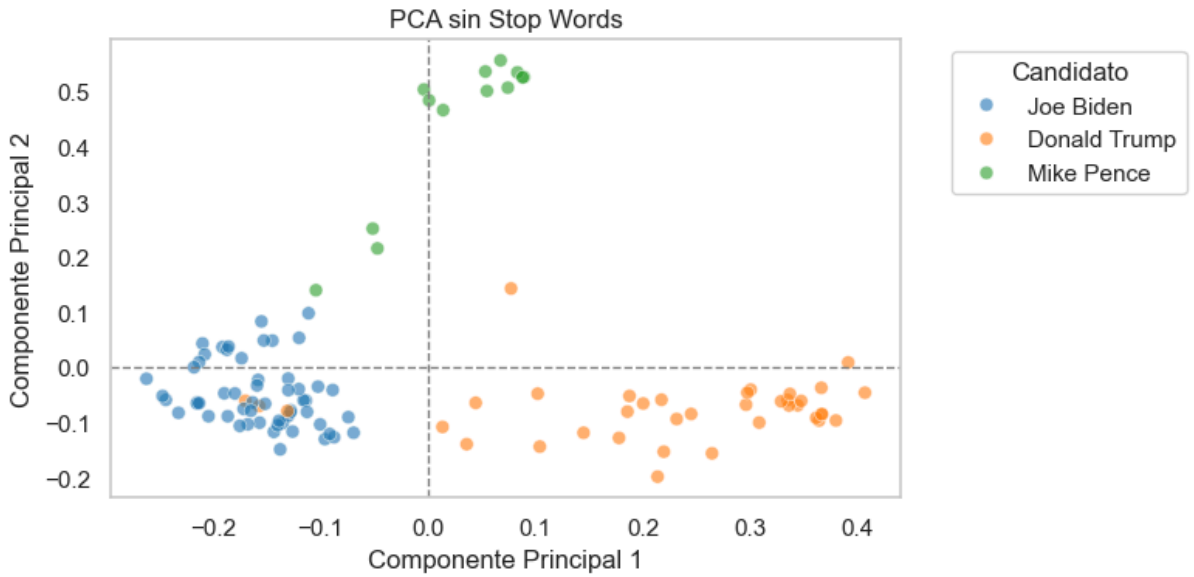


Figura 9. PCA sobre vectores TF-IDF del conjunto de entrenamiento, sin stop words, use_idf=True, ngram_range=(1,2).

Con los cambios realizados, se aprecia una nueva distribución de los datos, que aleja la mayoría de los discursos de Pence. Los discursos de Biden se agrupan más, y si bien hay textos de otros candidatos que quedan muy cerca o incluso se mezclan, en este caso se distingue mejor la presencia de 3 candidatos. Al igual que el anterior PCA, se nota una fuerte oposición entre los discursos de Trump y Biden, representada en este caso por el primer componente principal. Sin embargo, en este PCA se nota más distante Mike, con un discurso diferente al del resto de los candidatos en el segundo componente y más cercano a Trump en el primer componente.

Los resultados anteriores se generaron considerando las 2 primeras componentes principales. La Figura 10 presenta cómo varía la varianza explicada a medida que se agregan componentes principales. La proporción de varianza explicada de una componente principal representa cuánta información presente en los datos es capturada (o se pierde) por la proyección de las observaciones sobre dicho componente. La suma de las varianzas explicadas de todas las componentes principales será igual a 1.[7]

[7] Fuente: https://rpubs.com/cristina_gil/pca

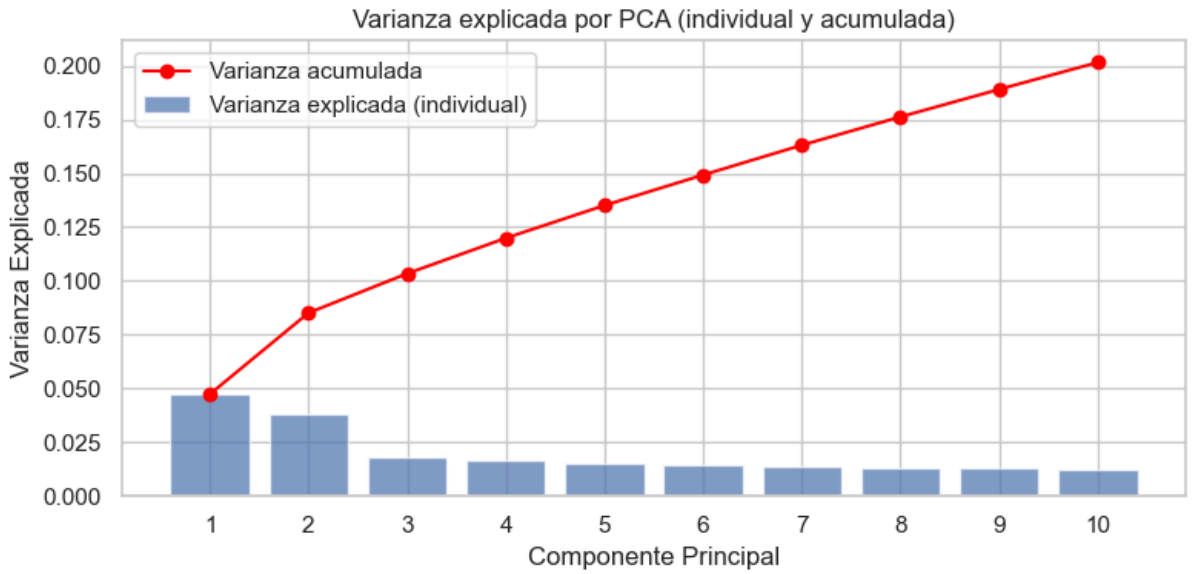


Figura 10. Variación de la varianza explicada a medida que se agregan hasta 10 componentes principales

Se observa que utilizando los primeros 10 componentes la varianza acumulada parece baja, capturando solamente el 20% de la información. Sin embargo, al graficar las 2 primeras componentes, los datos parecerían distinguirse aproximadamente bien entre los 3 candidatos, por más que la varianza acumulada en los dos ejes de inercia es menor al 10% de la total. Otra observación que surge de la Figura, es que a partir del segundo componente la variación de la varianza acumulada es aproximadamente lineal.

7. Entrenamiento y evaluación de modelos

7.1. Modelo Multinomial Naive Bayes

El algoritmo Naive Bayes es un clasificador probabilístico basado en el Teorema de Bayes. El término “naive” (ingenuo) se refiere a la forma en que el algoritmo analiza las características de una base de datos, ya que ignora la correlación entre las variables características (features).[8]

El modelo **Multinomial Naive Bayes** implementa el algoritmo Naive Bayes para datos con distribución multinomial, y es una de sus dos variantes clásicas utilizadas en la clasificación de texto.[9] Aplicado a la clasificación de texto, la distribución se parametriza en vectores θ_y con n cantidad de elementos, siendo n el tamaño del vocabulario, para cada clase y. Cada componente θ_{yi} del vector representa la probabilidad de que el término i esté presente en la clase y.

Los parámetros se estiman mediante una versión suavizada de máxima verosimilitud. El “suavizado” se logra a través del parámetro $\alpha \geq 0$; este permite tener en cuenta características que no están presentes en el conjunto de entrenamiento, evitando probabilidades nulas en cálculos posteriores. La adopción de $\alpha = 1$ se denomina suavizado de Laplace, y de $\alpha < 1$ se denomina suavizado de Lidstone.

[8] Fuente: https://datarmony.com/naive-bayes-que-es-algoritmo-ejemplos-python/#
[9] Fuente: https://scikit-learn.org/stable/modules/naive_bayes.html#multinomial-naive-bayes

Se realiza el entrenamiento del modelo con los datos dev y α = 1 (valor por defecto). Posteriormente, se realiza la predicción sobre los datos de testeo (test). Se obtiene un accuracy de 0.87, y la matriz de confusión presentada en la Figura 11.

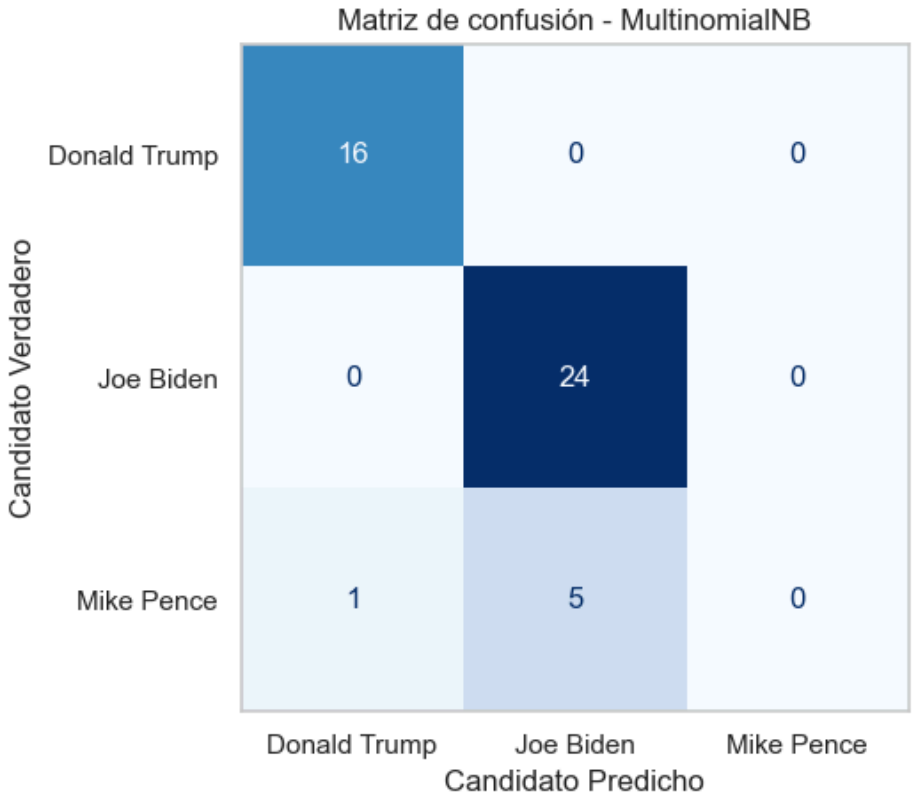


Figura 11. Matriz de confusión obtenida con el modelo Multinomial Naive Bayes, con α = 1

El valor de precision y recall para cada candidato se presenta en la Tabla 2.

Tabla 2. Precision y recall para cada candidato, con el modelo Multinomial Naive Bayes, con α = 1

Nombre de candidato/a	Precision	Recall
Donald Trump	0.94	1.00
Joe Biden	0.83	1.00
Mike Pence	0.00	0.00

Si bien el accuracy del modelo generado es alto (0.87), este no logra distinguir los discursos de Mike Pence del resto de los candidatos. Pence es, de los candidatos analizados, el que presenta menor cantidad de discursos (más adelante se comenta acerca de técnicas de submuestreo y sobremuestreo). Los valores de recall para Trump y Biden indican que los discursos que ellos efectivamente dan se predicen correctamente en el 100 % de los casos. Los valores de precision menor a 1 indican que hay discursos asignados a estos candidatos que no les corresponden.

Como se previó en el PCA, los discursos de Pence tienen similitudes tanto con Biden como con Trump, algo que no es tan intuitivo considerando que Pence pertenece al mismo partido que Trump.

Las predicciones de los modelos se pueden mejorar variando los **hiperparámetros**. Estos son parámetros que no se aprenden directamente, sino que se ingresan como argumentos. En el modelo adoptado, es el caso de α (alpha). Por otro lado, al variar estos parámetros se corre el riesgo de sobreajuste en el conjunto de testeo. Para solucionar este problema, se puede utilizar otra parte del conjunto de datos como "conjunto de validación": se entrena con dev, se evalúa en validación, y una vez definido el modelo se realiza la evaluación final en test.

El problema que puede surgir con esa alternativa, donde se dividen los datos originales en tres grupos, es que se reduce mucho la cantidad de muestras utilizadas para el aprendizaje del modelo. Además, los resultados pueden depender de la elección aleatoria específica para el par de conjuntos dev y validación.

Como solución a este problema se puede utilizar la **validación cruzada** (cross-validation, o CV). Este procedimiento conserva los conjuntos dev y test, pero no necesita un conjunto de validación. En su lugar, el conjunto de entrenamiento se divide en "k" subconjuntos más pequeños (5 por defecto). Se entrena un modelo utilizando k - 1 subconjuntos y el modelo resultante se valida en el subconjunto restante, obteniéndose las correspondientes medidas de performance (como accuracy). La medida de performance resulta del promedio de los valores calculados para los k modelos (fold). La Figura 12 esquematiza el procedimiento.[10]

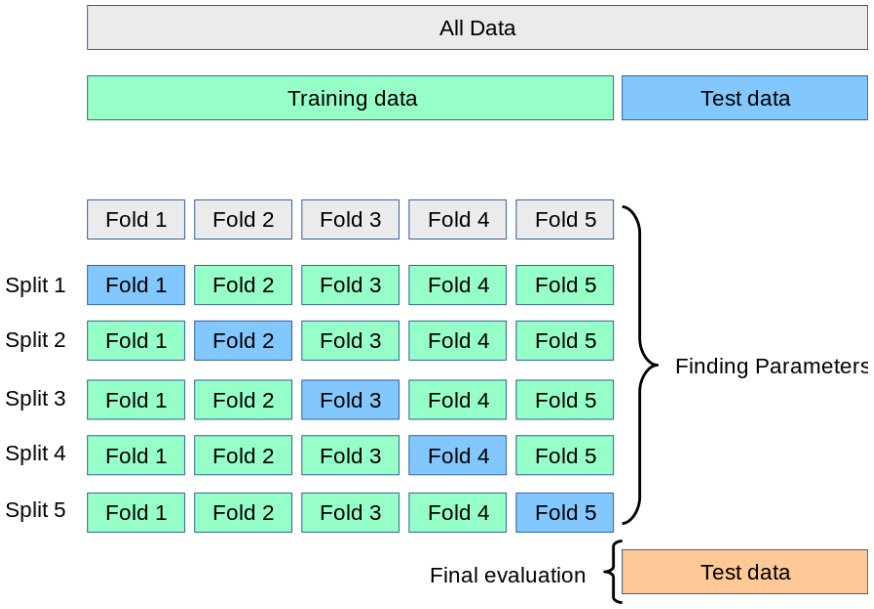


Figura 12. Esquema de validación cruzada (extraído de scikit-learn.org)

Este procedimiento se puede utilizar para optimizar la elección de los hiperparámetros. La implementación se puede hacer a través de **GridSearchCV**, que genera candidatos a partir de una cuadrícula de valores, evalúa todas las combinaciones posibles y conserva la mejor combinación.[11]

[10] Fuente: https://scikit-learn.org/stable/modules/cross_validation.html
[11] Fuente: https://scikit-learn.org/stable/modules/grid_search.html#grid-search

Se realiza una búsqueda de hiperparámetros para el modelo entrenado anteriormente, con los datos de entrenamiento, y con la siguiente grilla para alpha: 0.01, 0.1, 0.5, 1.0, 2.0 y 5.0. Como resultado, el mejor alpha es 0.01. La Figura 13 presenta la variación del accuracy para los diferentes parámetros considerados.

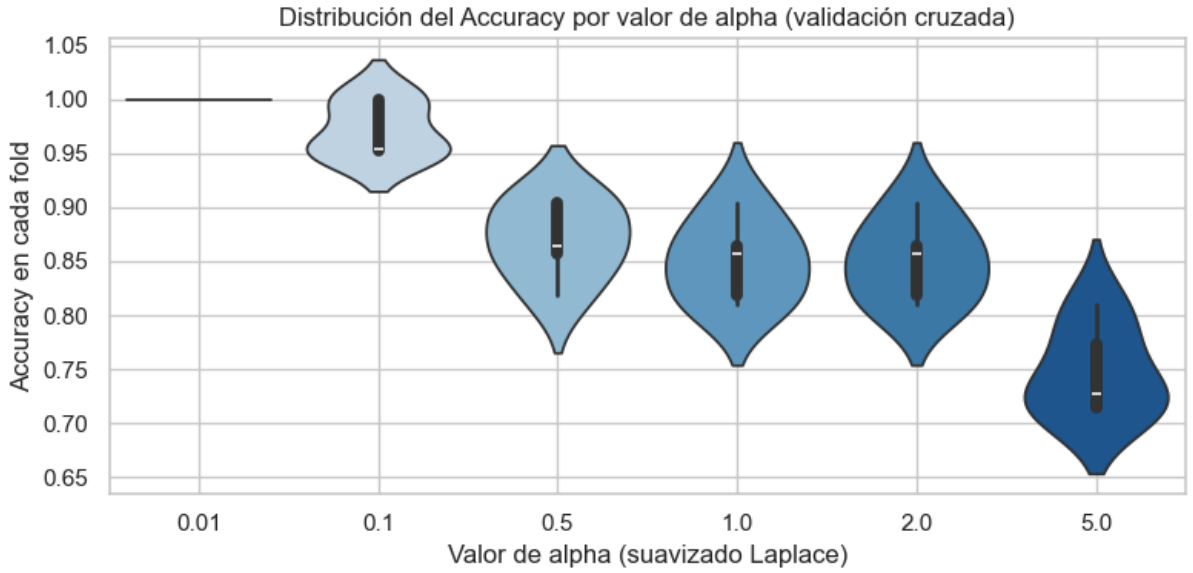


Figura 13. Distribución del accuracy en los distintos modelos entrenados

El valor promedio y la variabilidad es similar para valores de alpha entre 0.5 y 2. Para un alpha de 5 el promedio es más bajo, mientras que para alpha 0.1 o menos el promedio es alto y la variación disminuye, siendo inexistente en 0.01.

Si se utiliza el mejor modelo (mejor alpha), y se vuelve a entrenar sobre el conjunto de entrenamiento, se obtiene un accuracy de 1.00, y la matriz de confusión presentada en la Figura 14.

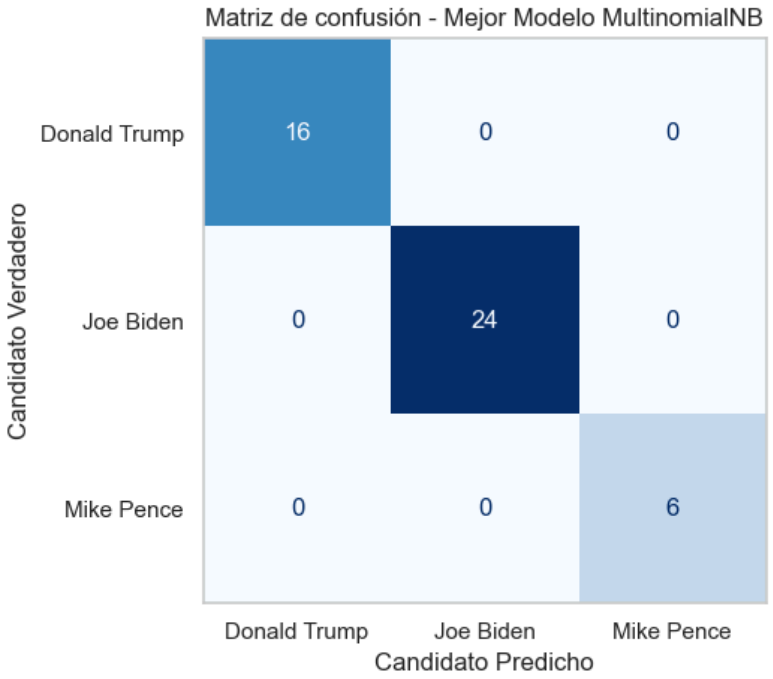


Figura 14. Matriz de confusión obtenida con el mejor modelo Multinomial Naive Bayes (α = 0.01)

El valor de precision y recall para cada candidato se presenta en la Tabla 3.

Tabla 3. Precision y recall para cada candidato, con el mejor modelo Multinomial Naive Bayes

Nombre de candidato/a	Precision	Recall
Donald Trump	1.00	1.00
Joe Biden	1.00	1.00
Mike Pence	1.00	1.00

Utilizando el mejor modelo en el conjunto de test no se tienen errores en la predicción.

Respecto a las limitaciones del uso de modelos basados en BoW o TF-IDF para el análisis del texto, se destaca el desconocimiento del contexto. Pueden haber palabras que cambien su significado dependiendo del contexto en que se utilizan, o sinónimos que se consideren como términos independientes pero en realidad están relacionados. De esta forma, se pierde información que puede ser relevante para distinguir mejor los discursos entre sí (o para realizar otro tipo de análisis más profundos, como sentimientos).

7.2. Evaluación de modelo alternativo - Random Forest

El modelo de Random Forrest combina varios clasificadores de árboles de decisión en diversas submuestras del conjunto de datos y utiliza el promedio para mejorar la precisión predictiva y controlar el sobreajuste. [12]

El tamaño de la submuestra se controla con el max_samples. Los algoritmos de Random Forrest tienen tres hiperparámetros principales, que deben establecerse antes del entrenamiento. Entre ellos están el tamaño de los nodos, el número de árboles y el número de características muestreadas. A partir de ahí, el clasificador de bosque aleatorio puede utilizarse para resolver problemas de regresión o clasificación.[13]

El modelo utilizado para predecir el candidato a partir de su discurso es un Random Forrest con una amplitud de nodos de 10, 100 árboles de decisión y la cantidad de características muestreadas es la raíz cuadrada de la cantidad de variables en los datos (11620^0.5). La matriz de confusión obtenida se presenta en la Figura 15.

[12] Fuente: https://www.ibm.com/think/topics/random-forest
[13] Fuente:https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

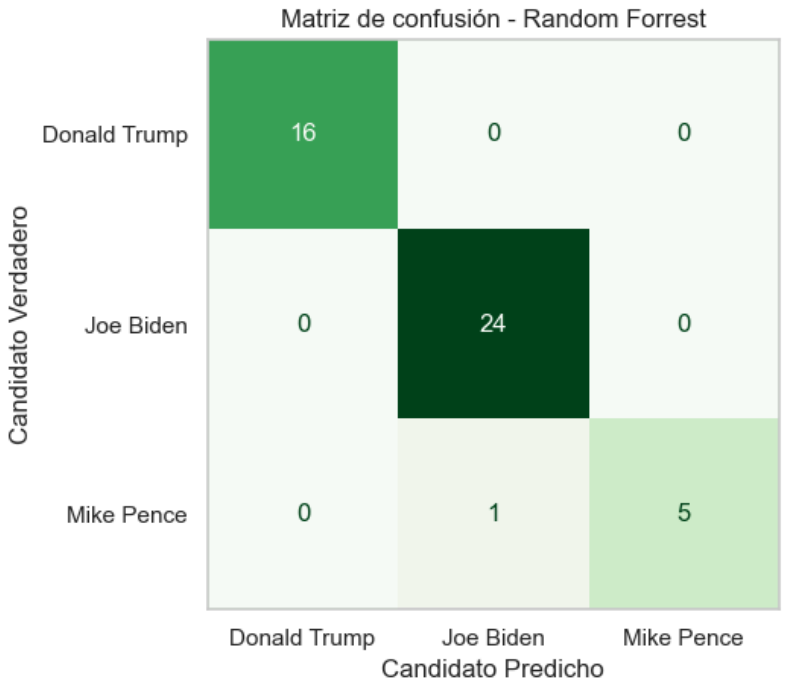


Figura 15. Matriz de confusión obtenida con Random Forest

El valor de precision y recall para cada candidato se presenta en la Tabla 4.

Tabla 4. Precision y recall para cada candidato, con el modelo Random Forest

Nombre de candidato/a	Precision	Recall
Donald Trump	1.00	1.00
Joe Biden	0.96	1.00
Mike Pence	1.00	0.83

El modelo tiene un accuracy del 98% al predecir el candidato que realizó el discurso en el conjunto de testeo, fallando solamente al asignar un discurso de Mike Pence. Un resultado significativamente consistente, sin muestras aparente de overfitting. Incluso, al observar el accuracy de cada folder al momento de realizar cross validation en la base de dev, el promedio fue de 89% con una desviación estándar de 0,0918, un resultado también consistente. En comparación con el mejor modelo de Multinomial Naive Bayes solo tiene una predicción distinta y errónea, ambos modelos se consideran consistentes y no se ve en las métricas muestras de overfitting, estrictamente el modelo de Multinomial Naive Bayes (con alpha 0.01) es mejor al predecir el 100% del conjunto de test correctamente.

7.3. Evaluación de modelo alternativo - Cambio de candidatos

Se realiza un análisis similar al realizado con los 3 candidatos con más discursos, pero cambiando candidatos: se consideran los discursos de Joe Biden, Kamala Harris y Bernie Sanders.

La Figura 16 presenta la distribución resultante de aplicar PCA sobre vectores TF-IDF del nuevo conjunto de entrenamiento, con filtrado de stop words (para idioma inglés), el parámetro use_idf = True y consideración de unigramas y bigramas (ngram_range=(1,2)),

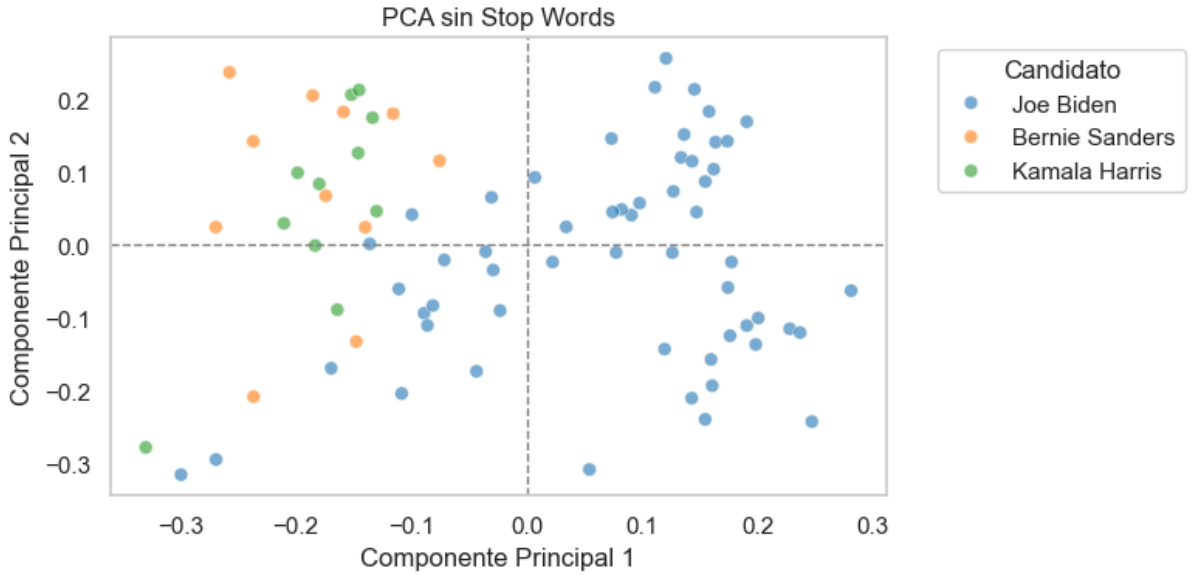


Figura 16. PCA sobre vectores TF-IDF del nuevo conjunto de entrenamiento.

En este caso no se observan agrupamientos bien definidos como pasaba con los otros 3 candidatos. En la Figura 16 también resulta evidente que hay muchos más datos de un candidato, Biden, que de los demás. Los discursos de los otros candidatos son pocos y se mezclan entre sí. Esto puede generar problemas a la hora de predecir quién dijo un nuevo discurso.

Se entrena un modelo Random Forest sobre el nuevo conjunto de entrenamiento, se predicen los candidatos para el conjunto de testeo, y se obtiene un accuracy de 0.74. La matriz de confusión para este caso se presenta en la Figura 17.

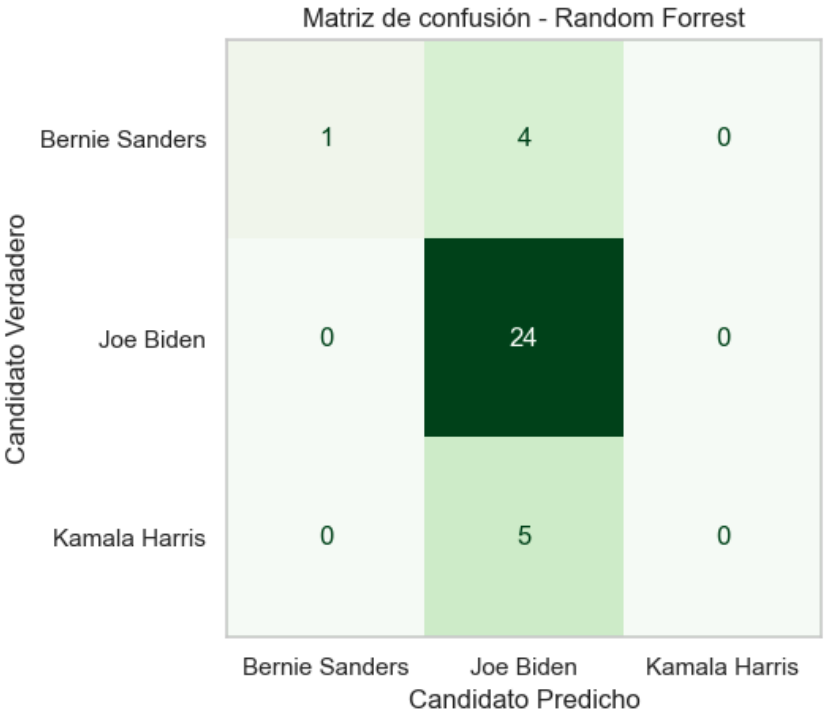


Figura 17. Matriz de confusión obtenida con Random Forest para el nuevo conjunto de datos

El valor de precision y recall para cada candidato/a se presenta en la Tabla 5.

Tabla 5. Precision y recall para cada candidato/a, para el nuevo conjunto de datos

Nombre de candidato/a	Precision	Recall
Bernie Sanders	1.00	0.20
Joe Biden	0.73	1.00
Kamala Harris	0.00	0.00

Se confirma que el modelo no logra identificar adecuadamente los discursos de los candidatos con menos cantidad de datos. Existen técnicas que abordan el problema de clases desbalanceadas, ayudando a crear conjuntos de datos equilibrados: estos son el **sobremuestreo y el submuestreo**. El sobremuestreo es una técnica que se utiliza cuando los datos de una clase dada son insuficientes (pocos); con ella busca aumentar los resultados de las minorías y obtener una cantidad más equilibrada de resultados positivos. Por otro lado, el submuestreo aborda las clases que tienen muchos datos en comparación al resto, buscando reducir esa cantidad para equilibrarla con las demás clases.[14]

En este caso, se tiene una clase con muchos datos en comparación a las otras 2, por lo tanto se podría recurrir a técnicas de submuestreo para equilibrar los datos.

8. Técnica alternativa para extraer características de texto

Las técnicas utilizadas en este trabajo para caracterizar los textos son técnicas basadas en conteo de palabras: BoW y TF-IDF. Tienen como ventaja que son de fácil implementación, pero son técnicas que no abarcan demasiado el contexto de los textos. Existen otras técnicas, como Word Embeddings, que ofrecen representaciones más ricas del texto.

Las Word Embeddings (incrustaciones de palabras) son una forma de representar palabras como vectores en un espacio multidimensional, donde la distancia y la dirección entre vectores reflejan la similitud y las relaciones entre las palabras correspondientes. A diferencia de los métodos tradicionales, se trata de vectores densos con valores continuos que son entrenados utilizando aprendizaje automático, usualmente basado en redes neuronales (requiere grandes cantidades de texto). Un método popular para entrenar Word Embeddings es Word2Vec, que utiliza una red neuronal para predecir las palabras que rodean a una palabra objetivo en un contexto determinado.[15]

[14] Fuente: https://www.techtarget.com/whatis/definition/over-sampling-and-under-sampling
[15] Fuente: https://www.ibm.com/think/topics/word-embeddings

ANEXO

Palabras eliminadas del texto mediante NLTK (stop words)[16]

{'other', 'more', "she's", 'haven', 'only', "i'm", "didn't", 'very', 'have', 'he', "i'll", 'his', 'shan', 'me', 'ours', "weren't", 'don', "mightn't", 'll', 'its', 'than', 'with', 'wouldn', 'which', 'yourselves', 'does', "he'd", 'hasn', 'nor', 'between', 'a', 'your', 'itself', 'few', 'further', 'we', 'him', 'themselves', 'were', 'i', 'this', 'after', "she'll", 'why', 'has', "you're", 'their', 'ma', "isn't", 'wasn', 'as', 'where', 'once', "we've", "he'll", 'of', 'if', 'each', 'they', "they've", 'didn', 'her', 'mustn', 'up', 'at', 'yourself', "should've", 'm', "you've", "you'll", 'then', 'doesn', 'theirs', "we're", 'yours', 'below', 'ourselves', "haven't", 'no', "hadn't", 'needn', 'herself', 'own', "we'll", "won't", 'aren', "doesn't", 'that', 'here', 'she', 'couldn', 'can', "hasn't", 'just', 're', 'about', 'against', 'off', 'too', 'my', 'o', 'hadn', 'during', "he's", 'an', 'these', 'is', 'weren', "i've", 'some', "aren't", 'will', 'to', 'had', 'all', 'are', "they'd", 'again', "wasn't", 'mightn', "couldn't", 'same', 'it', 'y', "she'd", 'now', 'or', "we'd", 'over', 'above', 'd', 'shouldn', 't', "they'll", 'when', 'them', 'from', "they're", "shouldn't", 'there', 'did', 'any', 'our', 'not', 'such', 'being', 'what', 'those', 'in', 'myself', 'won', "wouldn't", 's', 'on', 'been', "i'd", "shan't", 'but', 'while', 'and', 'into', 'having', 'ain', 'you', 'most', 'through', "that'll", 'himself', 'am', 'under', 'until', 'who', 'both', 'hers', 'do', 'for', "you'd", 'down', "don't", 'was', 'should', 'out', 've', 'the', "needn't", 'so', 'because', 'by', 'whom', 'isn', "mustn't", 'before', 'be', "it'll", "it'd", "it's", 'doing', 'how'}

[16] Fuente: https://pythonspot.com/nltk-stop-words/