Tarea 2

Análisis de discursos de candidatos presidenciales de Estados Unidos en 2020

Josefina Delgado, Mateo Bentancur

Introducción a la Ciencia de Datos, 2025

Índice

Introducción

En este trabajo se pretende continuar el análisis de datos del conjunto de datos utilizado en la Tarea 1, que contiene los discursos de los distintos candidatos a la presidencia de Estados Unidos en 2020. Se busca utilizar herramientas de aprendizaje automático para enriquecer el análisis.

1. Metodología

2.1. Dataset y representación numérica

Para realizar esta sección se utilizó la biblioteca de aprendizaje automático *scikit-learn*, que incluye algoritmos de clasificación, agrupamiento, entre otros.

2.1.1. Procesamiento del conjunto de datos

Se utilizó el *data frame* de los discursos dados por los candidatos durante la campaña electoral, este contiene el nombre del candidato, la fecha del discurso, el título, la transcripción completa, la ubicación donde se llevó a cabo y el tipo de evento o discurso.

El trabajo fue realizado en Google Colab, usando Python como lenguaje de programación. Finalmente, el trabajo se cargó en GitHub.

Se identificaron todos los candidatos que aparecen al menos una vez en la columna *speaker*. Para aquellos registros con múltiples oradores, se utilizó la función 'split' para separar los nombres utilizando la coma como separador. Se creó una nueva fila para cada orador manteniendo el resto de la información.

Con este data frame modificado, se contaron los discursos por candidatos. Se redujo el conjunto de datos a los tres con mayor cantidad de discursos.

Luego, se realizó una limpieza de texto de la columna discursos, donde se encuentra la transcripción.
La función consiste en:

a. Tomar solamente los fragmentos del discurso que pertenecen al orador principal.
b. Normalizar el texto, pasando a minúsculas, eliminando los acentos, la puntuación y los espacios múltiples.

2.1.2. Partición estratificada de datos

Se particionaron los datos en dos subconjuntos: un 70% para entrenamiento y un 30% para evaluación. Esta partición se realizó mediante un muestreo estratificado, que consiste en dividir el conjunto de datos de manera tal que la proporción de instancias por clase se conserve tanto en el conjunto de entrenamiento como en el de prueba. Esto es especialmente importante en tareas de clasificación cuando las clases están desbalanceadas.

Para implementarlo, se utilizó la función *train_test_split,* se utilizó el parámetro *stratify='speaker'* para asegurar que ambos subconjuntos tengan la misma proporción de estas clases, y un valor de *random_state* para obtener resultados reproducibles.

2.3. Técnica Bag of words

Se transformaron los textos del conjunto de entrenamiento a una representación numérica utilizando la técnica *Bag of Words*. Este método convierte cada documento en un vector de frecuencias, donde cada dimensión representa una palabra del vocabulario construido a partir del conjunto de entrenamiento, y cada valor indica cuántas veces aparece esa palabra en el texto correspondiente.

Para realizar esta transformación se utilizó la clase *CountVectorizer* de *scikit-learn*, que construye el vocabulario automáticamente a partir de los textos preprocesados, asigna a cada palabra una posición única en el vector y genera una matriz numérica en la que cada fila representa un discurso y cada columna una palabra del vocabulario. El valor de cada celda corresponde a la frecuencia absoluta de esa palabra en el discurso.

2.4. Técnica TF-IDF

Para complementar el análisis, se comparó la técnica de Bag of Words, que simplemente cuenta la frecuencia de cada palabra sin tener en cuenta su importancia relativa, con la técnica TF-IDF (Term Frequency – Inverse Document Frequency), que asigna un peso a cada palabra basado tanto en su frecuencia dentro de un documento como en su distribución. Esta técnica permite atenuar la influencia de palabras muy comunes y resaltar aquellas que son más características de un texto específico.

TF-IDF ajusta los conteos del modelo Bag of Words combinando dos factores:

- TF (Term Frequency): Mide cuántas veces aparece una palabra en un documento.
- IDF (Inverse Document Frequency): Penaliza las palabras que aparecen en muchos documentos, ya que aportan poca información para distinguir entre textos.

Como resultado, las palabras que son frecuentes en un documento pero poco comunes en el resto del cuerpo reciben un peso TF-IDF alto, ya que se consideran más representativas de ese contenido.

Para implementar esta representación se utilizó la clase *TfidfVectorizer* de scikit-learn, configurada para eliminar las *stopwords* (palabras funcionales sin valor semántico fuerte), aplicar penalización IDF sobre la frecuencia global de cada término, y considerar tanto unigramas como bigramas.

Un n-grama es una secuencia de n palabras consecutivas dentro del texto; en este caso, se incorporaron secuencias de una y dos palabras (unigramas y bigramas). Esto permite capturar patrones lingüísticos más específicos que los que se obtendrían considerando

únicamente palabras aisladas (como “united states” o “health care”), y contribuye a mejorar la capacidad del modelo para distinguir entre diferentes estilos de discurso.

2.5. Visualización usando PCA

Con el objetivo de visualizar los discursos en un espacio de menor dimensión y explorar si es posible distinguir a los candidatos en función de sus estilos de comunicación, se aplicó la técnica de Análisis de Componentes Principales (PCA) sobre los vectores TF-IDF generados previamente.

PCA es una técnica de reducción de dimensionalidad que transforma los datos originales en un nuevo sistema de coordenadas, donde las primeras componentes principales retienen la mayor parte de la varianza del conjunto. En este caso, se proyectaron los vectores TF-IDF en un espacio de dos dimensiones, seleccionando las dos componentes principales que capturan la mayor variabilidad entre los discursos.

Si los puntos correspondientes a los diferentes candidatos se agrupan visualmente en el plano 2D, esto sugiere que es posible separarlos utilizando únicamente esas dos dimensiones, lo que implica que los estilos lingüísticos de los candidatos son diferenciables incluso en un espacio reducido. Como complemento, también se aplicó PCA con tres componentes principales y se representaron los discursos en un espacio tridimensional. Esto permitió observar si la adición de una dimensión extra mejoraba la separación entre los grupos, proporcionando una visualización más completa de la estructura interna de los datos.

Por último, se realizó un gráfico que representa la varianza explicada por cada componente principal, a fin de observar que tanto aporta cada componente para la separación de los grupos.

2.6. Entrenamiento y evaluación de modelos

2.6.1 Clasificación con Multinomial Naive Bayes

Se entrenó un modelo Multinomial Naive Bayes utilizando los vectores TF-IDF del conjunto de entrenamiento, con el objetivo de clasificar cada discurso según su orador.

Se aplicó el modelo al conjunto de test y se evaluó su rendimiento con los parámetros *accuracy*, *precision* y *recall*, así como la matriz de confusión. Las métricas se calcularon de la siguiente manera:

- La accuracy se calculó como la proporción de predicciones correctas sobre el total de discursos evaluados.
- La precision para una clase se definió como la proporción de discursos correctamente asignados a esa clase entre todos aquellos que el modelo predijo como pertenecientes a ella.
- El recall se definió como la proporción de discursos de una clase que fueron correctamente identificados por el modelo, es decir, de todos los discursos reales de un orador, cuántos fueron clasificados correctamente.

Realizar el análisis solo observando la accuracy podría ser problemático en un escenario donde un candidato tenga muchos más discursos que los demás, ya que, un modelo podría alcanzar un accuracy alto simplemente favoreciendo esa clase mayoritaria, sin realmente aprender a diferenciar entre los oradores.

2.6.2. Técnica de validación cruzada

La validación cruzada consiste en dividir el conjunto de entrenamiento en *k* partes (o pliegues) del mismo tamaño. En cada iteración, se entrena el modelo con $k-1$ partes y se evalúa con la parte restante. Este proceso se repite *k* veces, rotando la parte utilizada para validar. Finalmente, se calcula el promedio de las métricas obtenidas en cada iteración.

Este método permite estimar cómo generalizaría el modelo a datos no vistos, sin utilizar aún el conjunto de test, y reduce la dependencia de una única partición del entrenamiento.

Se utilizó *GridSearchCV* para hacer una búsqueda de hiperparámetros con validación cruzada. Esta herramienta permite evaluar automáticamente todas las combinaciones posibles de parámetros definidos en una grilla, aplicando validación cruzada para cada una.

Se utilizaron 5 pliegues sobre el conjunto de entrenamiento para determinar el valor óptimo del hiperparámetro α del modelo Multinomial Naive Bayes. De esta manera, se puede evaluar el rendimiento promedio del modelo con distintos valores de α, calculando la accuracy y su desviación estándar para cada caso. Se representaron los resultados en un gráfico de violín.

Por último, se seleccionó el hiperparámetro con mejor rendimiento (a partir de la accuraccy) y se entrenó todo el modelo nuevamente con el set de entrenamiento completo.

Los modelos basados en Bag of Words o TF-IDF representan los textos como vectores numéricos según la frecuencia de las palabras, pero ignoran el orden y el contexto, por lo que no capturan el significado real del lenguaje. Además, generan vectores de alta dimensionalidad y muy dispersos, lo que puede dificultar el entrenamiento y aumentar el riesgo de sobreajuste. Tampoco reconocen sinónimos ni diferencias de sentido según el contexto, y no manejan bien palabras nuevas que no estaban en el entrenamiento.

2.6.3 Modelo de Regresión Logística

Se utilizó otro modelo proporcionado por la librería de scikit-learn y se le calcularon las métricas correspondientes para evaluar su desempeño, asi como también la matriz de confusión.

La regresión logística es un modelo de clasificación supervisada que estima la probabilidad de que una instancia pertenezca a una clase utilizando una función sigmoide. A diferencia de Multinomial Naive Bayes, que asume independencia entre las palabras, la regresión logística considera las correlaciones entre los atributos (en este caso, las palabras o bigramas) y aprende los pesos óptimos para cada uno, maximizando la verosimilitud del conjunto de entrenamiento.

2.7. Desbalance de clases

Se seleccionaron tres candidatos: Joe Biden (con muchos discursos) y otros dos con entre 3 y 14 discursos cada uno. Se filtraron sus discursos, se limpiaron los textos, y se dividieron en conjuntos de entrenamiento y prueba de manera estratificada.

Luego, se aplicó TF-IDF para transformar los textos en vectores numéricos, y se visualizó la distribución de los discursos en el espacio reducido mediante PCA en dos y tres dimensiones. Finalmente, se entrenó un modelo Multinomial Naive Bayes sobre los datos desbalanceados, y se evaluó su rendimiento con métricas de clasificación y una matriz de confusión.

3. Resultados

3.1. Datos de partida

3.1.1. Discursos por candidatos

El conjunto de datos se redujo a los tres candidatos con la mayor cantidad de discursos, los mismos se muestran en la Tabla 3.

Candidatos	Cantidad de discursos
Joe Biden	81
Donald Trump	54
Mike Pence	20

Tabla 1: Cantidad de discursos por candidato

3.1.2. Muestreo estratificado

En la Figura 2, se muestran los conjuntos de entrenamiento (train) y de prueba (test) de tal forma que un 70% del total de discursos se utilice para entrenar el modelo y el 30% restante para probarlo.

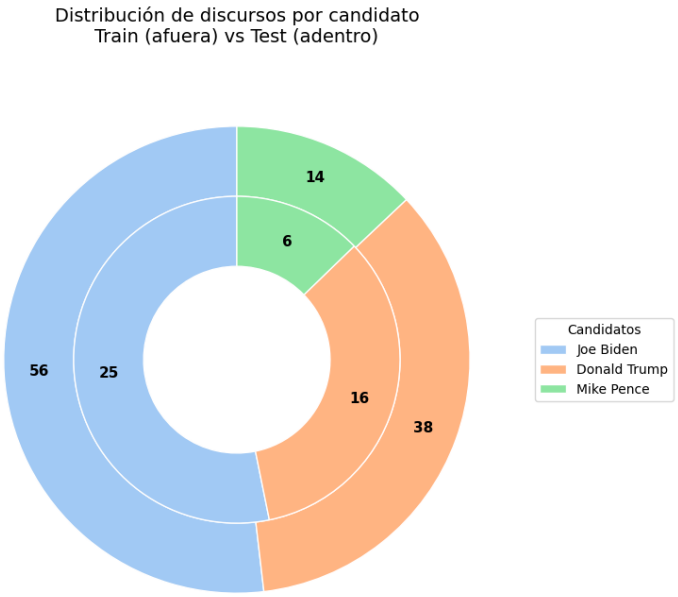

Figura 3: Distribución de discursos por candidato para los subconjuntos de entrenamiento y prueba. El número dentro de los anillos representa la cantidad de discursos que el candidato aportó para cada subconjunto.

3.2. Implementación de la metodología Bag of Words

La representación resultante es una matriz de alta dimensionalidad, ya que contiene una columna por cada palabra única del vocabulario. Como en cada discurso solo aparece una pequeña fracción de ese total de palabras, la mayoría de las celdas contienen el valor cero, lo que implica que se trata de una matriz dispersa (sparse).

En la Figura 4 se muestran las 10 primeras filas (que representan los discursos) de la matriz, en donde se observa la gran densidad de valores nulos y algunos valores unitarios para las palabras que aparecieron en el discurso.

Matriz Bag of Words (primeras filas):

	aa	aaa	aapi	ab	abandon	abandoned	abbott	abc	abdelaziz	abducted	\
0	0	0	0	0	0	0	0	1	0	0	
1	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	1	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	

	...	zigzag	zip	zo	zoes	zone	zones	zoning	zoom	zucker	zuckerberg
0	...	0	0	0	0	0	0	1	0	0	0
1	...	0	0	0	0	0	1	0	0	0	0
2	...	0	1	0	0	0	0	0	0	0	0
3	...	0	0	0	0	0	0	0	0	0	0
4	...	0	0	0	0	0	0	0	0	0	0
5	...	0	0	0	0	0	0	0	0	0	0
6	...	0	0	0	0	0	1	0	0	0	0
7	...	0	0	0	0	0	0	0	0	0	0
8	...	0	0	0	0	0	0	0	0	0	0
9	...	0	0	0	0	0	0	0	0	0	0

Figura 4: Primeras filas de la matriz de bag of words

Es interesante observar cómo se reconocen como palabras “aa” y “aaa”, que por más que no sean parte del inglés, estas pueden tener un significado claro a la hora de hablar de las distintas calificaciones crediticias de las empresas (empresas doble o triple A).

La matriz resultante tiene dimensiones de 108 x 11017, por ende, hay 11017 palabras únicas en el conjunto de entrenamiento que se compone de 108 discursos. Se verifica que es una matriz dispersa ya que posee un 92,6% de valores nulos.

3.3. Implementación de la técnica TF-IDF

Para la implementación de la TF-IDF, se utilizaron los filtros que eliminan las *stop words* y se incluyeron tanto unigramas como bigramas, ya que, luego se utilizarán para hacer la representación con PCA.

A pesar de eliminar las stop words, la matriz TF-IDF es mucho más grande, ya que, la cantidad de columnas es el número total de unigramas y bigramas únicos en el conjunto de entrenamiento. La matriz resultante tiene dimensiones de 108 x 132246 y también es una matriz dispersa.

La representación numérica de la matriz TF-IDF obtenida se muestra en la Figura 5, en donde se representan los n-gramas - en este caso unigramas, pues son los más comunes - con mayor puntuación TF-IDF para los 10 primeros discursos.

	going	people	president	know	said	thats	dont	just	think	im	great	like	theyre	trump	right	american	country	want	say	got
0	0.183	0.198	0.055	0.154	0.131	0.109	0.141	0.080	0.101	0.058	0.111	0.124	0.131	0.151	0.102	0.045	0.053	0.076	0.056	0.100
1	0.224	0.148	0.030	0.132	0.164	0.088	0.146	0.064	0.053	0.043	0.136	0.076	0.112	0.045	0.068	0.054	0.064	0.110	0.088	0.082
2	0.060	0.060	0.147	0.055	0.047	0.030	0.015	0.070	0.021	0.025	0.005	0.010	0.016	0.141	0.041	0.058	0.067	0.030	0.010	0.005
3	0.063	0.089	0.150	0.037	0.033	0.027	0.022	0.048	0.000	0.026	0.063	0.011	0.017	0.000	0.005	0.050	0.038	0.011	0.044	0.011
4	0.008	0.042	0.043	0.025	0.000	0.068	0.017	0.034	0.009	0.008	0.000	0.043	0.000	0.029	0.026	0.009	0.061	0.068	0.035	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.120	0.104	0.011	0.066	0.046	0.056	0.073	0.072	0.092	0.022	0.125	0.034	0.041	0.012	0.034	0.029	0.091	0.134	0.079	0.088
7	0.105	0.055	0.226	0.087	0.057	0.032	0.012	0.071	0.025	0.031	0.055	0.028	0.013	0.097	0.053	0.125	0.065	0.032	0.024	0.059
8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
9	0.104	0.052	0.220	0.068	0.084	0.004	0.016	0.040	0.038	0.028	0.074	0.041	0.000	0.154	0.037	0.136	0.041	0.012	0.021	0.013

Figura 5: Fragmento de la matriz TF-IDF.

Como complemento al análisis, se muestran en la Figura 6 los diez unigramas y bigramas con mayor TF-IDF acumulado a lo largo de todos los discursos. Como cada n-grama tiene un valor de TF-IDF distinto por discurso, para evaluar cuales son los que tienen mayor relevancia se realizó la suma a lo largo de todos los discursos y se los representó de mayor a menor.

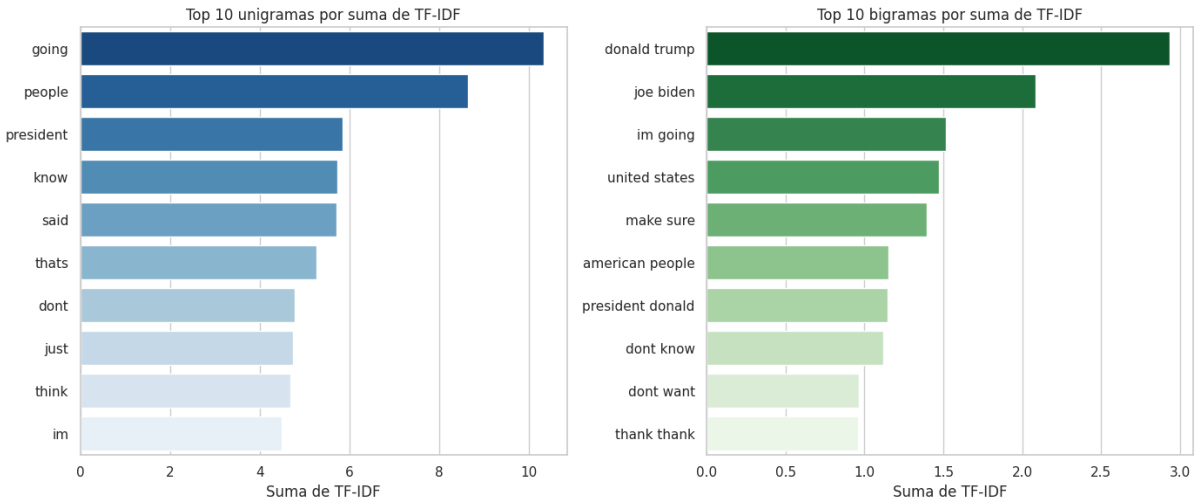

Figura 6: Los diez unigramas y bigramas con mayor valor de TF-IDF acumulado entre todos los discursos.

3.4. Representación de las componentes principales

El resultado de representar el conjunto de entrenamiento en un espacio de dos dimensiones se muestra en la Figura 7. Se observan los *clusters* formados para cada candidato, en donde se puede separar claramente a Mike Pence, mientras que Joe Biden y Donald Trump se encuentran superpuestos en bastantes casos.

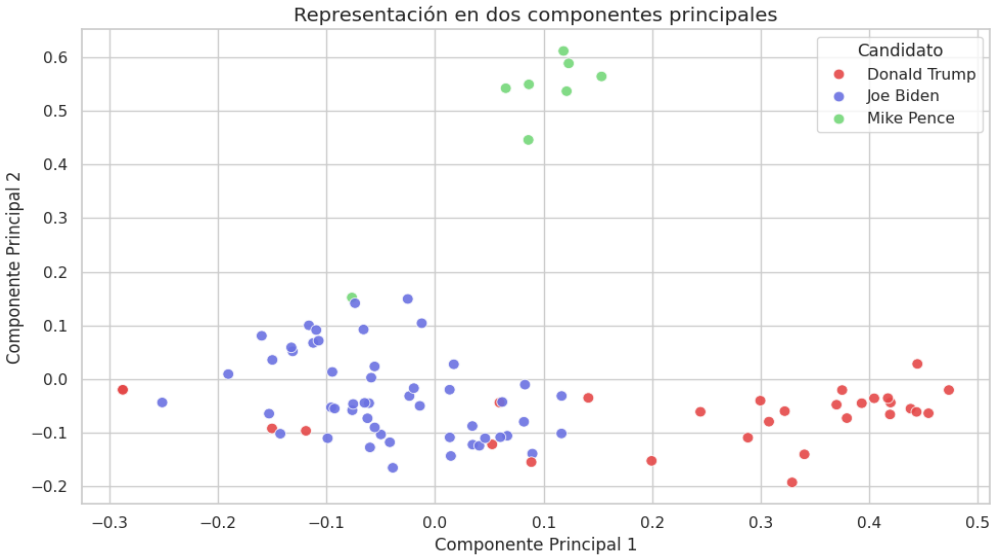

Figura 7: Representación de las clases en dos componentes principales

No alcanzan dos componentes para lograr una separación clara entre clases (aunque ya se captaron patrones útiles), ya que, estos explican un porcentaje limitado de la varianza total.

Adicionalmente, se realizó la representación en tres componentes, que se puede graficar de manera clara en un espacio de tres dimensiones. La misma se exhibe en la Figura 8, donde se ve que aumenta el grado de distinción entre las clases.

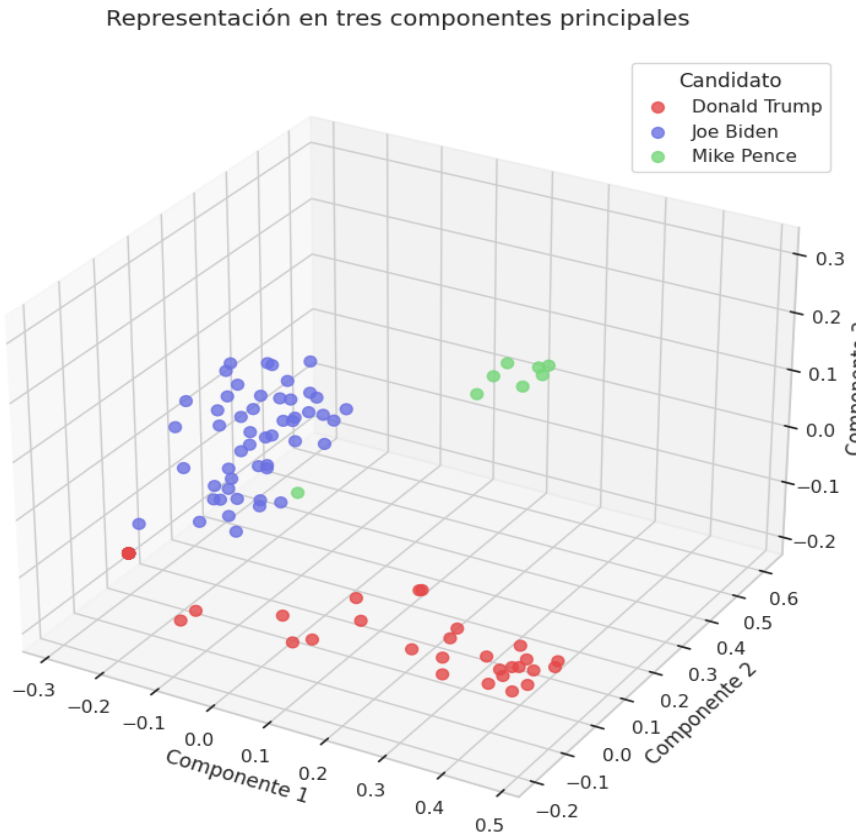

Figura 8: Representación de las clases en tres componentes principales

Aunque la representación tridimensional permite observar una separación general entre los tres candidatos, las tres primeras componentes principales explican únicamente el 14 % de la varianza total (ver Figura 9). Esto indica que la información está altamente dispersa entre muchas dimensiones (lo cual es esperable por la complejidad del problema). Es posible que ciertos discursos atípicos (como el de Mike Pence que se superpone en el cluster de Joe Biden) estén más alejados de su grupo, y por tanto requieren más componentes para lograr una separación completa. Aun así, la visualización resulta útil para identificar patrones globales.

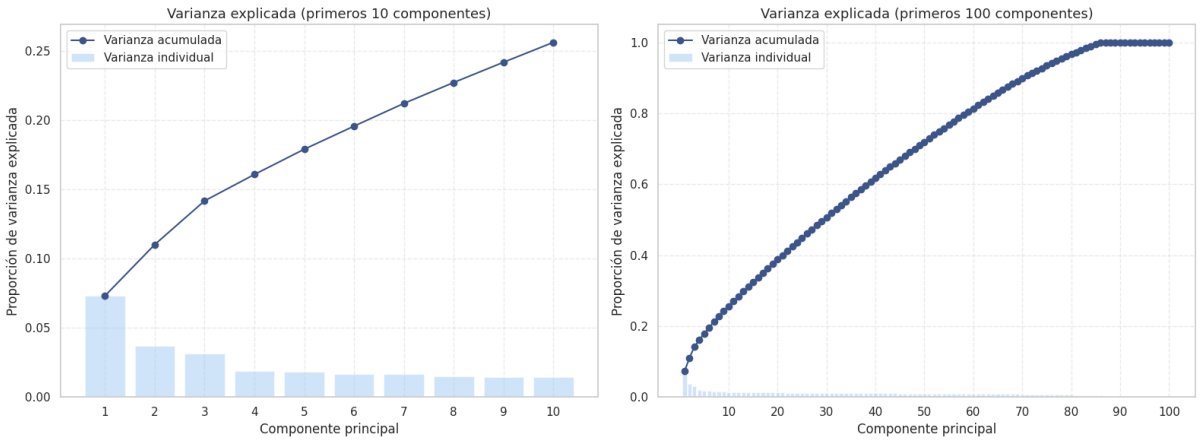

Figura 9: Proporción de la varianza explicada según la cantidad de componentes principales.

3.5. Entrenamiento y evaluación de modelos

3.5.1. Implementación del modelo Multinomial Naive Bayes

La prueba tuvo una métrica de accuracy del 75% y a continuación se reportan en la Tabla 2 los valores de precision y recall para los tres candidatos.

Candidatos	precision	recall
Joe Biden	0,676	1,000
Donald Trump	1,000	0,625
Mike Pence	0,000	0,000

Tabla 2: Métricas para la evaluación del modelo Multinomial Naive Bayes para cada candidato

La matriz de confusión (Figura 10) revela que el modelo clasifica correctamente todos los discursos de Joe Biden, mientras que tiene dificultades para distinguir los de Donald Trump, y especialmente los de Mike Pence, que son completamente absorbidos por la clase de Biden. Esta situación puede estar influenciada por dos factores: la similitud en el lenguaje utilizado entre los candidatos y un posible desbalance en la cantidad de discursos disponibles por clase.

Por desbalance se entiende que algunas clases (candidatos) están representadas con muchos más ejemplos que otras en el conjunto de entrenamiento, a pesar de haber realizado un muestreo estratificado para mantener las proporciones. En este caso, como Biden tiene muchos más discursos que Pence, el modelo tiende a “aprender más” sobre Biden, y en ausencia de suficiente información para Pence, asigna sus discursos a la clase mayoritaria.

Esto afecta directamente a la métrica accuracy, ya que un modelo puede lograr un valor alto simplemente prediciendo siempre la clase más frecuente. Por ejemplo, si la mayoría de los discursos son de Biden, un modelo que siempre predice a Biden tendría una alta accuracy, aunque su desempeño real en distinguir entre candidatos sea bajo. Por eso, cuando hay clases desbalanceadas, es bueno complementar la evaluación con métricas como precision y recall por clase.

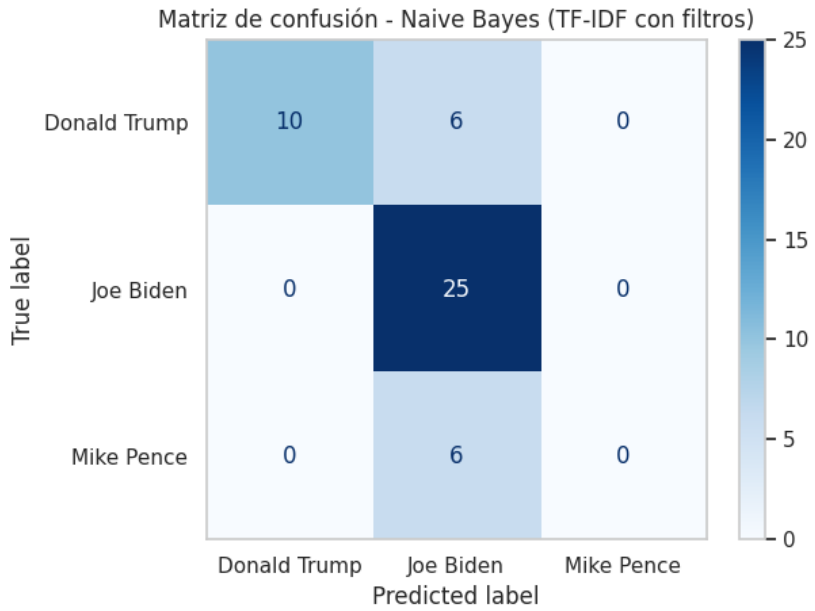

Figura 10: Matriz de confusión a partir de la evaluación del conjunto *test* utilizando un modelo tipo Multinomial Naive Bayes.

3.5.2. Implementación de GridsearchCV

En la Figura 11 se muestra cómo varía la accuracy en validación cruzada del modelo Naive Bayes según el valor del hiperparámetro α, utilizando un gráfico de violín. Este tipo de visualización combina la forma de una distribución (como un histograma suavizado) con un diagrama de caja en su interior, lo que permite ver tanto la dispersión como la mediana y el rango intercuartílico.

Se observa que valores bajos de α, se logra una mayor precisión y menor variabilidad. A medida que α aumenta, la precisión disminuye notablemente y la distribución se vuelve más amplia o desplazada hacia valores bajos, lo que indica que un exceso de suavizado degrada el desempeño del modelo.

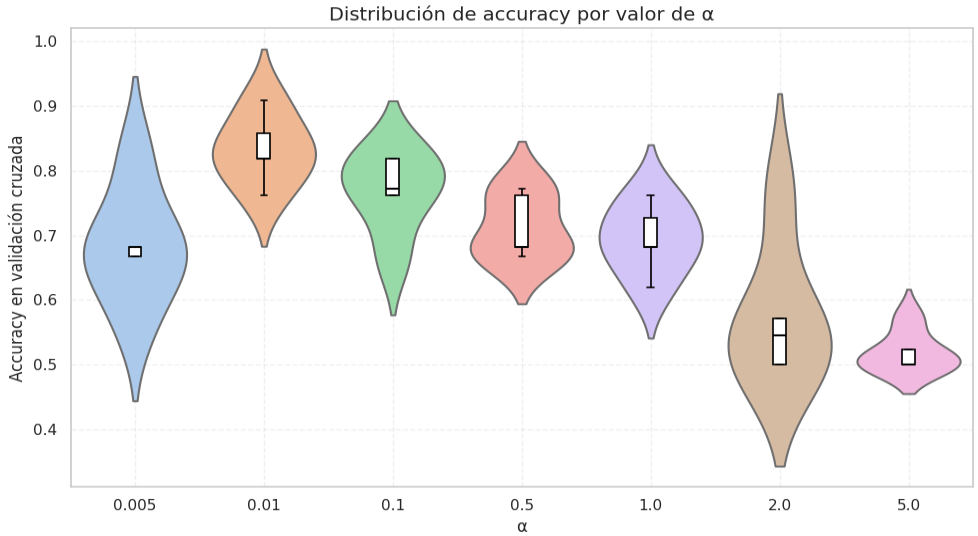

Figura 11: Gráfico de violín para la métrica *accuracy* para distintos valores de α (hiperparámetro).

Se entrenó el modelo para el mejor valor de α, que resultó en 0,015 y se obtuvo una accuracy del 85 %. Las métricas se reportan en la Tabla 3, así como su matriz de confusión (Figura 12).

Candidatos	precision	recall
Joe Biden	0,781	1,000
Donald Trump	1,000	0,688
Mike Pence	1,000	0,667

Tabla 3: Métricas de evaluación aplicando validación cruzada con GridsearchCV con el valor del hiperparámetro α que presenta una mayor accuracy global

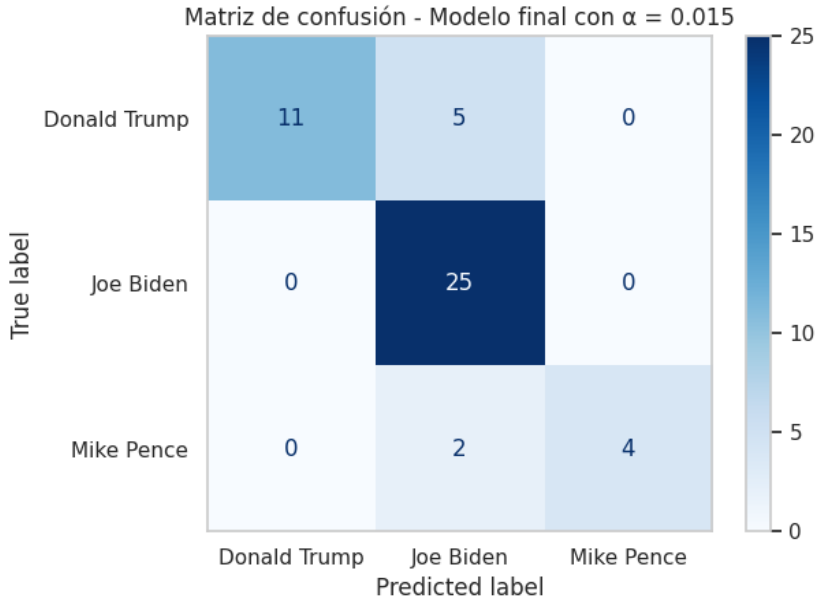

Figura 12: Matriz de confusión para el modelo Multinomial Naive Bayes aplicando validación cruzada con GridsearchCV para un α = 0,015.

Tras ajustar el valor de α mediante validación cruzada, el modelo final mejora el reconocimiento de los discursos de Mike Pence, clasificando correctamente 4 de ellos, cuando antes no lograba identificar ninguno. Además, mantiene el desempeño sobre los discursos de Joe Biden, con 25 aciertos sobre 25 en ambos casos.

En cuanto a Donald Trump, el modelo final muestra una leve mejora: pasa de 10 a 11 aciertos, reduciendo ligeramente las confusiones con Biden (de 6 a 5). También disminuyen los errores sistemáticos donde los discursos de Pence eran asignados incorrectamente a Biden (de 6 a 2).

3.5.2. Implementación del modelo de Regresión Logística

En la Tabla 3 se muestran las métricas obtenidas utilizando el modelo de regresión logística, así como su matriz de confusión en la Figura 13.

Este modelo presenta prácticamente los mismos resultados que el Multinomial Naive Bayes utilizando el mejor valor del hiperparámetro α.

Candidatos	precision	recall
Joe Biden	0,758	1,000
Donald Trump	1,000	0,688
Mike Pence	1,000	0,500

Tabla 4: Métricas para la evaluación del modelo de regresión logística

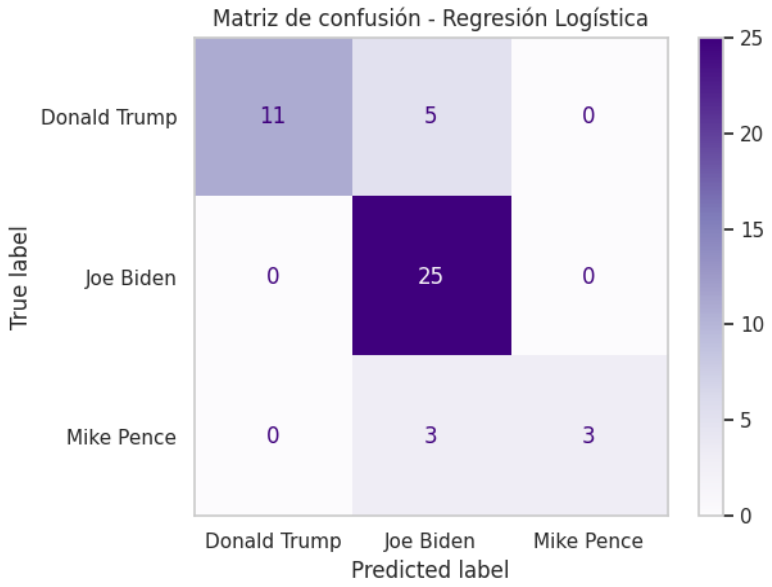

Figura 13: Matriz de confusión resultante de utilizar el modelo de regresión logística

3.6. Análisis forzando el desbalance de clases

Los candidatos evaluados fueron Joe Biden, Elizabeth Warren y Barack Obama, siendo Biden el candidato con la amplia mayoría de discursos (81). Warren y Obama solo tienen cuatro discursos, por lo que estamos ante un claro ejemplo de desbalance de clases.

Incluir candidatos con pocos discursos afecta negativamente el rendimiento del modelo. Con tan pocos ejemplos, el modelo no puede aprender patrones representativos, lo que lleva a errores sistemáticos y baja precisión en esas clases. Además, la métrica accuracy se vuelve

engañosa, ya que el modelo puede acertar simplemente favoreciendo a la clase mayoritaria. Esto limita la capacidad del modelo para generalizar y reconocer correctamente a los candidatos con menos representación en el conjunto de datos.

En la Figura 14, se observa que utilizando dos componentes principales no hay una distinción clara entre los candidatos. Si se utilizaran tres, tampoco se logra una mejora como se ve en la Figura 15

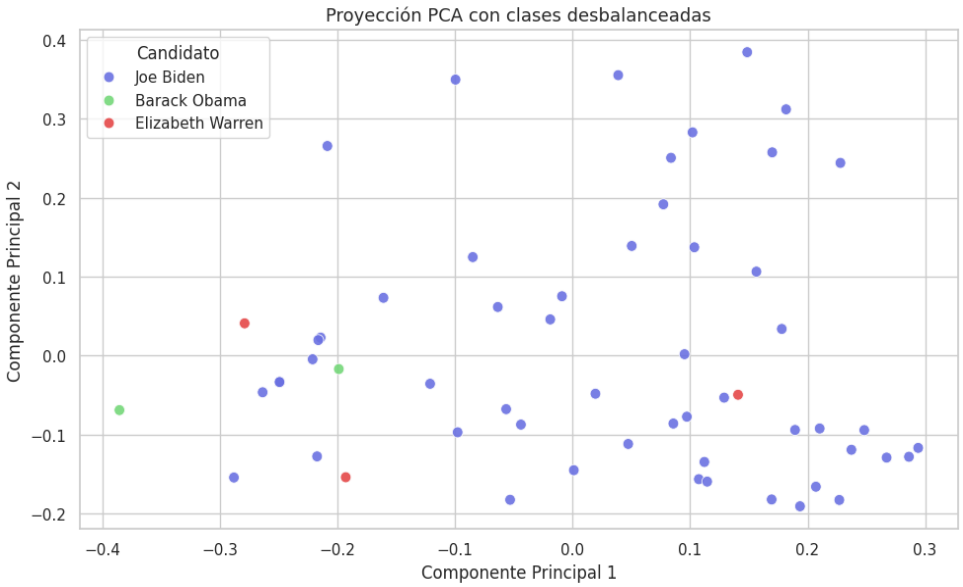

Figura 14: Proyección en dos dimensiones de los candidatos utilizando dos componentes principales

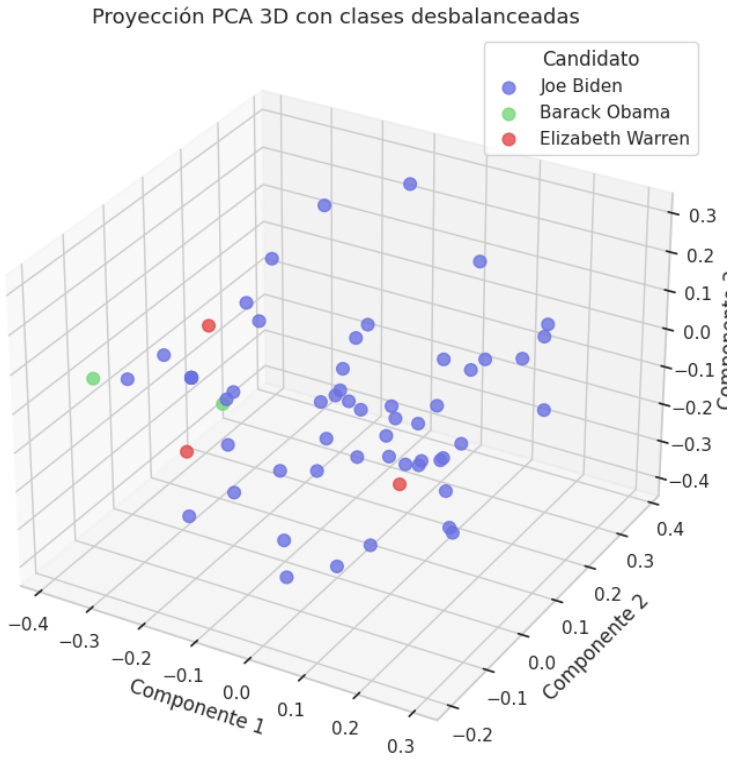

Figura 15: Proyección en tres dimensiones de los candidatos utilizando dos componentes principales

En la Tabla 5 se muestran las métricas de evaluación para el modeo de regresión logística por candidato

Candidatos	Cantidad de discursos	precision	recall
Joe Biden	81	0,930	1,000
Elizabeth Warren	4	0,000	0,000
Barack Obama	4	0,000	0,000

Tabla 5: Métricas para la evaluación del modelo de regresión logística

La matriz de confusión, para el modelo Multinomial Naive Bayes, se muestra en la Figura 16. Se observa que el modelo no puede predecir correctamente ninguno de los dos candidatos con pocos discursos. A pesar de esto, la accuracy resulta en 92%, ya que, como fue explicado antes, el modelo acierta siempre con Biden que es el candidato con la amplia mayoría de discursos.

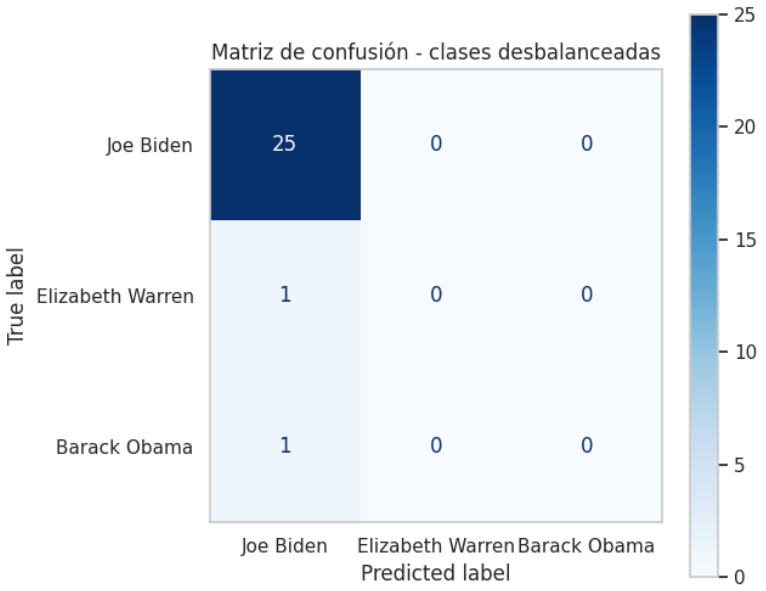

Figura 16: Matriz de confusión para el modelo Multinomial Naive Bayes cuando se utilizan candidatos con pocos discursos

4. Alternativas para la extracción de features de un texto

Una técnica alternativa para extraer características de texto es el uso de embeddings contextuales, siendo BERT (Bidirectional Encoder Representations from Transformers) uno de los más destacados. A diferencia de métodos como TF-IDF, que representan las palabras por su frecuencia sin considerar el contexto, BERT genera vectores numéricos que capturan

el significado de cada palabra según el contexto en el que aparece. Utiliza una arquitectura basada en *Transformers* y analiza el texto en ambas direcciones (izquierda y derecha), lo que le permite construir representaciones profundas y contextuales del lenguaje.

Se espera que el uso de embeddings como BERT mejore considerablemente el rendimiento en tareas de clasificación de texto, especialmente en casos donde el significado de una palabra varía según el contexto o cuando las diferencias entre clases son sutiles. Además, al capturar relaciones semánticas y sintácticas, estos modelos pueden distinguir estilos y enfoques discursivos con mayor precisión que los enfoques basados únicamente en frecuencia. Como contrapartida, requieren mayor poder computacional y tiempos de procesamiento más largos.

5. Conclusión

Este trabajo permitió explorar distintas técnicas de procesamiento y clasificación de texto aplicadas a discursos políticos, destacando tanto su potencia como sus limitaciones. A través del análisis, visualización y evaluación de resultados, se comprendió cómo factores como el desbalance de clases o la representación de texto impactan en el desempeño del modelo.