María Eugenia Curi
Lucía Osorio

Tarea 2 - Introducción a la Ciencia de Datos

Introducción

En el presente trabajo se continúa la exploración de la base de datos de Shakespeare, haciendo énfasis en los textos que se encuentran en la tabla paragraphs. En este caso, la tabla paragraphs contiene la información que se muestra en la Figura 1.

paragraphs	
id	int
ParagraphNum	int
PlainText	text
character_id	int
chapter_id	int

Fig.1. Estructura de la tabla Paragraphs de Shakespeare

Se realiza una limpieza de los textos que se encuentran en PlainText, utilizando la misma función clean_text() que se usó en la Tarea 1, de forma de explorar los personajes por palabras. El análisis se realizará, mayoritariamente, para 3 personajes: Antony, Cleopatra y Queen Margaret.

En la siguiente sección se presentará el preprocesamiento de los datos, a partir de la descripción de algunas técnicas de transformación de texto como Bag-of-Words y TF-IDF. Posteriormente, se evaluarán distintos modelos de aprendizaje automático, como el Multinomial Naive Bayes y Support Vector Machines, para clasificar los textos por personaje. Finalmente, se abordarán métodos para mejorar el balance de los datos y se explorarán técnicas avanzadas de procesamiento de texto.

Preprocesamiento y exploración de los datos

Muestreo estratificado

Se realizó un muestreo estratificado antes de aplicar métodos de aprendizaje automático para asegurar la representatividad de los datos. Como resultado, se obtuvo un conjunto de datos de entrenamiento compuesto por el 70% de los párrafos (correspondiente a 438) y un conjunto de prueba con el 30% restante (188 párrafos).

Al analizar los datos, se observa que el personaje con mayor cantidad de párrafos es Antony, seguido de Cleopatra y, finalmente, Queen Margaret. Además, se asegura un equilibrio en la cantidad de párrafos entre los conjuntos de entrenamiento y prueba para cada personaje. Este balance se refleja en los porcentajes correspondientes a cada intervalo.

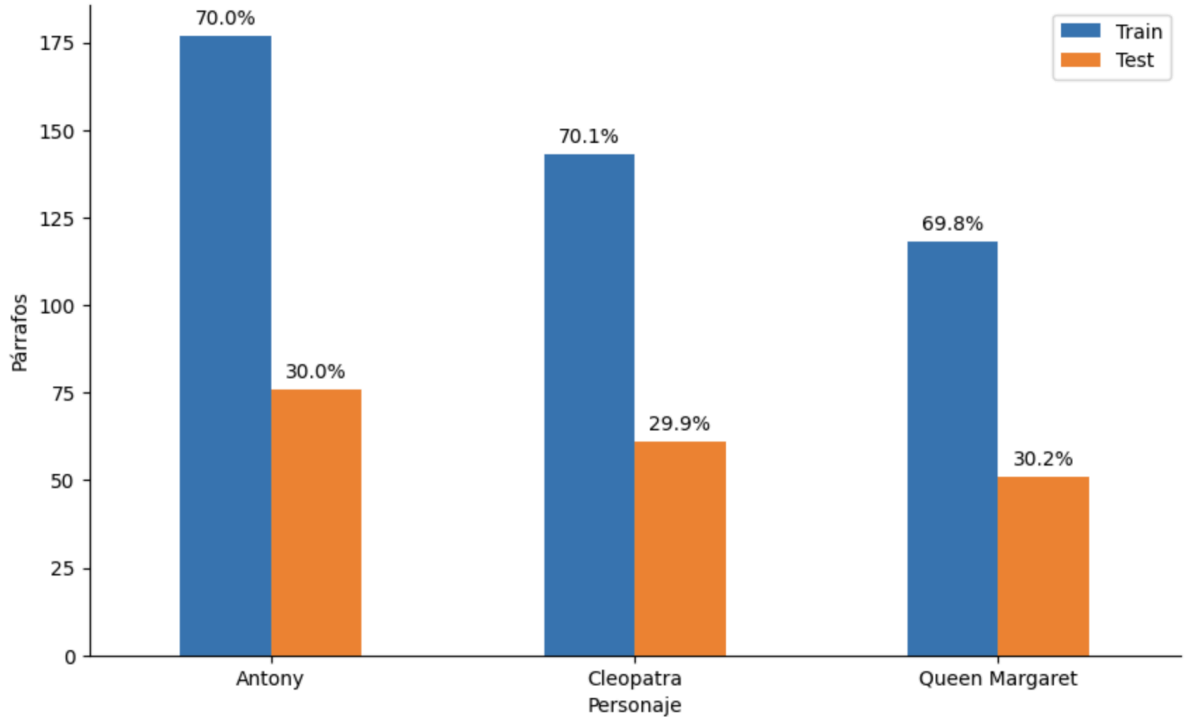

Fig 2. Balance de párrafos por personaje en conjunto de entrenamiento y test

Dado que se va a realizar un análisis de los párrafos por personaje, va a ser pertinente una correcta transformación de los mismos, en este caso por ser textos puede resultar útil la división en unidades más pequeñas (tokenización). Esta preparación de los datos permite que sean más manejables para los algoritmos de análisis.

Bag-of-Words (BoW)

Una de las formas más intuitivas de transformar el texto es representándolo como una bolsa de palabras, conocido como Bag-of-Words en inglés. Este método implica un conjunto desordenado de palabras donde no importa su posición, pero sí la frecuencia de cada palabra en el texto (Jurafsky, D., & Martin, J. H., 2019). Esta técnica convierte el texto en una matriz de conteo de palabras a través de todo el conjunto de datos o corpus, donde cada palabra se representa por la cantidad de veces que aparece en cada línea.

Usando los datos de entrenamiento y tomando como ejemplo las últimas 12 palabras, podemos observar que en el párrafo 75 "*youth*" aparece una vez, y en el 84 la palabra "*yield*" aparece una vez, mientras que la palabra "york" aparece dos veces en el párrafo 62 y una vez en el 86. La Figura 3 es la representación del ejemplo anterior.

	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86
yes	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yesterday	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yield	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
yielded	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yoke	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yond	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
yonder	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
york	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
yorks	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
young	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
younger	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
youth	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Fig 3. Ejemplo de representación de palabras en el corpus utilizando el método bag-of-words.

Al aplicar la técnica Bag-of-Words en el conjunto de entrenamiento, se obtiene una matriz de 438 filas y 2591 columnas, con 5946 elementos almacenados. Como se puede ver en el ejemplo reducido de la Figura 3, esta matriz contiene principalmente ceros, lo que la convierte en una matriz dispersa (sparse matrix en inglés).

En los datos de entrenamiento la palabra más frecuente es “thou” con 157 ocurrencias. Por otro lado, la palabra con más ocurrencia por párrafo es “thy” obteniendo 13 apariciones en párrafo. En la siguiente figura se observan las palabras con mayor frecuencia en un solo párrafo:

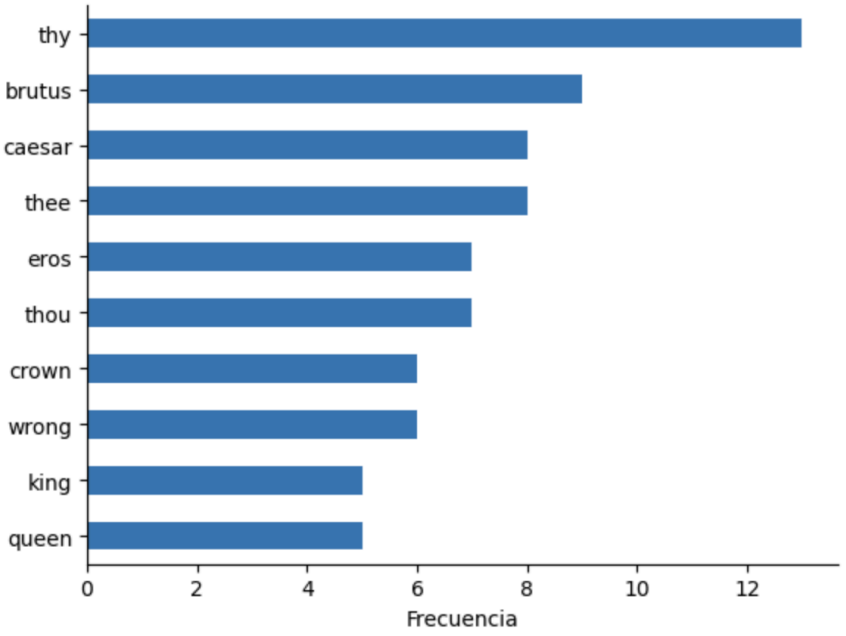

Fig 4. Palabras más frecuentes en un solo párrafo utilizando el método bag-of-words.

N-grama

Un n-grama es una secuencia de n elementos que pueden ser caracteres, tokens o palabras, lo cual ayuda a entender el contexto de los elementos y su forma más frecuente de agruparse.

En caso de tener un elemento se denominan Unigrama (1-grama), si contienen dos elementos se llama Bigrama (2-grama). En este último caso la matriz resultante se obtiene uniendo pares de palabras, a partir del conjunto de datos de entrenamiento, en la siguiente imagen se puede observar las agrupaciones de palabras.

	70	71	72	73	74	75
young man	0	0	0	0	0	0
young octavius	0	0	0	0	0	0
young prince	0	0	0	0	0	0
young roman	0	0	0	0	0	0
younger brown	0	0	0	0	0	0
youth behold	0	0	0	0	0	0
youth cut	0	0	0	0	0	0
youth like	0	0	0	0	0	0
youth means	0	0	0	0	0	1
youth world	0	0	0	0	0	0

Fig 5. Representación de la frecuencia de términos agrupados en bigramas

Term Frequency times Inverse Document Frequency

La frecuencia de palabras, tal como vimos con Bag-of-Words, no es la mejor medida de asociación entre palabras ya que es sesgada y poco discriminativa (Jurafsky, D., & Martin, J. H., 2019). Para saber sobre los contextos donde aparecen las palabras podremos utilizar la transformación “Term Frequency times Inverse Document Frequency” (TF-IDF).

La primera parte “Term Frequency” (TF) hace referencia a la cantidad de veces que aparece una palabra en un documento, ponderado por el largo del documento. Por otro lado, la "Frecuencia Inversa de Documento" (IDF) otorga mayor peso a los términos que aparecen en menos ocasiones en el corpus, considerándolos más relevantes.

Para el conjunto de entrenamiento podemos observar que la palabra “*york”* en la línea 61 aparece dos veces y solamente una vez en la línea 86. Sin embargo es una palabra que aparece 13 veces en todo el corpus, por lo tanto el valor TF-IDF es 0.408248 y 0.109109 respectivamente. La palabra “*yield”* aparece 4 veces en todo el corpus, por lo tanto en la línea 84 que aparece una vez, se representa con el valor 0.188982 en la matriz TF-IDF. Esto muestra cómo se pondera distinto según la frecuencia de palabras en todo el texto.

	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86
yes	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000
yesterday	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000
yield	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.188982	0.0	0.000000
yielded	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000
yoke	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000
yond	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000
yonder	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000
york	0.408248	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.109109
yorks	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000
young	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000
younger	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000
youth	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.152499	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000

Fig 6. Ejemplo de representación de palabras en el corpus utilizando el método TF-IDF.

La matriz TF-IDF es también una matriz dispersa, similar a la matriz BoW, ya que la mayoría de sus entradas son ceros. Esto se debe a que cada documento contiene solo un pequeño subconjunto de las palabras del vocabulario total del corpus.

Análisis de componentes principales (PCA)

El PCA es una técnica que reduce la dimensionalidad de los datos, permitiendo visualizar patrones y relaciones entre las muestras. Los principales usos son la extracción de la información más importante, la compresión del tamaño del conjunto de datos manteniendo sólo esta importante información, la simplificación de la descripción del conjunto de datos y el análisis del mismo (Abdi, H., & Williams, L. J., 2010).

Se realiza el análisis de las dos componentes principales para distintos ejemplos. En las figuras 7 (7.1, 7.2 y 7.3) se muestran distintas exploraciones.

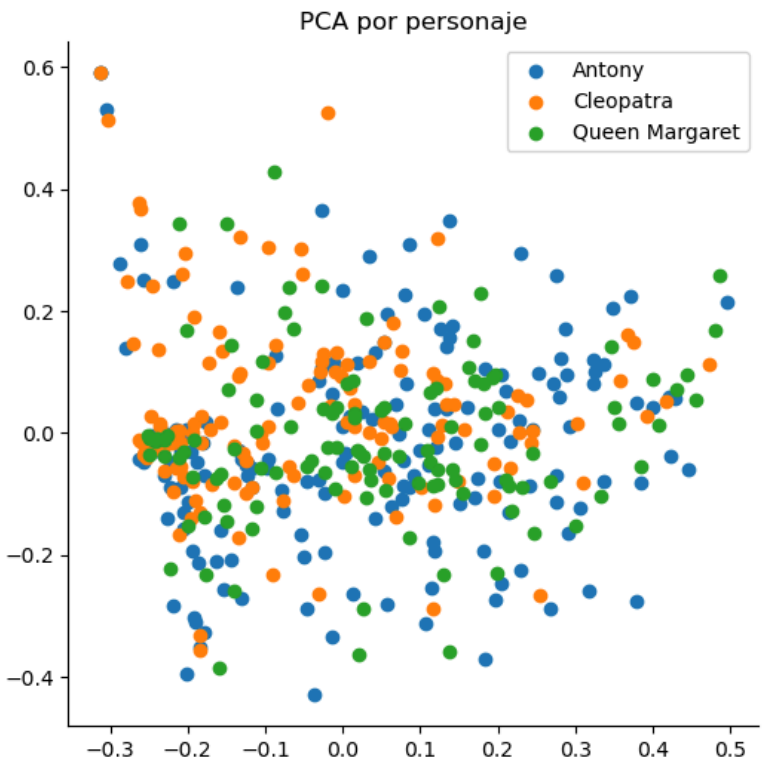

Fig 7.1 Representación de primeras 2 componentes principales

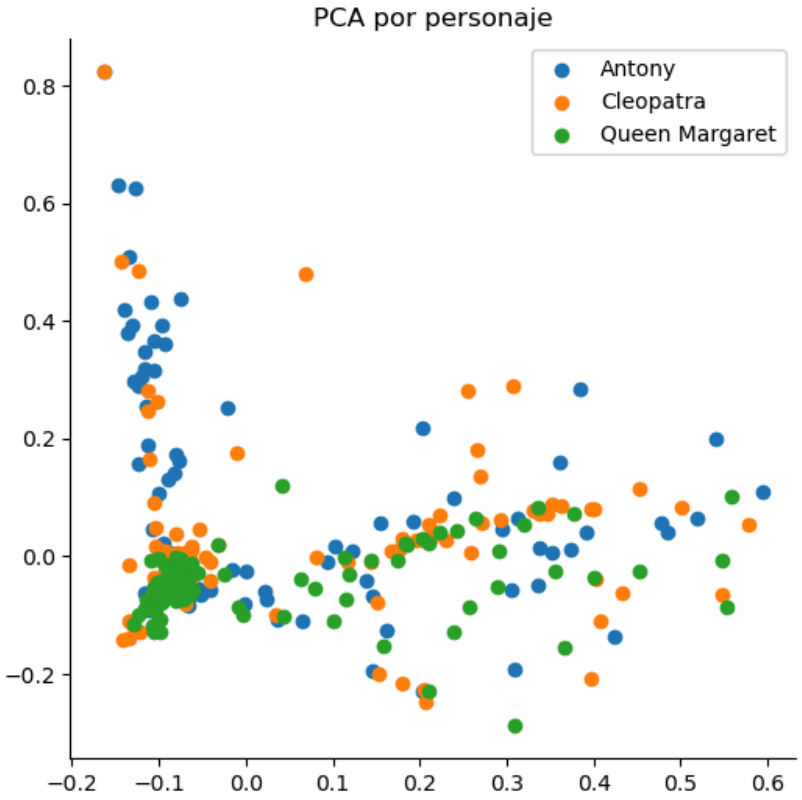

Fig 7.2 Representación de primeras 2 componentes principales sin stop words

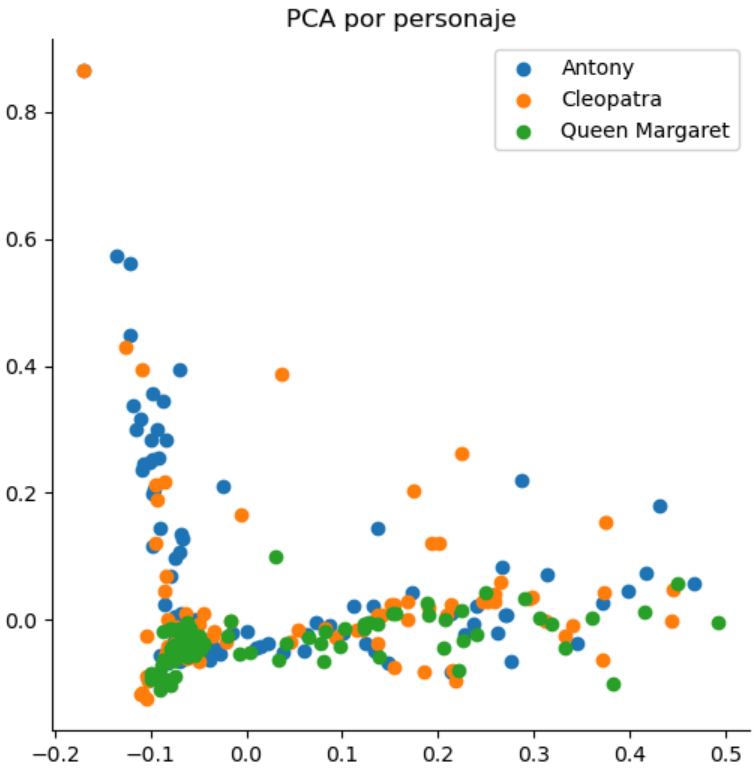

Fig 7.3 Representación de primeras 2 componentes principales sin stop words y mayor n-grama

En las representaciones anteriores no se observa una diferenciación consistente entre los personajes utilizando este método, lo cual se evidencia en el solapamiento casi total de los 3 personajes. Sin embargo, la remoción de las stop-words del inglés y el aumento del tamaño del n-grama (utilizando unigrama y bigrama) parecería mostrar más diferencias entre el personaje de Anthony y Queen Margaret. Si bien la discriminación entre personajes no es perfecta, la reducción de la dispersión y por tanto el aumento de la agrupación de los datos puede significar que con este procesamiento podemos encontrar diferencias en los estilos o contenidos de los textos de los personajes.

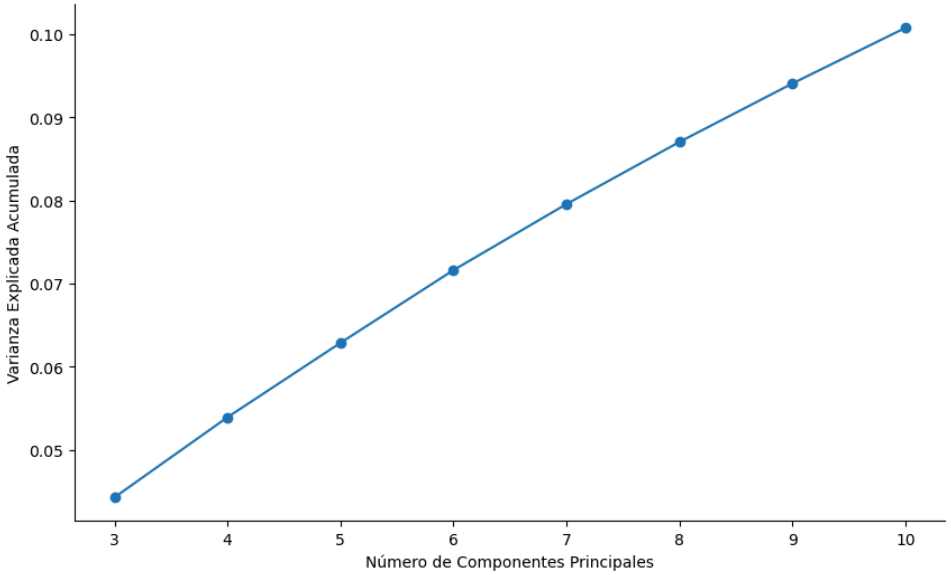

Fig 8. Varianza explicada acumulada por número de componentes principales

Como se observa en la figura 8, la inclusión de más componentes principales incrementa la varianza explicada, pero el incremento es pequeño. Esto puede indicar que la mayoría de la varianza en los datos no se captura con los primeros 10 componentes principales, y puede ser necesario incluir muchos más componentes para explicar una cantidad significativa de la varianza total.

Entrenamiento y evaluación del modelo

Multinomial Naive Bayes

Un modelo Multinomial Naive Bayes es un tipo de modelo de clasificación probabilístico basado en el teorema de Bayes con la suposición "naive" o ingenua de independencia condicional entre las características. Este modelo es particularmente útil cuando se trabaja con características discretas (como recuentos de palabras en textos), típicamente en aplicaciones como la clasificación de textos o documentos. En este caso lo utilizaremos para predecir el personaje en base a sus textos asociados.

En la figura 9 se puede observar la matriz de confusión que obtenemos a partir de este modelo utilizando los datos de prueba.

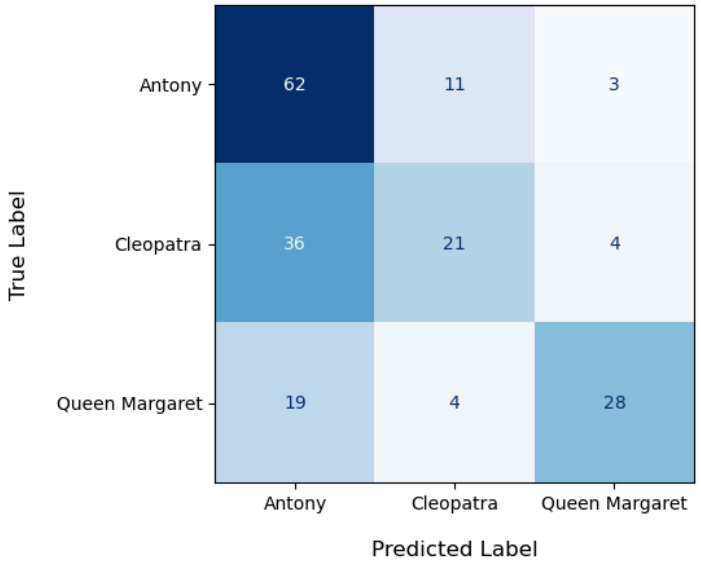

Fig 9. Matriz de confusión para datos de prueba

Los valores de precisión (accuracy) de este modelo fue de 0.59. Los valores de precision y recall de cada personaje fueron los siguientes:

Personaje	Precisión	Recall
Antony	0.5299	0.8158
Cleopatra	0.5833	0.3443
Queen Margaret	0.8000	0.5490

Esto significa que de todos los textos que el modelo atribuyó al personaje Antony el 52,99% efectivamente pertenecían al personaje. Por otra parte, de los textos que realmente son de Antony el modelo identificó correctamente el 81,5%. De los textos que el modelo predijo como del personaje Cleopatra el 58% pertenecen al mismo. Del total de los textos de Cleopatra el modelo pudo identificar correctamente el 34%. Y finalmente, de los textos que el modelo predijo como del personaje Queen Margaret el 80% pertenecen al mismo, mientras que del total de textos de este personaje el modelo pudo identificar correctamente el 55%.

En este caso, parecería que el modelo sobreestima la asignación de textos al personaje Antony. Esto puede deberse a la mayor cantidad de textos de este personaje en nuestro conjunto de entrenamiento. Este desbalance puede generar un valor de precisión engañosamente alto porque el modelo estima por demás la clase más frecuente del cuerpo de datos. Lo podemos confirmar con el alto valor de recall para el personaje Antony.

Búsqueda de hiper-parámetros con validación cruzada

La validación cruzada (cross-validation) es una técnica de evaluación del rendimiento de un modelo de aprendizaje automático, de forma de obtener robustez y menor sesgo en el entrenamiento. La técnica funciona de la siguiente forma:

- Dividir al conjunto de entrenamiento en un cantidad n de subconjuntos del mismo tamaño, también llamados pliegues (fold).
- Para cada uno de los n subconjuntos, se entrena el modelo con el resto de los n-1 subconjuntos y se lo evalúa con el subconjunto de la iteración.
- Luego de haber utilizado como conjunto de entrenamiento a todos los subconjuntos n, se promedian las métricas de rendimiento obtenidas, para obtener el rendimiento del modelo.

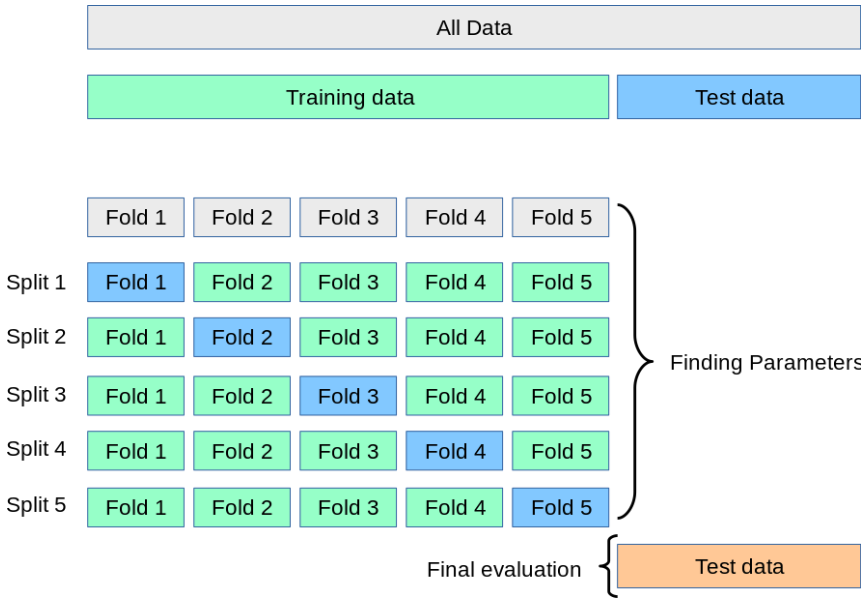

Fig 10. Ilustración del procedimiento de la guía de usuario scikit-learn para 5 pliegues[1]

Luego de entrenar el modelo utilizando validación cruzada, 3 pliegues, para distintas combinaciones de hiper-parámetros, se representa el correspondiente rendimiento en forma de gráfica de violín. Como se puede ver, cuando se eliminan las stopwords se obtiene mejor precisión, y cuando es unigrama obtiene mejores resultados que el unigrama y bigrama. Finalmente, el modelo con mejor rendimiento es aquel que quita los stopwords, unigrama y el IDF habilitado.

[1] https://scikit-learn.org/stable/modules/cross_validation.html

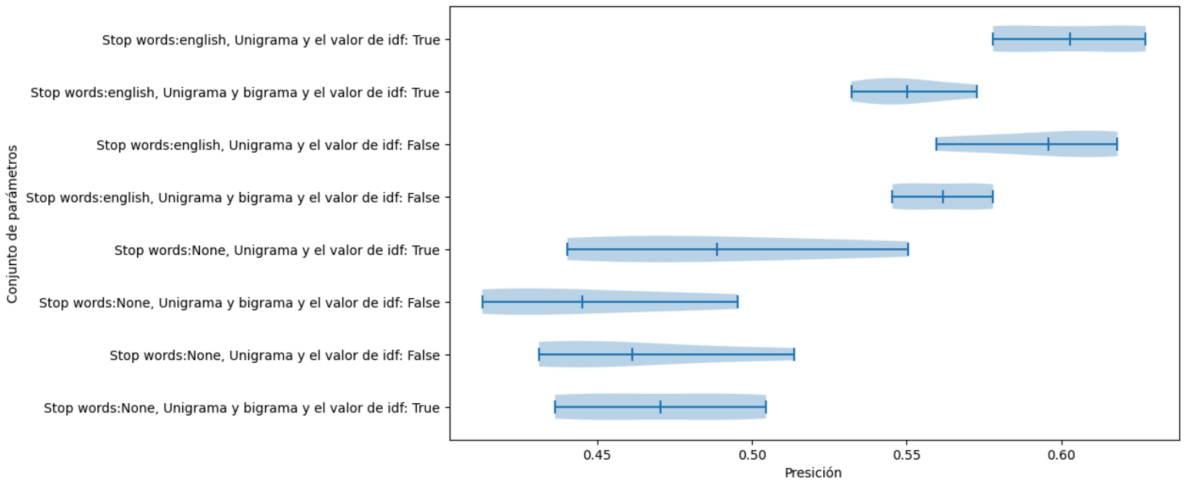

Fig 11. Gráfico comparativo de combinaciones de hiper-parámetros utilizando validación cruzada.

Al entrenar el modelo con mejor rendimiento, se obtiene una precisión de 0.57 y la matriz de confusión se presenta en la figura 12. Sin embargo, al probar el modelo sin stop word, con n-grama (1,1) pero el idf deshabilitado, la precisión es muy similar (0.564).

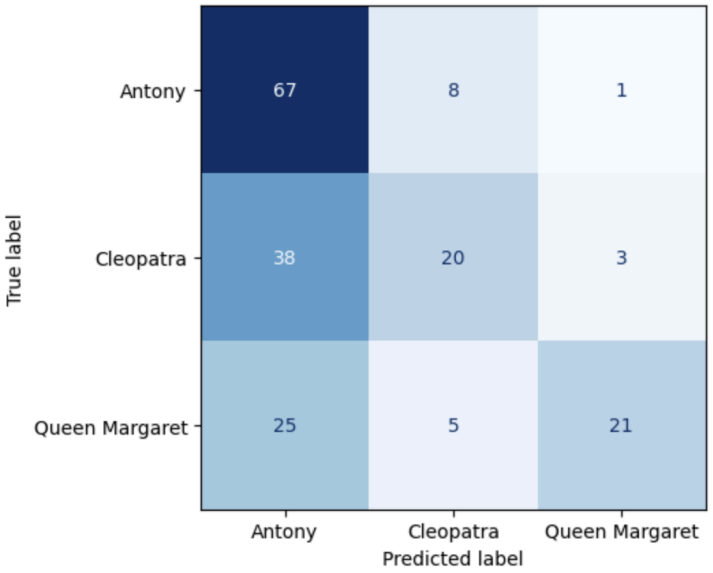

Fig 12. Matriz de confusión para el modelo sin stop words, con n-grama (1,1) y el idf habilitado.

Algunas de las limitaciones que pueden tener las técnicas bag of words o TF-IDF en el procesamiento de texto tienen que ver con el nivel de importancia que se les da a las palabras en un documento. Si bien, TF-IDF le da distinta relevancia a las palabras según la cantidad de ocurrencias, reduciendo el peso de la frecuencia de un término por un factor que aumenta con la cantidad de apariciones (Manning, C. D., 2008), en ninguno de los dos casos se le da importancia a la posición donde se encuentran las palabras (Jurafsky, D., & Martin, J. H., 2019). Este posicionamiento puede ser útil para comprender a las palabras en su contexto, incluso para darle significado. Además en ambos casos, la representación del texto puede llevar a

vectores muy grandes y dispersos, ya que un documento puede no utilizar todas las palabras, lo que puede causar problemas de almacenamiento y procesamiento.

Análisis utilizando Support Vector Machines

Los Support Vector Machines (SVM) son algoritmos de aprendizaje supervisado que se utilizan tanto para la clasificación como para la regresión. Sin embargo, son más conocidos y utilizados para el primer tipo de problemas. La idea básica detrás de los SVM es encontrar un hiperplano, en un espacio con muchas características, que separe las diferentes clases de datos.

Utilizando los mismos parámetros que en la sección anterior (unigrama, IDF=true) obtenemos un accuracy de 0.67, lo cual es discretamente mejor que el modelo Bayesiano. Podemos observar en la figura 13 la matriz de confusión pareciera indicar que este modelo es mejor determinando los dos personajes con menor texto (Cleopatra y Queen Margaret), manteniendo un buen desempeño prediciendo los textos de Antony.

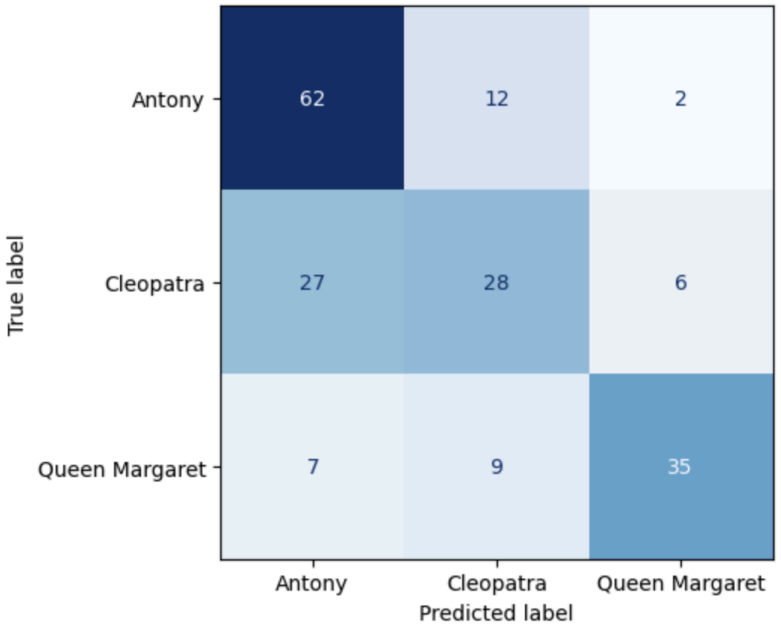

Fig 13. Matriz de confusión para el modelo de SVM

Otras aproximaciones

Se utilizaron dos modelos distintos, Random Forest y Gradient Boosting, pero ambos dan menor precisión. El primero hace referencia a una serie de árboles de decisión sobre muestras de entrenamiento generadas por bootstrap. Mientras que el segundo optimiza la función objetivo del modelo al agregar árboles de manera secuencial, ajustando cada nuevo árbol para corregir los errores de los árboles anteriores.

Cambio de personajes

Como observamos hasta el momento, la cantidad de datos que tenemos de cada personaje influye en gran medida en el rendimiento que tienen los modelos para predecirlos. Por esta

razón, exploramos los 3 personajes con mayor cantidad de párrafos en todo el conjunto de datos.

De la misma forma que hicimos inicialmente, nos aseguramos que la proporción de datos de entrenamiento y de test sean similares (figura 14).

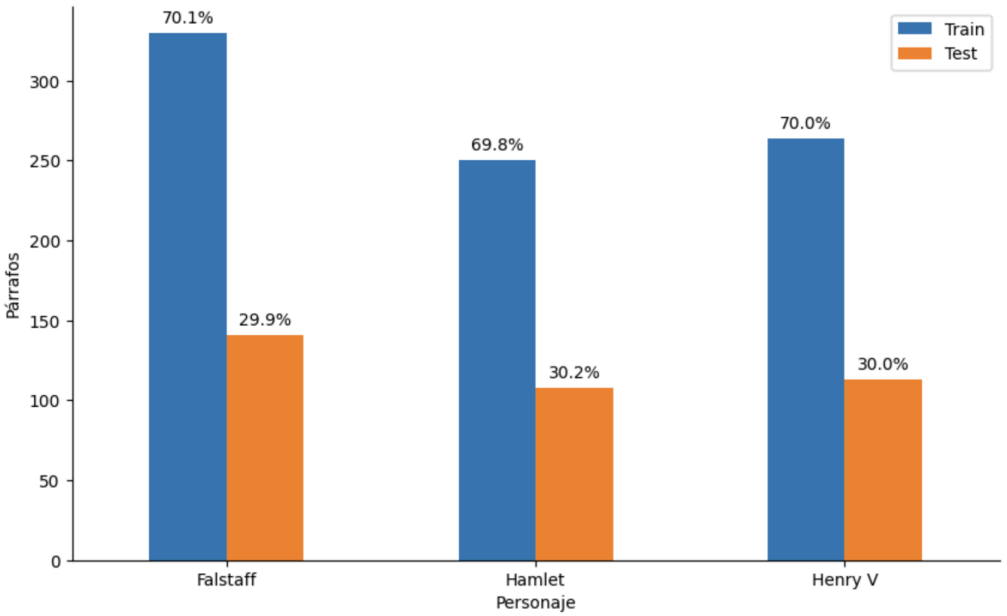

Fig 14. Cantidad de párrafos de entrenamiento y prueba por personaje

De la misma forma que pasó con Antony, Cleopatra y Queen Margaret, al realizar una visualización de los 2 componentes principales utilizando PCA podemos notar una gran superposición de datos (figura 15).

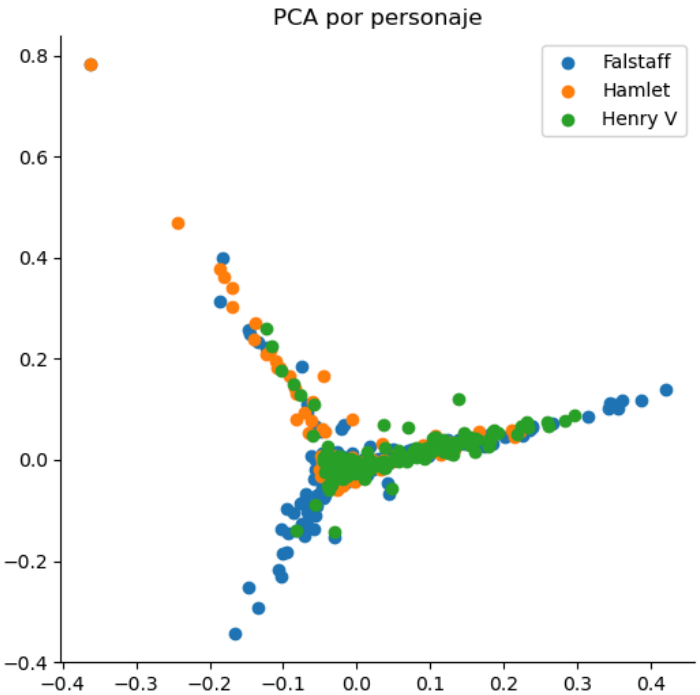

Fig 15. Representación de primeras 2 componentes principales sin stop words y n-grama=2

Utilizando este mismo conjunto de datos, se entrena un modelo multinomial de Bayes, sin embargo, los resultados no son demasiado alentadores (figura 16). Se obtiene nuevamente una sobreestimación del personaje con más párrafos. Esto sugiere que el desbalance de datos influye significativamente en el rendimiento de estos modelos.

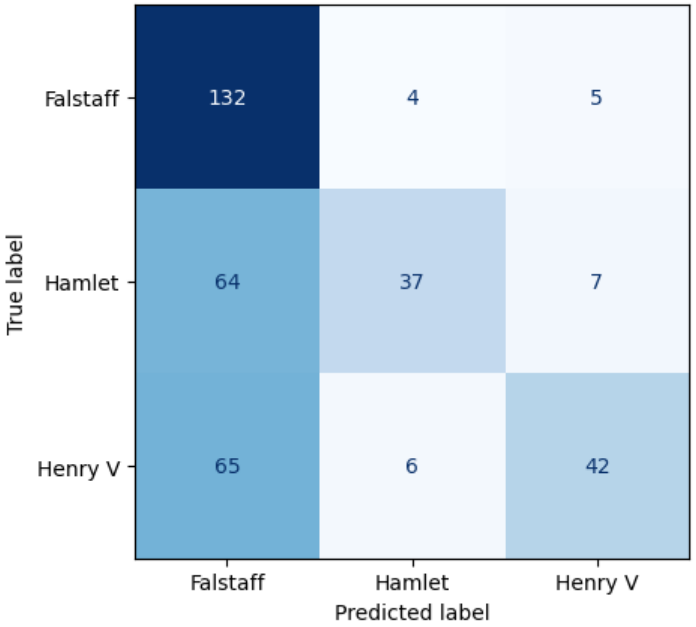

Fig 16. Matriz de confusión para el modelo multinomial de Bayes para personajes con más párrafos

Para sortear este tipo de desequilibrios se podrían utilizar técnicas de sobremuestreo o submuestreo, que consisten en equiparar la cantidad de muestras de las diferentes categorías. En el caso del sobremuestreo aumentando los textos de los personajes con menos representación y en el caso del submuestreo reduciendo la cantidad de textos del personaje con más representación.

Ambas técnicas tienen sus pros y sus contras, en el caso del sobremuestreo duplicar la información puede favorecer el sobreajuste; mientras que en el caso del submuestreo perdemos información lo cual puede resultar en introducción involuntaria de sesgos.

Técnicas alternativas

Una técnica diferente para extraer características de texto es Word2Vec. A diferencia de métodos más tradicionales como Bag-of-Words y TF-IDF, Word2Vec utiliza el aprendizaje profundo para representar palabras en un espacio vectorial continuo. Esto significa que, además de capturar la presencia de palabras, también comprende su contexto semántico.

Word2Vec funciona con dos modelos principales: Continuous Bag of Words (CBOW) que predice la palabra central basándose en las palabras circundantes, y Skip-Gram para predecir las palabras del contexto dadas una palabra central.

Ambos modelos entrenan una red neuronal para aprender representaciones vectoriales de palabras (denominadas embeddings en inglés). Estos vectores se diseñan de manera que las palabras con significados similares queden cercanas entre sí en el espacio vectorial.

El uso de Word2Vec puede mejorar considerablemente el rendimiento en tareas que requieren entender el contexto y las relaciones semánticas, como el análisis de sentimientos y la detección de sinónimos. Esto es especialmente útil para asignar personajes a párrafos, ya que ayuda a comprender mejor el contexto de la obra o escena.

Fast Text

FastText es una librería de aprendizaje automático que se utiliza para tareas de procesamiento de textos, debido a ser rápida y eficiente para entrenar modelos y generar embeddings precisos. Se entrenó un modelo de aprendizaje supervisado utilizando los datos de entrenamiento y la precisión fue intermedia entre el Modelo Bayesiano y el SVM (0.61). Además, la matriz de confusión resultante se presenta en la figura 17, y la tabla a continuación presenta la precisión y el recall para cada personaje:

Personaje	Precisión	Recall
Antony	0.65	0.65
Cleopatra	0.65	0.55
Queen Margaret	0.53	0.61

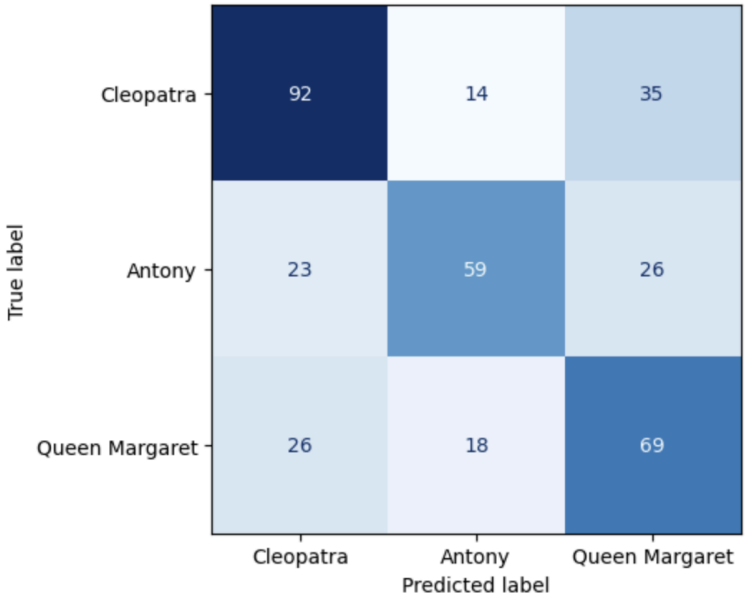

Fig 17. Matriz de confusión para el modelo Fast Text

Realizando un análisis de lo observado, podemos decir que si bien las tres medidas son balanceadas, aún hay margen de mejora en el modelo para poder predecir mayor cantidad de párrafos correctos.

Conclusiones

En el presente trabajo, hemos explorado la base de datos de Shakespeare, centrándonos en la tabla "paragraphs" y realizando un análisis detallado para los personajes Antony, Cleopatra y Queen Margaret. Mediante la aplicación de diversas técnicas de preprocesamiento y transformación de texto, como Bag-of-Words, n-gramas y TF-IDF, y evaluamos varios modelos de aprendizaje automático, incluidos Multinomial Naive Bayes y Support Vector Machines (SVM). A través de esta exploración, hemos observado un desbalance en la cantidad de datos de cada personaje que afecta significativamente el rendimiento de los modelos, resultando en una sobreestimación de los textos para los personajes con más datos disponibles.

Referencias

Abdi, H., & Williams, L. J. (2010). Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4), 433-459.

Jurafsky, D., & Martin, J. H. (2019). Speech and Language Processing. Prentice Hall.

Manning, C. D. (2008). Introduction to information retrieval. Syngress Publishing,.