Introducción a la Ciencia de Datos

Tarea 2

Rossina Primavera
Alina Méndez

En el presente trabajo, se realizará el entrenamiento de distintos modelos de clasificación de texto.

Se comienza el trabajo creando un dataset que contiene datos de las obras de Shakespeare (misma base de datos de la Tarea 1), reduciéndolo a tres personajes (Antony, Cleopatra y Iago). El dataset tendrá el año en que fue publicada la obra, el personaje, el párrafo en el que participa el personaje (que es sometido previamente a una función de limpieza), el título de la obra y el género de la misma.

A partir de este nuevo dataset, se realiza un muestreo estratificado de forma de generar un subconjunto de test y otro de entrenamiento, con una proporción 30/70 respectivamente. Posteriormente, realizamos una visualización que permite ver la distribución de párrafos por personaje en conjuntos de entrenamiento y prueba, de forma de observar la proporción de los conjuntos por cada personaje.

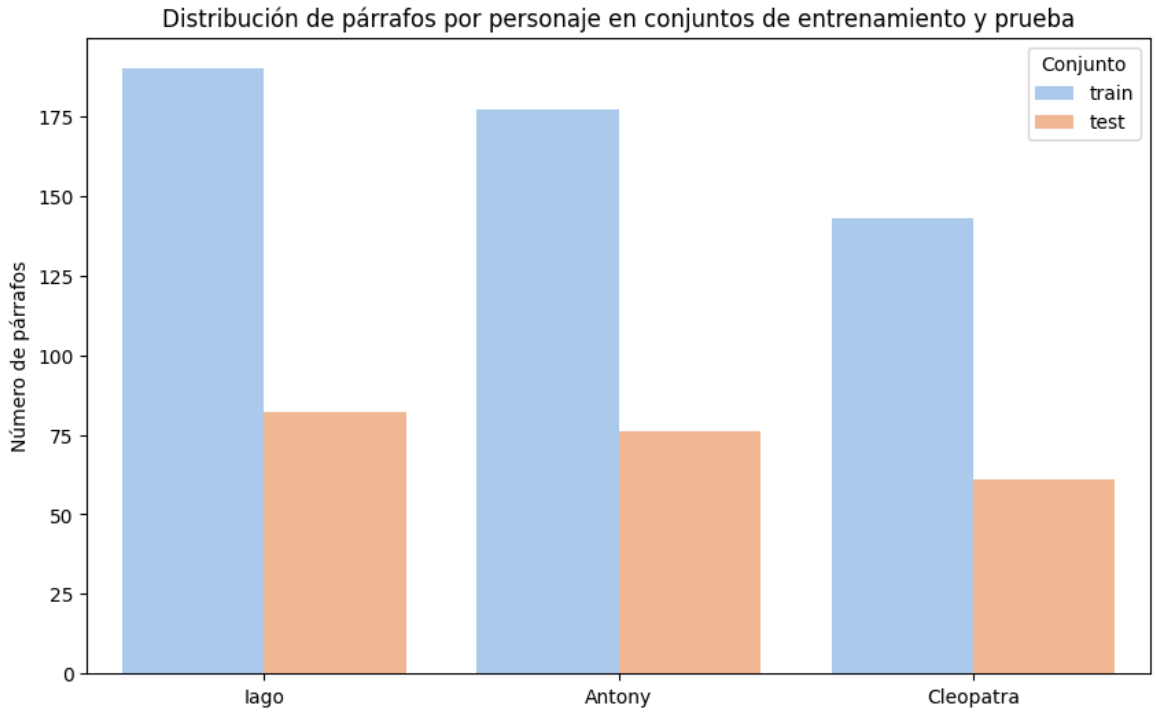


Figura 1: Distribución de párrafos por personaje en conjuntos de entrenamiento y prueba

La siguiente tabla permite visualizar la proporción de párrafos de cada personaje en train del total.

Iago	69.85%
Antony	69.96%
Cleopatra	70.10%

Tabla 1: Proporción de párrafos por personaje en conjunto de entrenamiento

Posteriormente, se busca analizar el balance del muestreo, esto quiere decir, el porcentaje del subconjunto train y test, que tiene cada uno de los personajes. Es de esperar, que un dataset balanceado tenga porcentajes similares entre personajes, en cada subconjunto.

Este punto es de suma importancia, ya que un dataset desbalanceado puede significar un sesgo hacia la clase mayoritaria, y una mala predicción de la clase minoritaria, por no tener suficientes datos de entrenamiento para que el modelo logre aprenderlo de forma óptima.

Debajo se muestra el porcentaje de train y de test para cada uno de los personajes, donde podemos observar que hay diferencias entre los personajes, pero todos se encuentran en el entorno del 30%, por lo que a efectos de este trabajo consideraremos que el dataset es balanceado.

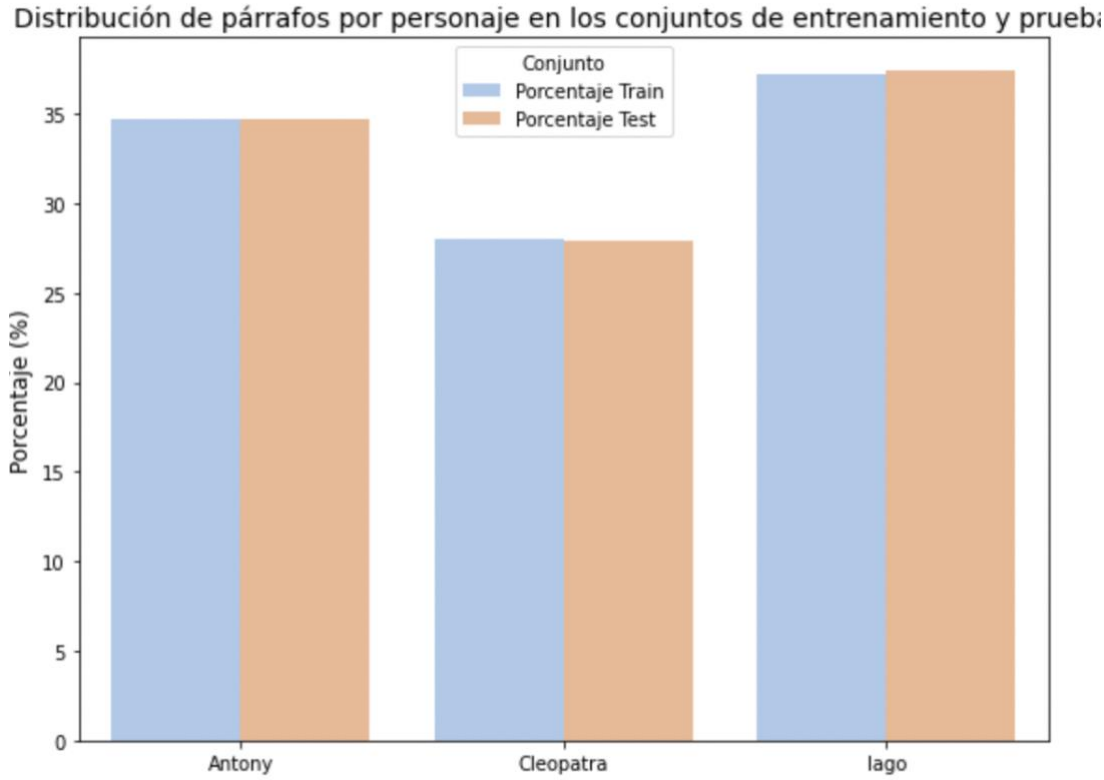


Figura 2: Distribución de párrafos por personaje en conjuntos de entrenamiento y prueba para análisis de balance

Siguiendo con el análisis, realizamos una transformación del conjunto de entrenamiento a la representación numérica (features) de conteo de palabras o bag of words. Esta técnica es una representación de texto en la que un documento se representa como una bolsa (bag) de sus palabras, sin tener en cuenta el orden, pero manteniendo la multiplicidad. Cada documento se convierte en un vector en el que cada elemento representa el número de veces que una palabra aparece en el documento.

Para facilitar la comprensión de esta técnica exponemos un ejemplo:

- Supongamos que hay 3 párrafos y un total de 10 palabras (solo para ilustrar).
- Cada documento puede contener entre 2 y 5 palabras.

Aquí está una posible distribución:

Documento	Palabras
Doc 1	["word1", "word2", "word3"]
Doc 2	["word4", "word2"]
Doc 3	["word5", "word6", "word7"]

Tabla 2: Ejemplo de distribución de palabras en documentos

La matriz BoW resultante será:

	word1	word2	word3	word4	word5	word6	word7	word8	word9	word10
Doc 1	1	1	1	0	0	0	0	0	0	0
Doc 2	0	1	0	1	0	0	0	0	0	0
Doc 3	0	0	0	0	1	1	1	0	0	0

Tabla 3: Ejemplo de matriz BoW

Aquí, la matriz es de tamaño 3x10, y tiene solo 7 elementos no nulos sobre un total de 30, lo que la hace una matriz dispersa.

Volviendo a nuestro dataset de interés, la matriz resultante de la transformación del conjunto de entrenamiento es de tamaño 510 x 2736, es decir, 510 párrafos resultantes y 2736 palabras únicas en todos los párrafos del conjunto de entrenamiento. Dicha matriz es dispersa porque la mayoría de los elementos son ceros, reflejando que cada documento contiene solo una pequeña fracción de las palabras del vocabulario total. En términos de eficiencia de almacenamiento, usar una representación de *sparse matrix* (matriz dispersa) es ventajoso porque se ahorra espacio al almacenar solamente elementos no nulos y sus índices. Además, muchas operaciones matemáticas y algebraicas pueden realizarse más eficientemente debido a la menor cantidad de datos a procesar.

Siguiendo con la aplicación de técnicas, nos parece importante mencionar la existencia de los n-gramas y la representación numérica Term Frequency - Inverse Document Frequency (TF - IDF). Estas técnicas están vinculadas en la forma en que los n-gramas pueden ser utilizados como unidades de análisis antes de aplicar la técnica de TF-IDF para representar documentos de manera más rica y contextualmente relevante. Un n-grama es una secuencia contigua de *n* elementos de un texto o de un discurso. En el contexto del procesamiento de lenguaje natural (NLP), estos elementos son típicamente palabras o caracteres. Los n-gramas se utilizan para capturar la relación entre los elementos en el texto y para modelar el lenguaje de una manera más rica que las representaciones unigramas (basadas en palabras individuales).

A continuación, se procederá a entrenar varios modelos y compararlos entre sí, de forma de concluir cuál es el mejor.

Modelo 1

Primeramente, aplicamos una nueva transformación del conjunto de entrenamiento al generar la representación numérica TF - IDF. Esta técnica de análisis de texto se utiliza para evaluar la importancia de una palabra en un documento dentro de un conjunto de documentos

(corpus). Combina dos métricas: la frecuencia de término (TF) y la frecuencia inversa de documentos (IDF).

Posteriormente, se utiliza la técnica de PCA (Análisis de Componentes Principales) para reducción de dimensionalidad. Los componentes principales son combinaciones lineales de las variables originales que capturan la mayor cantidad de variabilidad en los datos. El primer componente principal captura la mayor parte de la varianza en los datos. El segundo componente principal captura la mayor parte de la varianza restante, y así sucesivamente.

En línea con lo antes expuesto, generamos una visualización sobre el conjunto de entrenamiento, utilizando las dos primeras componentes PCA sobre los vectores de TF - IDF (creados previamente).

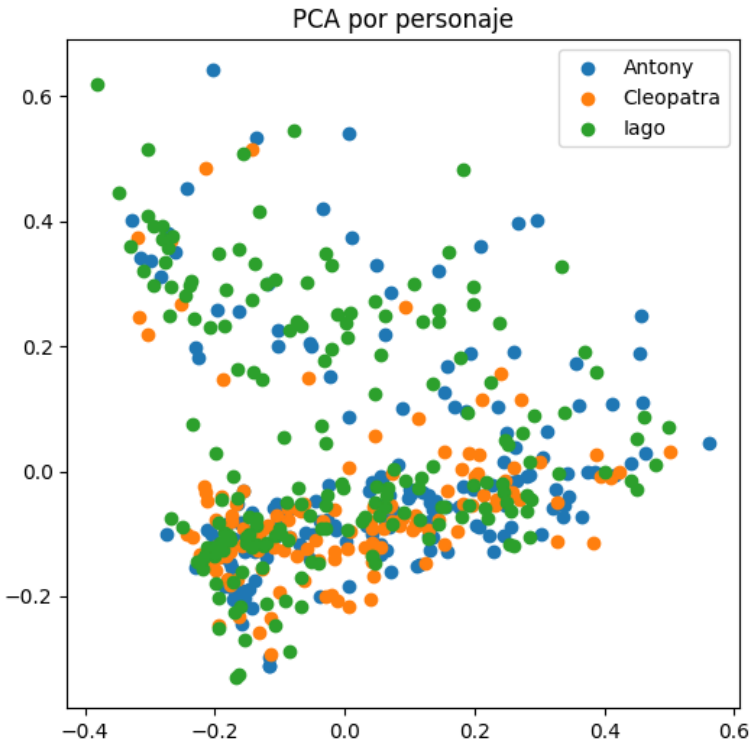


Figura 3: Gráfica de reducción de dimensionalidad del dataset de trabajo para Modelo 1

De la misma, podemos mencionar que la considerable superposición de los puntos sugiere que los estilos de los párrafos entre los personajes no son completamente distintos cuando se representan con TF - IDF y se reducen a las dos primeras componentes principales.

Además, la dispersión de los puntos en todo el gráfico muestra que hay variabilidad significativa en el uso del lenguaje dentro de los párrafos de cada personaje. Por último, cabe destacar que aunque PCA ayuda a reducir la dimensionalidad, la superposición indica que puede ser necesario explorar más componentes principales o utilizar técnicas adicionales para una mejor separación y comprensión de las diferencias estilísticas entre los personajes.

Para complementar este análisis, decidimos agregar más componentes PCA para entender cómo varía la varianza explicada. A continuación presentamos el gráfico generado que permite visualizar esto.

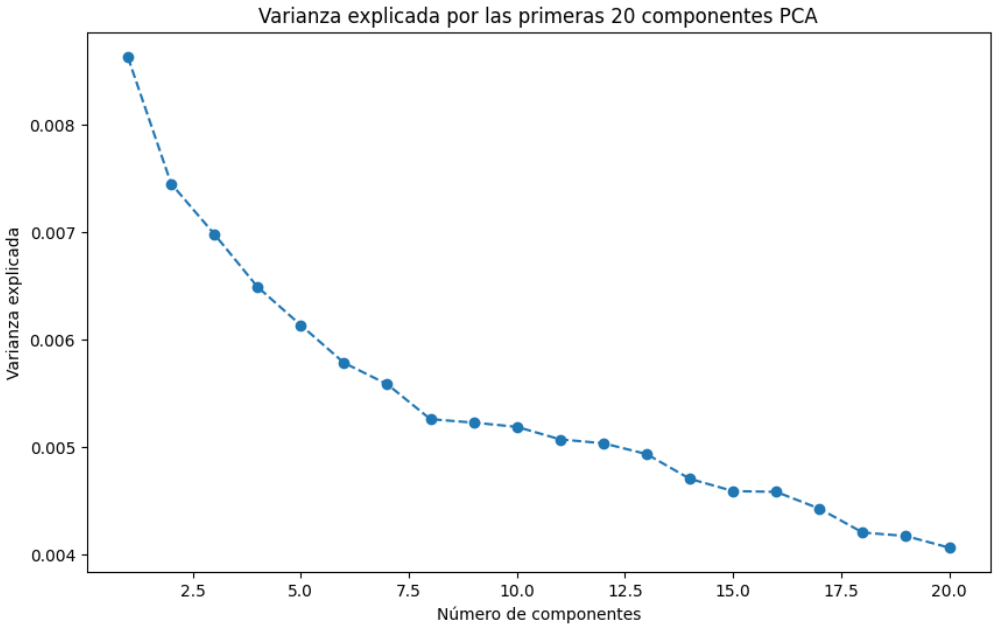


Figura 4: Varianza explicada por las primeras 20 componentes PCA para el modelo 1

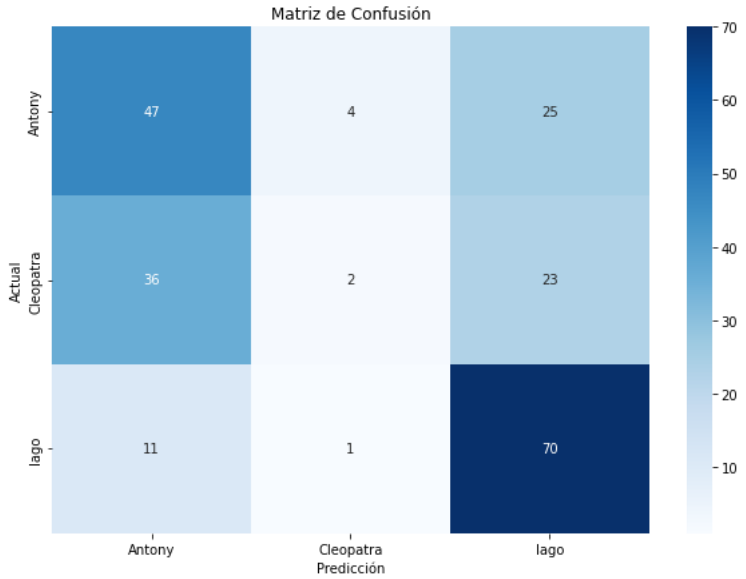


Figura 5: Matriz de confusión para Modelo 1

	Precision	**Recall**	**F1-Score**	**Support**
Antony	0.54	0.61	0.57	76
Cleopatra	0.47	0.46	0.47	61
Iago	0.68	0.62	0.65	82
Accuracy			0.57	219
Macro avg	0.57	0.56	0.56	219
Weighted avg	0.57	0.57	0.57	219

Tabla 1: Métricas del Modelo 1

Modelo 2

A diferencia del modelo anterior, en este caso procedemos a filtrar las stop words del idioma inglés, y presentamos debajo los resultados.

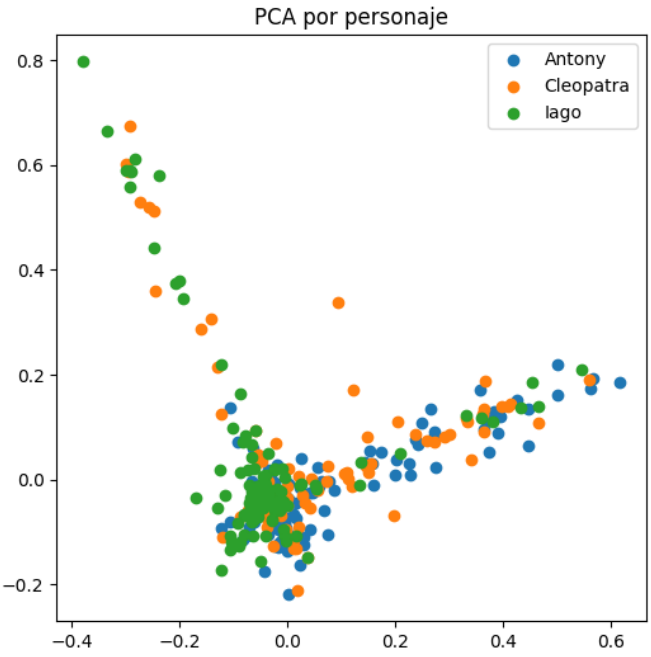


Figura 6: Gráfica de reducción de dimensionalidad del dataset de trabajo para Modelo 2

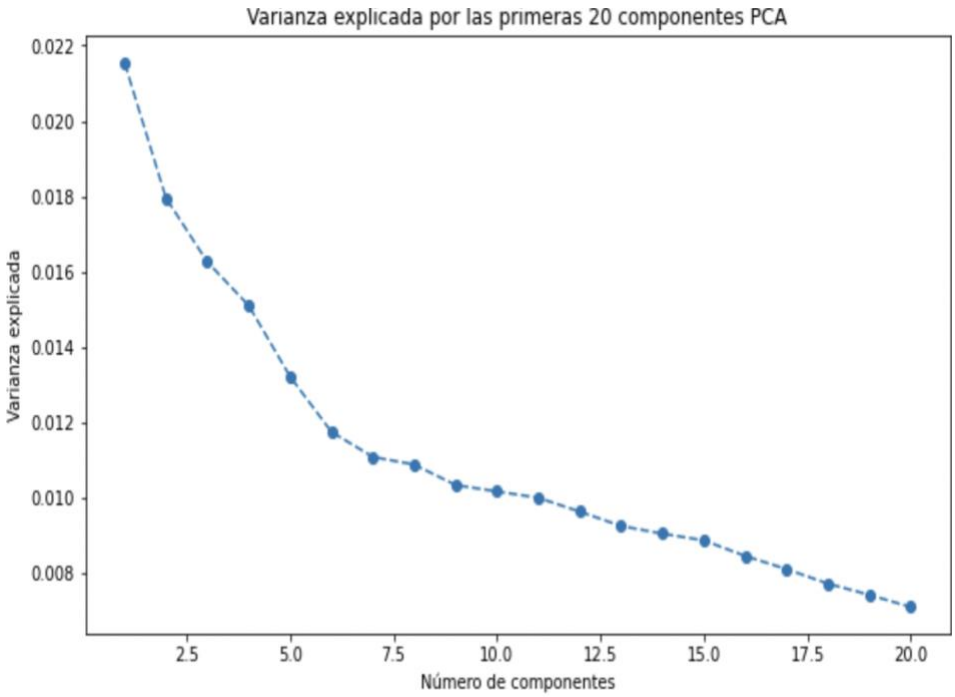


Figura 7: Varianza explicada por las primeras 20 componentes PCA para el modelo 2

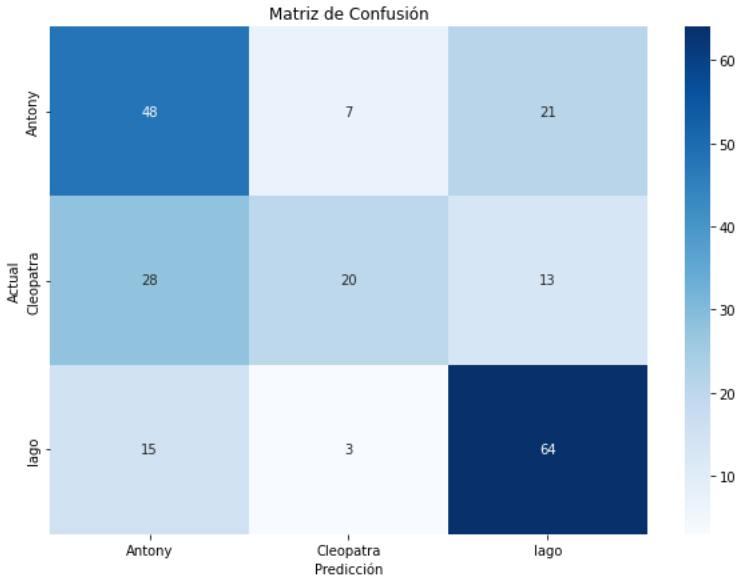


Figura 8: Matriz de confusión para Modelo 2

	Precision	Recall	F1-Score	Support
Antony	0.53	0.63	0.57	76
Cleopatra	0.67	0.33	0.44	61
Iago	0.65	0.78	0.71	82
Accuracy			0.60	219
Macro Avg	0.62	0.58	0.58	219
Weighted Avg	0.61	0.60	0.59	219

Tabla 2: Métricas del Modelo 2

Con estos cambios, vemos una mejora en la separación de los puntos correspondientes a diferentes personajes, gracias al filtrado de stop words y la inclusión de bigramas. Sin embargo, a pesar de la mejora, todavía hay una notable superposición, especialmente en el área central, indicando que los estilos de los personajes aún comparten muchas características comunes.

Este ejercicio resalta la importancia del preprocesamiento del texto (como el filtrado de stop words y el uso de n-gramas) para mejorar la diferenciación de características en análisis de textos.

Modelo 3

En este modelo se plantea el uso de use_idf = “True”, como diferencia al modelo 1. Esta condición mejora la relevancia de los términos raros en el conjunto de documentos, reduciendo la importancia de los términos comunes.

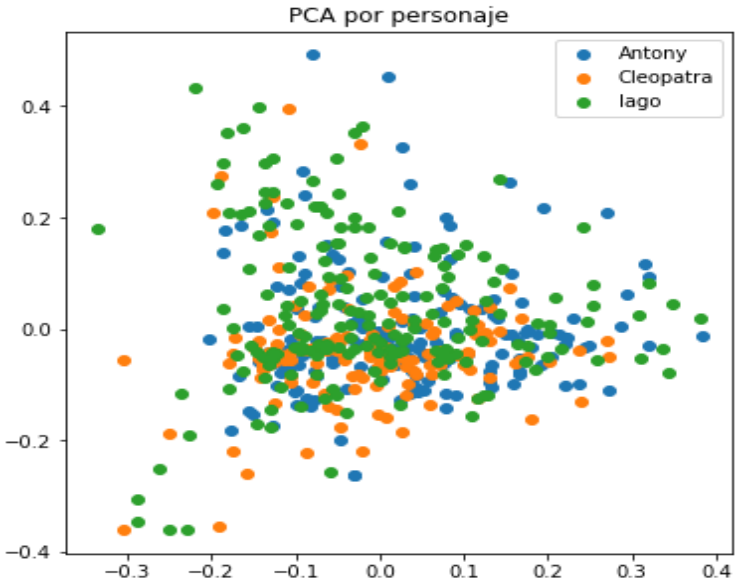


Figura 9: Gráfica de reducción de dimensionalidad del dataset de trabajo para Modelo 3

Con este cambio, no se ven grandes mejoras en la separación de los puntos para cada personaje, parece ser que ahora se encuentran más superpuestos en la área central.

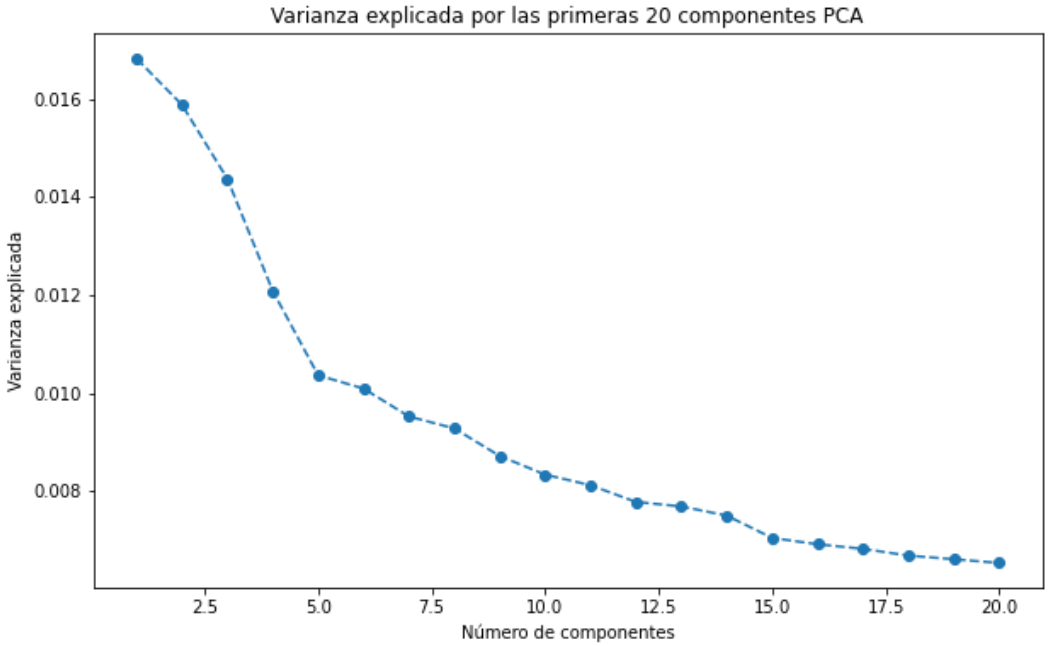


Figura 10: Varianza explicada por las primeras 20 componentes PCA para modelo 3

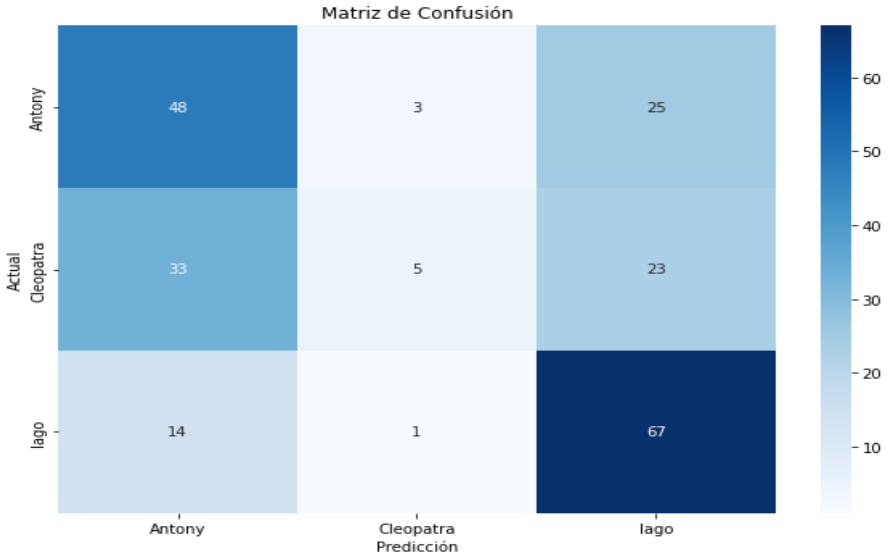


Figura 11: Matriz de confusión para Modelo 3

	Precision	Recall	F1-Score	Support
Antony	0.51	0.63	0.56	76
Cleopatra	0.56	0.08	0.14	61
Iago	0.58	0.82	0.68	82
Accuracy			0.55	219
Macro Avg	0.55	0.51	0.46	219
Weighted Avg	0.55	0.55	0.49	219

Tabla 3: Métricas del Modelo 3

A partir de la matriz de confusión precedente podemos concluir que el personaje "Iago" parece ser el más fácilmente reconocido por el modelo, con 67 instancias correctamente clasificadas. En cambio, los personajes "Antony" y "Cleopatra" tienen más instancias incorrectamente clasificadas como otros personajes. El personaje "Cleopatra" tiene la menor cantidad de verdaderos positivos (TP = 5), lo que indica que el modelo tiene más dificultades para identificar correctamente este personaje.

Modelo 4

Este modelo, cambia respecto al modelo 1, el uso de ngram_range = (1, 2).

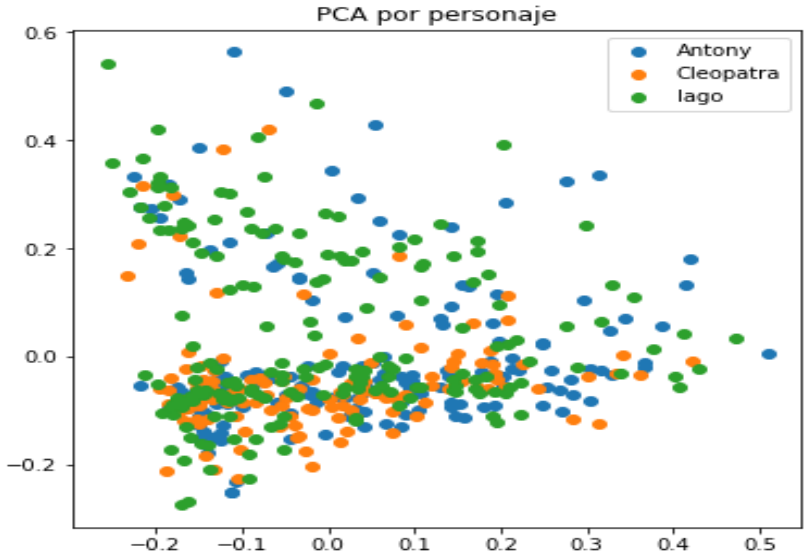


Figura 12: Gráfica de reducción de dimensionalidad del dataset de trabajo para Modelo 4

Al analizar el gráfico vemos que los puntos siguen estando dispersos en todo el gráfico, sin una separación clara entre los diferentes personajes. Esto sugiere que las dos primeras componentes principales no capturan diferencias claras entre los documentos de los tres personajes. Además, persiste el solapamiento considerable entre los puntos de diferentes colores. En particular, hay muchos puntos verdes (Iago) y naranjas (Cleopatra) que se solapan con puntos azules (Antony), lo que indica que es difícil distinguir entre estos personajes basándose solo en estas dos primeras componentes principales. A su vez, cabe mencionar que no se observan clusters bien definidos, lo que podría indicar que los datos de los personajes no se separan claramente en el espacio de estas dos componentes principales.

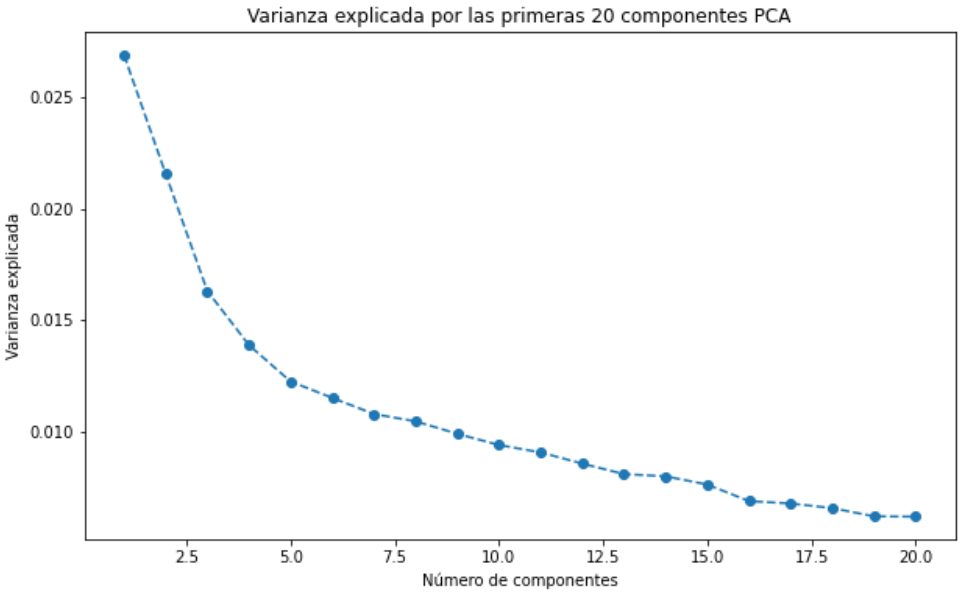


Figura 13: Varianza explicada por las primeras 20 componentes PCA para el modelo 4

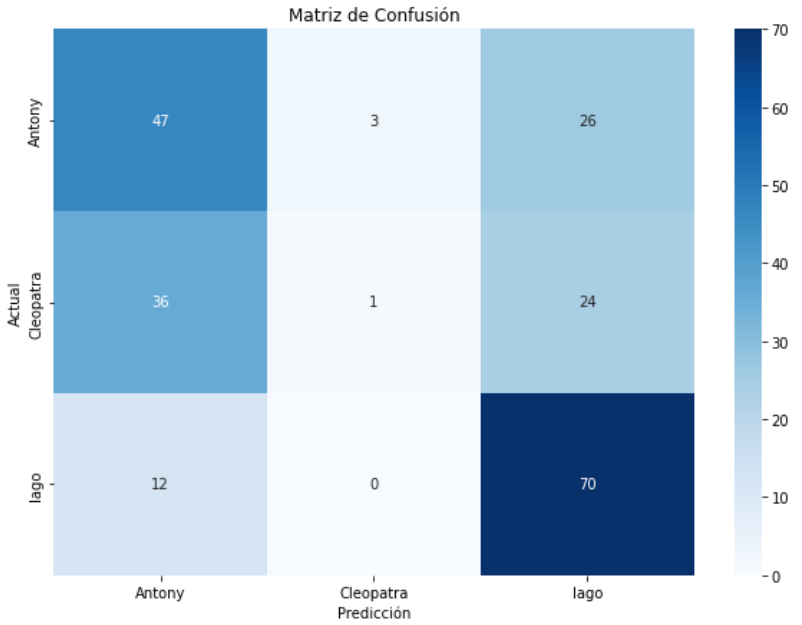


Figura 14: Matriz de confusión para Modelo 4

	Precision	Recall	F1-Score	Support
Antony	0.49	0.62	0.55	76
Cleopatra	0.25	0.02	0.03	61
Iago	0.58	0.85	0.69	82
Accuracy			0.54	219
Macro Avg	0.44	0.50	0.42	219
Weighted Avg	0.46	0.54	0.46	219

Tabla 4: Métricas del Modelo 4

En este modelo, en similitud con el anterior, Iago es el personaje más reconocido, dado que tiene 70 instancias correctamente clasificadas. Le sigue el personaje Antony, con 47 instancias correctamente clasficadas. En cambio, Cleopatra es el personaje menos reconocido por el modelo al tener solo 1 instancia correctamente clasificada.

El gráfico debajo muestra la varianza explicada por las primeras 20 componentes PCA para los 4 modelos expuestos. Cabe destacar del mismo que el primer modelo captura la mayor proporción de varianza en las primeras componentes en comparación con los otros modelos. La primera componente principal por sí sola explica más del 4% de la varianza.

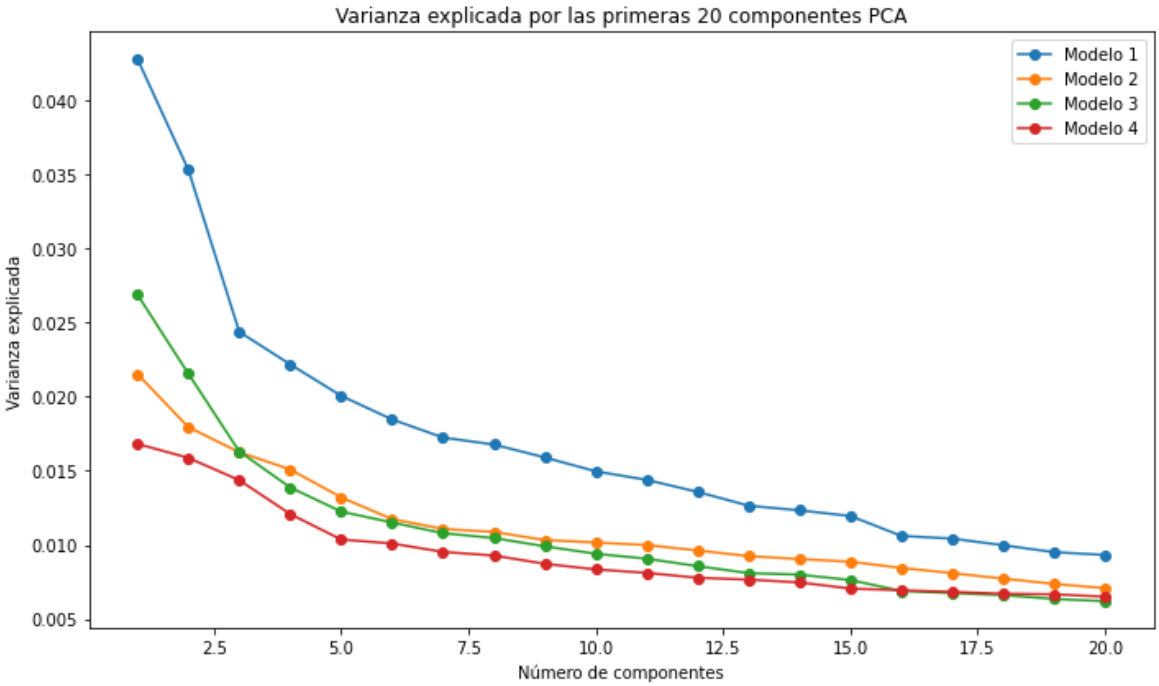


Figura 15: Varianzas explicadas por las primeras 20 componentes PCA de los modelos 1 al 4

Modelo 5

Se aplica un modelo SVC, estos modelos vienen de la familia de los SVM, que son un conjunto de métodos de aprendizaje supervisado utilizados para clasificación, regresión y detección de valores atípicos. La idea principal es encontrar un hiperplano en un espacio de alta dimensión que separe las clases de manera óptima.

SVC es una versión de SVM enfocada en problemas de clasificación. Su objetivo es encontrar el hiperplano que maximiza el margen entre las clases. Los puntos de datos más cercanos al hiperplano se llaman vectores de soporte.

Este modelo es un modelo flexible adecuado para una variedad de aplicaciones complejas, mientras que Multinomial Naive Bayes es simple, eficiente y particularmente efectivo para tareas de clasificación de texto.

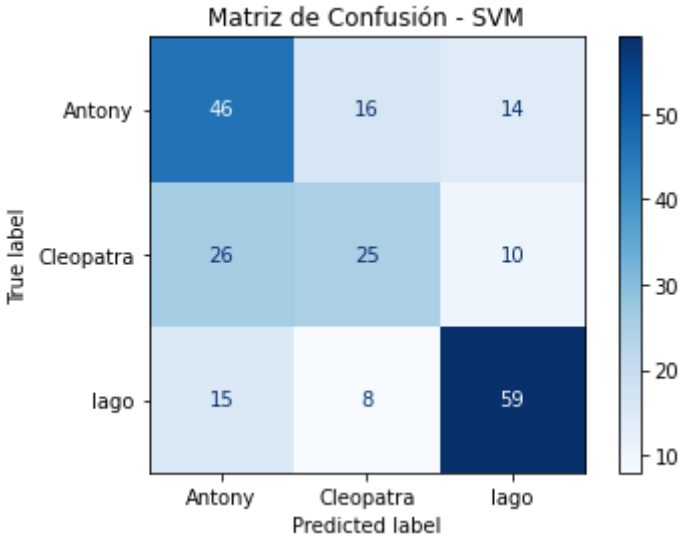


Figura 16: Matriz de confusión para Modelo 5

	Precision	Recall	F1-Score	Support
Personaje 1	0.53	0.61	0.56	76
Personaje 2	0.51	0.41	0.45	61
Personaje 3	0.71	0.72	0.72	82
Accuracy			0.59	219
Macro Avg	0.58	0.58	0.58	219
Weighted Avg	0.59	0.59	0.59	219

Tabla 5: Métricas del Modelo 5

Ahora, la matriz de confusión precedente muestra que si bien Iago sigue siendo el personaje más reconocido por el modelo, disminuyen las instancias correctamente explicadas a 59, en comparación con los anteriores dos modelos. El personaje Antony permanece prácticamente sin cambios, mientras que el personaje Cleopatra mejora sustancialmente respecto a los dos modelos anteriores al alcanzar 25 instancias correctamentes explicadas a partir del modelo.

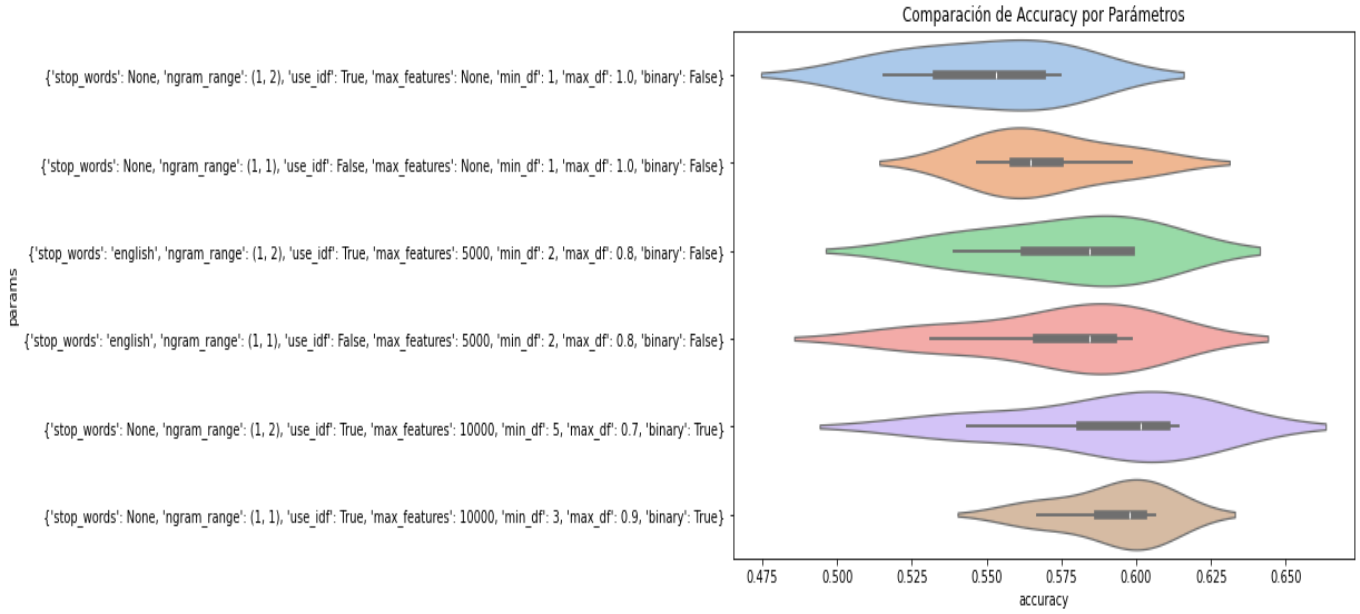
Modelo 6

Adicionalmente, efectuamos la búsqueda de hiper-parámetros con Cross Validation. Esta técnica es utilizada en el aprendizaje automático y la estadística para evaluar la capacidad de generalización de un modelo predictivo. Su objetivo es garantizar que el modelo no solo funcione bien en los datos de entrenamiento, sino que también se desempeñe adecuadamente en datos no vistos (de prueba). La misma consiste en dividir los datos disponibles en múltiples subconjuntos, entrenar el modelo en algunos de estos subconjuntos y validar su rendimiento en los subconjuntos restantes. Este proceso se repite varias veces con diferentes particiones de los datos para obtener una evaluación más robusta y menos dependiente de una sola división de datos.

Al código proporcionado por los profesores le agregamos las sentencias que faltaba completar para poder efectuar la búsqueda de hiper-parámetros con Cross Validation. En este

sentido, se inicializa un diccionario results para almacenar las precisiones obtenidas para cada conjunto de parámetros. En cada iteración de la validación cruzada, se calcula la precisión y se agrega al diccionario correspondiente a los parámetros actuales.

A su vez, los datos de precisión recopilados se convierten en una lista de tuplas que contiene los parámetros y las precisiones correspondientes. Esta lista se convierte en un DataFrame de pandas para facilitar la visualización con seaborn. Se utiliza seaborn para crear un gráfico de violín que muestre la distribución de las precisiones para cada conjunto de parámetros.



-

Figura 17: Accuracy de los parámetros del modelo 6

El parámetro n° 5 (stop words = none, ngram:range (1,2), use_idf = true) es elegido como el mejor dado que alcanza el valor más alto de precisión. Usaremos este parámetro para volver a entrenar el modelo sobre todo el conjunto de entrenamiento disponible.

El valor final de las métricas al haber seleccionado el mejor parámetro es:

- Accuracy: 0.77
- Precisión: 0.79
- Recall: 0.77
- F1-score: 0.76

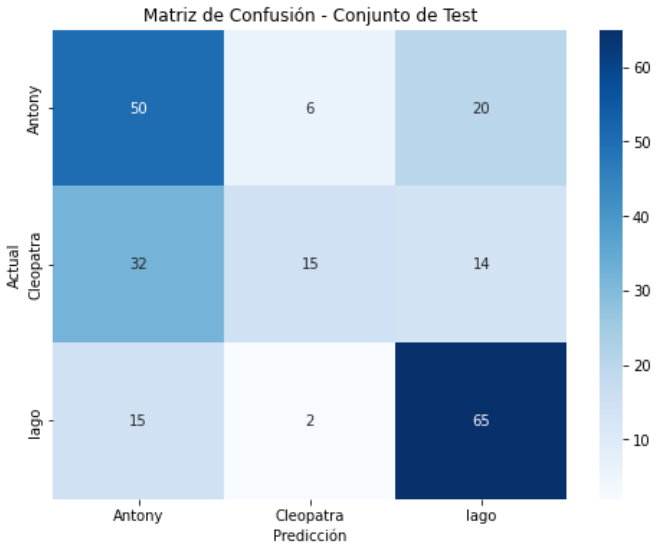


Figura 18: Matriz de confusión para Modelo 6

Bag-of-Words y TF-IDF son métodos poderosos y ampliamente utilizados en el procesamiento de texto, es importante estar consciente de sus limitaciones en términos de captura de significado contextual y semántico profundo. Para escenarios donde se requiere una comprensión más avanzada del texto, pueden ser necesarios enfoques más sofisticados como modelos basados en redes neuronales y embeddings de palabras.

Técnicas alternativas a BoW o TF-IDF

Una técnica alternativa popular para extraer características de texto es el uso de Word Embeddings (Incrustaciones de Palabras). A diferencia de las representaciones tradicionales como Bag-of-Words (BoW) o TF-IDF, que generan matrices dispersas al contar las ocurrencias de palabras o combinaciones de palabras, los embeddings de palabras representan palabras como vectores densos de números reales.

Funcionamiento de Word Embeddings:

- **Representación Semántica:** Los word embeddings capturan el significado semántico y las relaciones entre palabras. Cada palabra se asigna a un vector de alta dimensionalidad, donde la proximidad entre los vectores refleja la similitud semántica entre las palabras.
- **Entrenamiento:** Los embeddings de palabras pueden ser aprendidos de manera supervisada o no supervisada. Métodos no supervisados como Word2Vec, GloVe o FastText aprenden a partir de grandes corpus de texto para predecir palabras vecinas o contextuales. En métodos supervisados, los embeddings pueden ser ajustados específicamente para una tarea, como la clasificación de documentos.

Beneficios:

- **Generalización Mejorada:** Los embeddings capturan relaciones semánticas, posicionando palabras con significados similares cerca en el espacio vectorial.
- **Reducción de Dimensionalidad:** Los embeddings suelen tener una dimensionalidad menor que las representaciones BoW o TF-IDF, facilitando el entrenamiento de modelos y mejorando la eficiencia computacional.

Ventajas sobre BoW y TF-IDF:

- **Mejora en la precisión del Modelo:** Capturan mejor el significado contextual de las palabras, mejorando el rendimiento en tareas de procesamiento de lenguaje natural como la clasificación de texto.
- **Capacidad de Generalización:** Manejan mejor palabras que no aparecen en el conjunto de entrenamiento, gracias a su representación continua y densa.
- **Eficiencia en el Uso de Recursos:** A pesar de tener alta dimensionalidad, su representación densa permite un procesamiento más eficiente y una mejor captura de relaciones semánticas complejas.

Modelo de Word Embeddings:

- Un modelo de word embeddings está formado por un vocabulario (conjunto de palabras distintas en una colección de textos) y sus respectivos embeddings (vectores).
- Necesita ser entrenado a partir de una colección de textos lo suficientemente grande para aprender buenas representaciones de las palabras. Un ejemplo sería la colección de textos de Wikipedia en inglés (mil millones de palabras distintas).
- Los word embeddings muestran buenos resultados en muchas aplicaciones de procesamiento de lenguaje natural, capturando relaciones como sinonimia, capitales y países, o singular y plural.

Ventajas y Desventajas de Utilizar FastText

Ventajas:

1. **Captura de Información Subpalabra:** FastText representa palabras individuales como vectores y también considera subpalabras (n-gramas de caracteres), permitiendo manejar palabras fuera del vocabulario conocido o palabras compuestas.
2. **Manejo de Palabras Raras:** Representa palabras con subpalabras, mejorando el manejo de palabras raras o poco frecuentes.
3. **Eficiencia en Clasificación de Texto:** FastText es conocido por su eficiencia computacional debido a su representación densa y de baja dimensionalidad.
4. **Modelos Preentrenados:** FastText ofrece modelos preentrenados en grandes corpus de texto en varios idiomas, útil si no se dispone de suficientes datos para entrenar desde cero.

Desventajas:

1. **Mayor Complejidad Computacional:** Requiere más recursos computacionales en comparación con modelos más simples como Multinomial Naive Bayes.
2. **Requiere Más Datos de Entrenamiento:** Necesita más datos para obtener representaciones de subpalabras significativas.
3. **Interpretación de Resultados:** La interpretación de los vectores de FastText puede ser más compleja que las características explícitas de BoW o TF-IDF.

En resumen, utilizar FastText puede ser ventajoso cuando se busca mejorar la precisión y el manejo de palabras raras en tareas de clasificación de texto, a costa de una mayor complejidad computacional y requerimientos de datos de entrenamiento. La elección entre

FastText y modelos tradicionales depende de las características específicas del problema y de los recursos disponibles.

Modelo 7

En línea con lo expuesto antes, se entrena un modelo supervisado de FastText, y se obtienen los siguientes resultados.

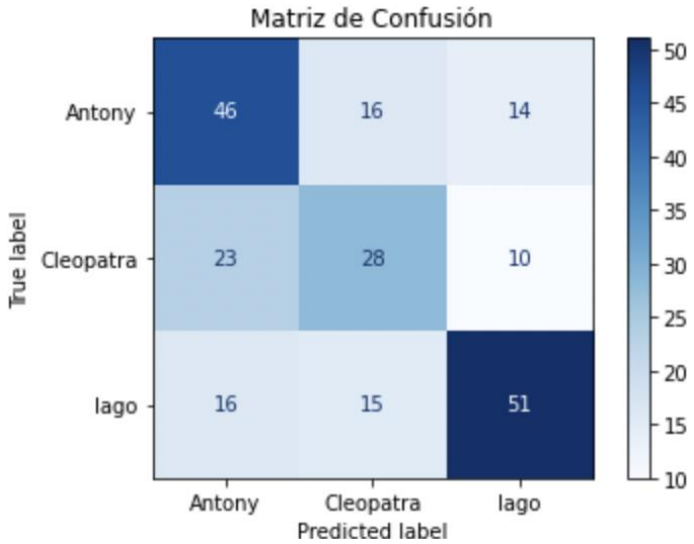


Figura 19: Matriz de confusión para Modelo 7

	Precision	Recall	F1-Score	Support
Antony	0.54	0.61	0.57	76
Cleopatra	0.47	0.46	0.47	61
Iago	0.68	0.62	0.65	82
Accuracy			0.57	219
Macro Avg	0.57	0.56	0.56	219
Weighted Avg	0.57	0.57	0.57	219

Tabla 6: Métricas del Modelo 7

Podemos observar una precision menor a la del modelo de cross validation. Sin embargo se observan mejoras en el personaje de Cleopatra, que pasa de 15 a 28 aciertos. En este caso, entendiendo que el modelo requiere un mayor costo computacional, consideramos que el mejor es el multinomial entrenado con cross validation (modelo 6).

Modelo 8 - desbalanceo

Para terminar el análisis, se procede a realizar los mismos modelos pero con un dataset desbalanceado. En este caso se eligen los personajes de Falstaff, Cleopatra y Juliet, quedando la proporción de la forma que se muestra debajo.

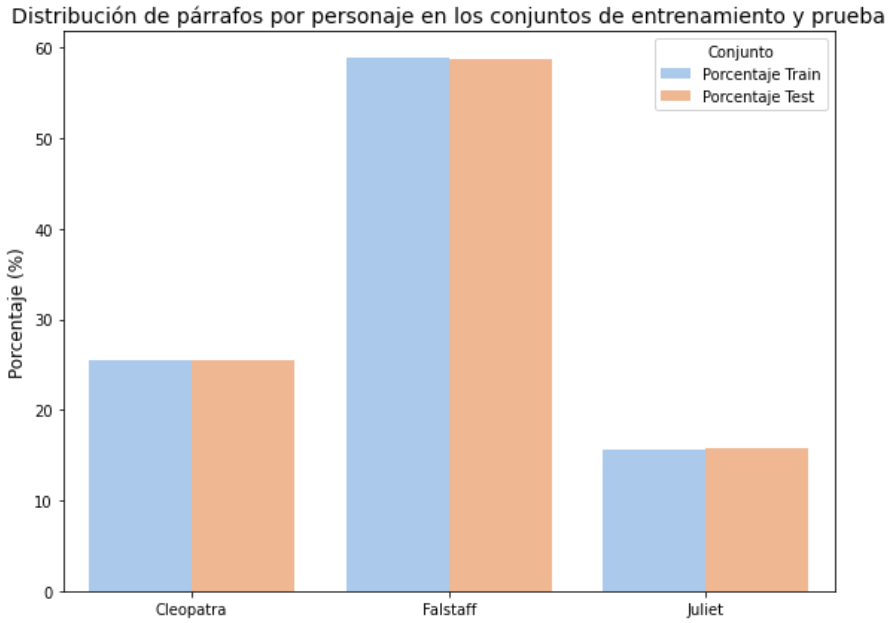


Figura 20: Distribución de párrafos por personaje en conjuntos de entrenamiento y prueba con desbalance

Se entrenan todos los modelos que se entrenaron para el set de personajes anterior, pero se presenta a continuación el modelo 1, de forma de ilustrar el impacto del desbalanceo y no profundizar en el entrenamiento del modelo.

La matriz de confusión resultante, deja claro el sesgo sobre Falstaff, donde la precisión para los personajes minoritarios se ve afectada en gran medida, además de generar un alto número de falsos positivos al personaje mayoritario.

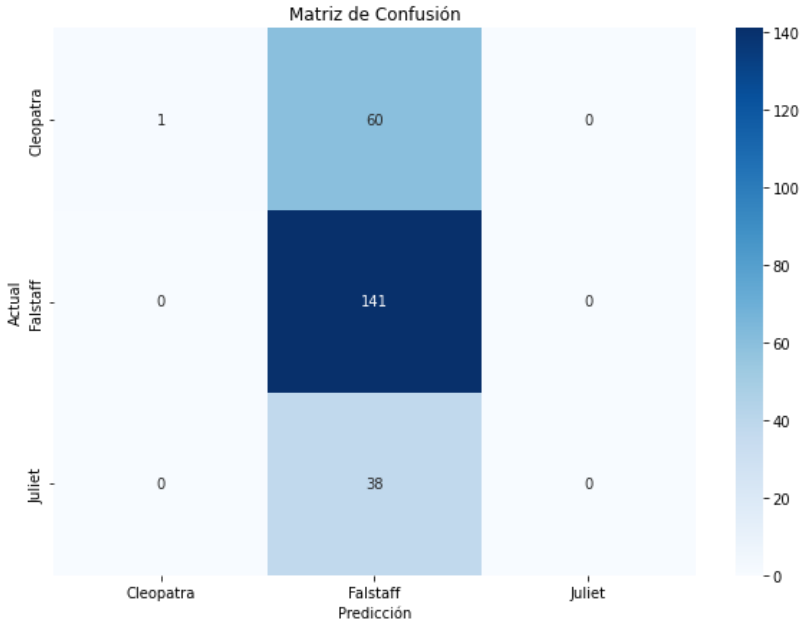


Figura 21: Matriz de confusión para Modelo 8

	Precision	Recall	F1-Score	Support
Cleopatra	1.00	0.02	0.03	61
Falstaff	0.59	1.00	0.74	141
Juliet	0.00	0.00	0.00	38
Accuracy			0.59	240
Macro Avg	0.53	0.34	0.26	240
Weighted Avg	0.60	0.59	0.44	240

Tabla 7: Métricas del Modelo 8