Universidad de la República
Facultad de Ingeniería

Introducción a la Ciencia de Datos

Análisis de Texto de la Obra de Shakespeare

Quién Lo Dice?

Nombre	**CI**
Mauro Martínez	4.755.540-2
Victoria Luz	5.324.661-1

7 de julio de 2023

Objetivo

Esta tarea tiene por objetivo poner en práctica algunos conceptos clave de aprendizaje automático, aplicado a técnicas de procesamiento de lenguaje natural.

Para ello, en primer lugar, aplicaremos distintas técnicas para la representación numérica de texto (features) como: Bag of Words (BoW) y Term Frequency - Inverse Document Frequency (TF-IDF). Una vez hecho esto, aplicaremos la técnica de reducción de dimensiones PCA, para visualizar la representación obtenida y evaluar una posible separación de clases.

Luego, entrenaremos el modelo de clasificación Multinomial Naive Bayes y evaluaremos su desempeño para distintos hiper-parámetros. En ese sentido, utilizaremos la técnica de validación cruzada para tales fines. Finalmente, evaluaremos y compararemos otros modelos de clasificación de texto, utilizando las mismas features.

En una segunda etapa, cambiaremos de personajes y evaluaremos el mismo problema anteriormente detallado.

Por último, investigaremos sobre técnicas alternativas para la extracción de features de texto. En particular entrenaremos el modelo de FastText y evaluaremos su desempeño en comparación con los modelos anteriormente utilizados.

Descripción de la Base de Datos

Se utilizará una base de datos relacional abierta con la obra completa de William Shakespeare [1]. La estructura de dicha base de datos se puede ver en la Figura 1.

Tabla "works"

Esta tabla contiene todos los trabajos realizados por Shakespeare. Cada instancia, es decir, cada trabajo, tiene un identificador único, el título completo, un título corto, el año en que fue publicado y el género al que pertenece.

Tabla "chapters"

Contiene todos los capítulos de todas las obras de Shakespeare. Se vincula con la tabla "works" a través de "work_id". Es una relación $N : 1$, donde cada trabajo puede tener N capítulos, pero cada capítulo tiene 1 y solo 1 trabajo asociado. La tabla posee además un identificador único para cada capítulo, el acto al cual pertenece, la escena y una breve descripción.

La cantidad de capítulos por trabajo es variada, pudiendo encontrar un trabajo de un solo capítulo y otro de 154 capítulos.

[1] https://relational.fit.cvut.cz/dataset/Shakespeare

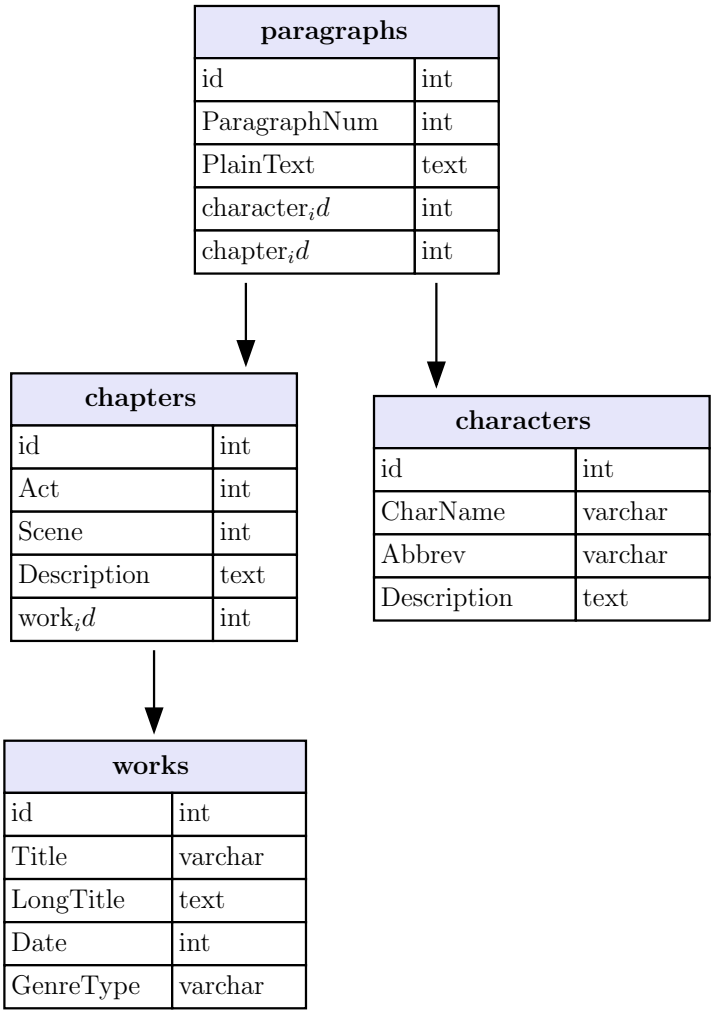

Figura 1: Estructura de la Base de Datos Relacional

Tabla "paragraphs"

Esta tabla contiene el texto plano ("PlainText") correspondiente al diálogo de cada personaje, el cual está identificado por el atributo "character_id", siendo esta la forma en que se vincula con la tabla "characters". Es una relación $N : 1$, donde cada párrafo tiene asociado un único personaje, pero para cada personaje puede haber más de un párrafo.

Por otro lado, cada párrafo tiene asociado un único capítulo al cual pertenece y se identifica con el atributo "chapter_id". Esta es la forma en que se vincula con la tabla "chapters". Es una relación $N : 1$ puesto que un capítulo puede contener mas de un párrafo.

El atributo "ParagraphNum" indica el número de párrafo en el correspondiente capítulo. Finalmente, cada párrafo tiene un identificador único "id".

Tabla “characters”

La tabla “characters”contiene todos los personajes de todas las obras de Shakespeare. Cada personaje tiene un único identificador “id”, una abreviatura y una descripción.

Acá ya podemos ver un problema de calidad de datos ya que existen muchos personajes sin una descripción, es decir, tenemos datos faltantes.

Exploración de los Datos

Como primer análisis básico de la base de datos en cuestión, comenzaremos por analizar la cantidad de obras escritas por Shakespeare por año, lo cual se puede apreciar en la Figura 2.

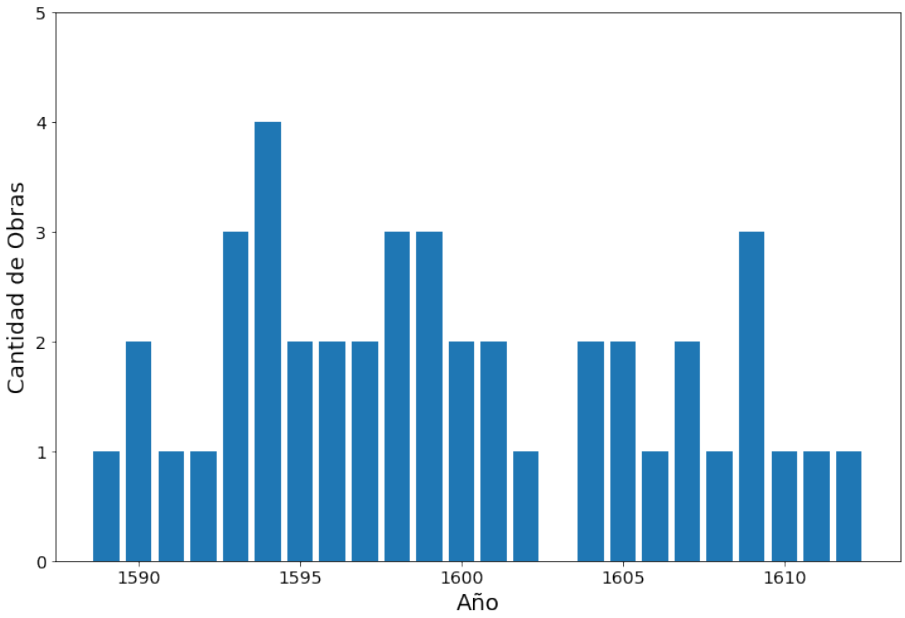

Figura 2: *Cantidad de obras de Shakespeare por año*

Podemos observar que tanto al comienzo como al final, el rítmo de producción es menor comparado con su producción entre los años 1593 y 1609. Entre estos años, logra un pico alto de 4 obras en 1594, entre las que se encuentra “Romeo and Juliet”, por lo que podemos decir que corresponde a un punto álgido del autor. Luego se mantiene en un promedio de 2 obras por año, aproximadamente.

También podemos notar que en 1603 el autor no publica ninguna obra, lo cual puede deberse a problemas de salud, problemas legales, falta de inspiración, obras que fueron perdidas, entre otros. Deberíamos contrastar con su biografía.

En linea con el último comentario realizado, en la Figura 3 notamos que la producción en los años siguientes a 1603, donde no publica ninguna obra, se corresponde en

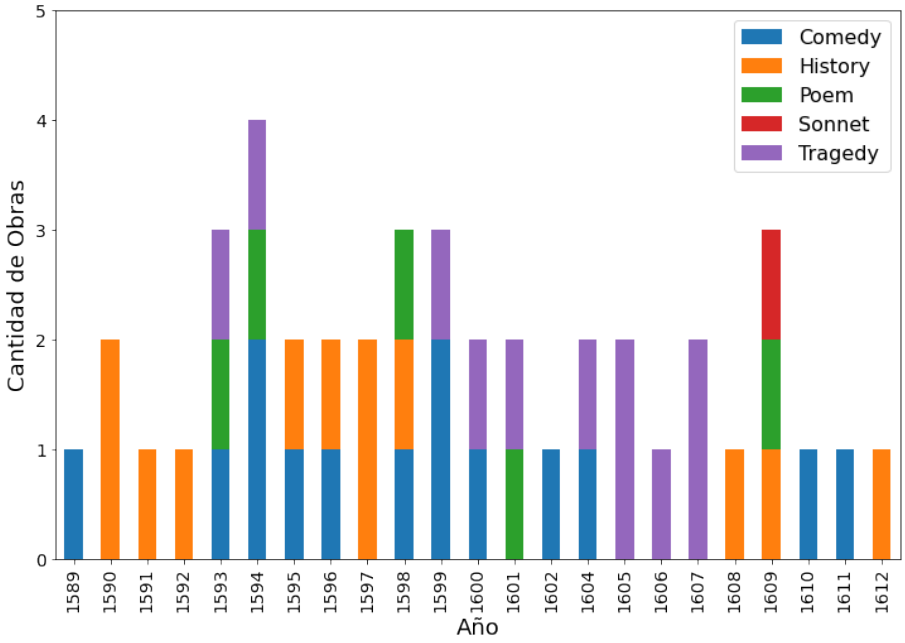

Figura 3: *Cantidad de obras de Shakespeare por año y género. Notar que en el año 1603 no publica ninguna obra.*

su mayoría al género "Tragedy". Por lo cuál, podríamos llegar a pensar que el autor se encontraba en un momento trágico de su vida debido a un suceso acontecido en el año 1603.

Podemos observar también que el género "Comedy" ha estado presente desde el comienzo, exceptuando ese período, lo cual refuerza la teoría que he planteado. De nuevo, esto es una suposición que deberíamos corroborar con su biografía, si es que está mencionado.

Calidad de los Datos

Datos Faltantes

Al hacer una exploración de posibles datos faltantes, notamos que la única tabla que presenta este problema de calidad, corresponde a la tabla "df_characters". En particular, hemos observado que en su gran mayoría, corresponden a la descripción del personaje.

Un análisis en detalle, revela que hay personajes genéricos que son comunes a varias obras, como por ejemplo: "Messanger", "Servant", "Lord". Lo cual se corresponde con la época. Sin embargo desde el punto de vista de consistencia de los datos, si efectivamente se trata del mismo personaje, tenemos varios identificadores para lo que sería el mismo personaje. Incluso, analizando el caso particular de "Messenger", también notamos que tiene varias abreviaciones posibles.

Pero también notamos que hay referencias a las primeras entradas de los personajes, como son “First Lord”, “Second Lord”, “First Gentleman”, “Second Gentleman” o “First Citizen”. También encontramos personajes que no lo son propiamente sino que refrieren a un grupo de personajes, como “All” o “both”. Posiblemente se correspondan a diálogos en simultaneo.

Cantidad de Párrafos por Personaje

Al analizar la cantidad de párrafos por personaje, notamos que el personaje con más párrafos resulta ser “(stage directions)”, es decir, las direcciones de escena. Claramente, en los libretos estan incluídas las direcciones de escena y le fueron asignadas como personaje “(stage directions)”. Aquí encontramos una inconsistencia en los datos. Este es un punto que debemos discutir, si pretendemos analizar los personajes a través de las palabras.

Luego, el segundo personaje con más párrafos es “Poet”. Tras analizar cuáles son los trabajos que tienen a este personaje, nos encontramos con otra inconsistencia. Resulta que los Poemas y el Soneto tienen por personaje asignado “Poet”. Por otro lado, hay dos tragedias escritas que lo tienen por personaje. En este caso, consideramos que hace referencia genérica a un poeta.

Finalmente, volvemos a notar personajes que no lo son propiamente y refieren a las primeras apariciones o a personajes genéricos como “Third Apparition”, “Thieves”, “First Apparition”o “First Messenger”.

Limpieza de Datos

Primera Etapa de Limpieza: Personajes

Visto y considerando lo anterior, decidimos quitar las direcciones de escena como personaje y todas sus palabras asociadas. De la misma forma apartamos todos los poemas y a sonetos.

Además de esto, hemos notado que existen direcciones de escena incluidas en algunos párrafos entre paréntesis rectos, bajo la estructura: “[stage directions]”. En su mayoría, contienen frecuentemente las palabras: “enter”o “exeunt”. Estas direcciones de escena puede alterar dicho conteo de palabras. Por este motivo, decidimos quitarlas.

Segunda Etapa de Limpieza: Normalización

Para el análisis de texto y conteo de palabras, es necesario normalizar el texto. Esto implica pasar todo el texto a minúsculas, eliminar los signos de puntuación o sustituir las contracciones por las correspondientes frases completas.

Hecho esto, hemos sustituido cada uno de los párrafos normalizados por una lista de palabras. Luego, explotamos la tabla en cada una de las palabras.

Tercera Etapa de Limpieza: Stop Words

Tras realizar esta normalización y primer limpieza, nos dispusimos a buscar las palabras más frecuentes a lo largo de todas las obras. De esta forma podríamos quizás inferir alguna temática común o cuáles eran las principales preocupaciones del autor. Sin embargo, nos topamos con que las palabras más frecuentes resultan ser las denominadas "Stop Words".

Conteo de Palabras

Tras realizar la limpieza de los datos de acuerdo a lo anteriormente detallado, podemos calcular la frecuencia de las palabras a lo largo de toda la obra, lo cual se muestra en la Figura 4. Observamos que la palabra con sentido propio que más se repite es "lord", seguida de "sir", las cuales se corresponden con las autoridades típicas de la época. Hemos leído que Shakespeare denotaba rebeldía en sus obras contra la autoridad[2]. Por otro lado, la palabra que sigue con más frecuencia es "love", lo que puede significar un tema recurrente en las obras de Shakespeare.

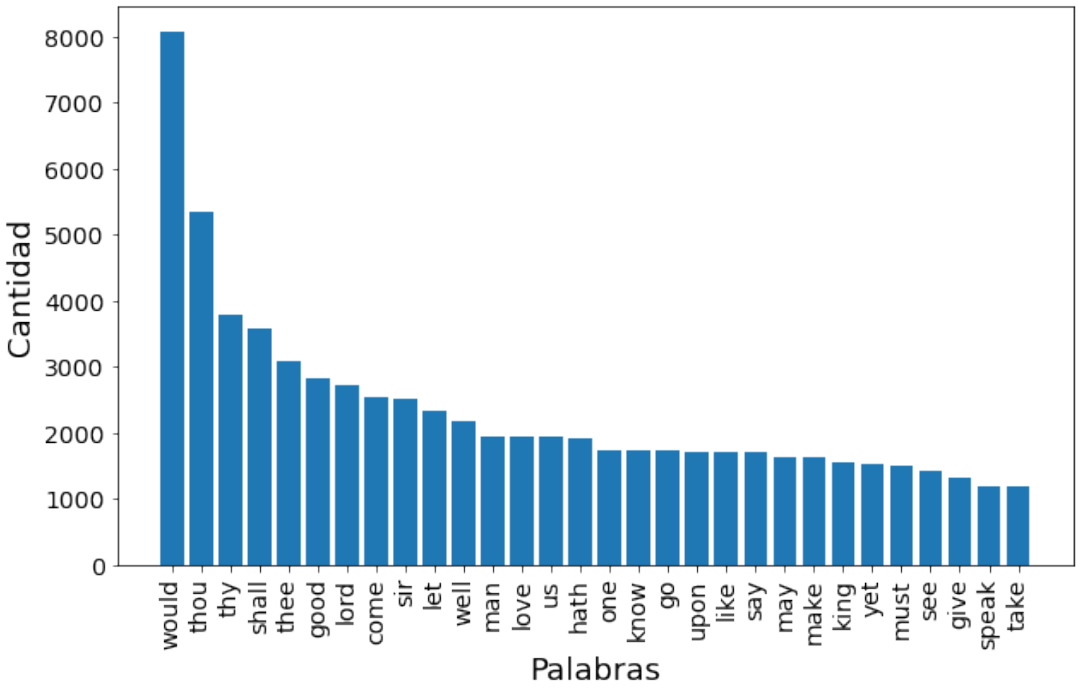

Figura 4: *Las 20 palabras más frecuentes sin considerar las direcciones de escena, ni los poemas o el soneto y luego de aplicar el filtro con la librería NLTK de Python.*

También podemos calcular la cantidad de palabras por personaje, lo cual se muestra en la Figura 5. En este sentido, vemos que los personajes con más palabras y por lo tanto, posiblemente los de mayor importancia, son: "Henry V", seguido de "Falstaff"y "Hamlet".

[2]https://humanidades.com/william-shakespeare/

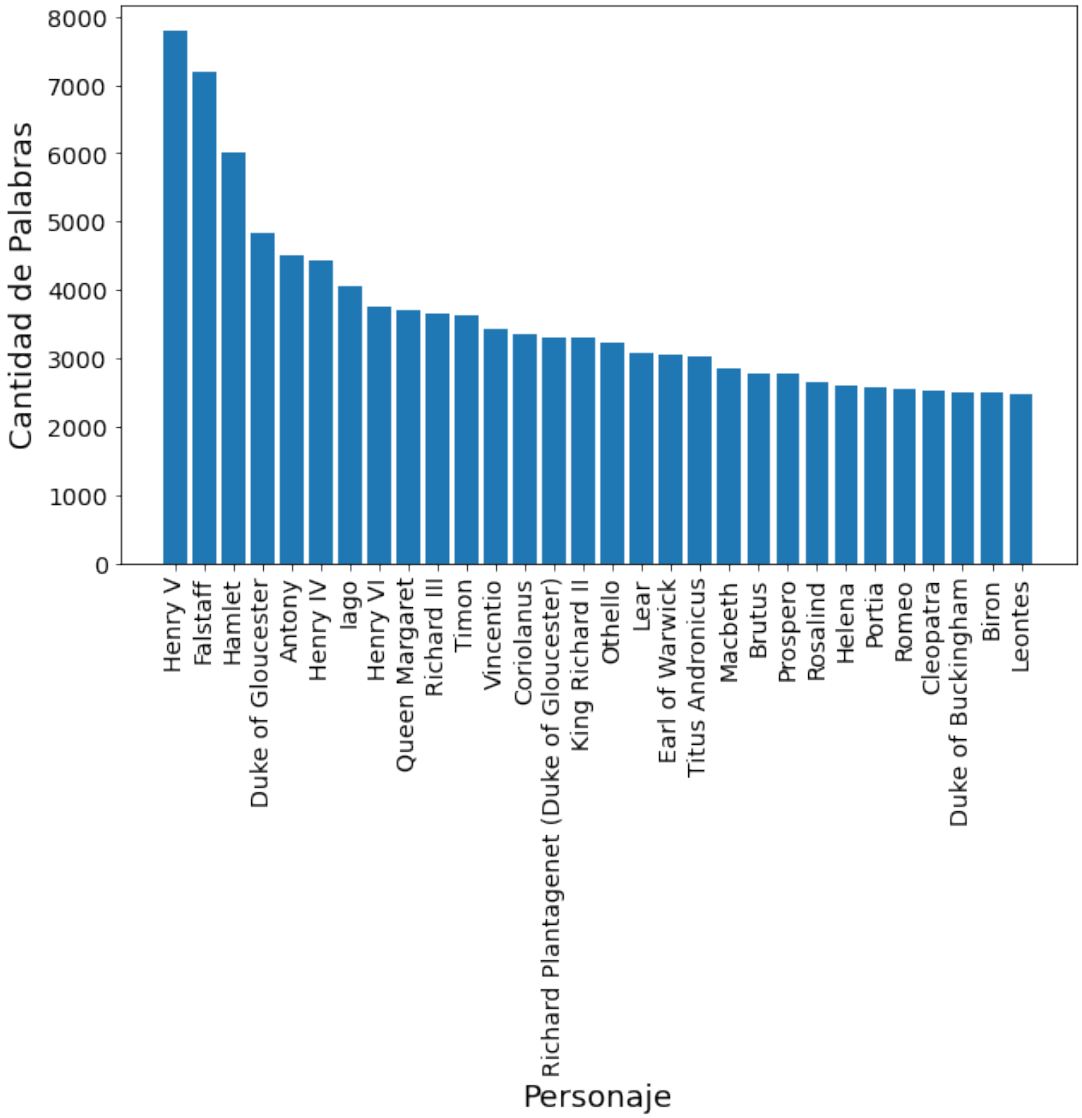

Figura 5: *Los 20 personajes con más palabras, sin considerar las direcciones de escena, ni los poemas o el soneto y luego de aplicar el filtro de la librería NLTK de Python.*

Dataset

Se seleccionaron los párrafos correspondientes a tres personajes específicos: Antony, Cleopatra y Queen Margaret, los cuales va a ser utilizados para el análisis a continuación.

Luego, se llevó a cabo una partición de los datos utilizando un enfoque de muestreo estratificado. En primer lugar, se decide separar un conjunto de prueba o testeo que corresponde al 30 % del total de datos disponibles. Este enfoque asegura que el conjunto de prueba sea representativo de la distribución de los datos completos.

Para lograr una partición estratificada, se realiza un cálculo del conteo de personajes presentes en los conjuntos de entrenamiento y prueba. Esto implica determinar la cantidad de apariciones de cada personaje en ambos conjuntos. Luego, se calcula la proporción de

párrafos en términos porcentuales para cada personaje en cada conjunto. Esta información permite comprender la distribución relativa de los párrafos asociados a cada personaje en cada conjunto de datos.

Finalmente, con el propósito de visualizar de manera clara y concisa estas proporciones, se crea un gráfico de barras el cual se puede apreciar en la Figura 16.

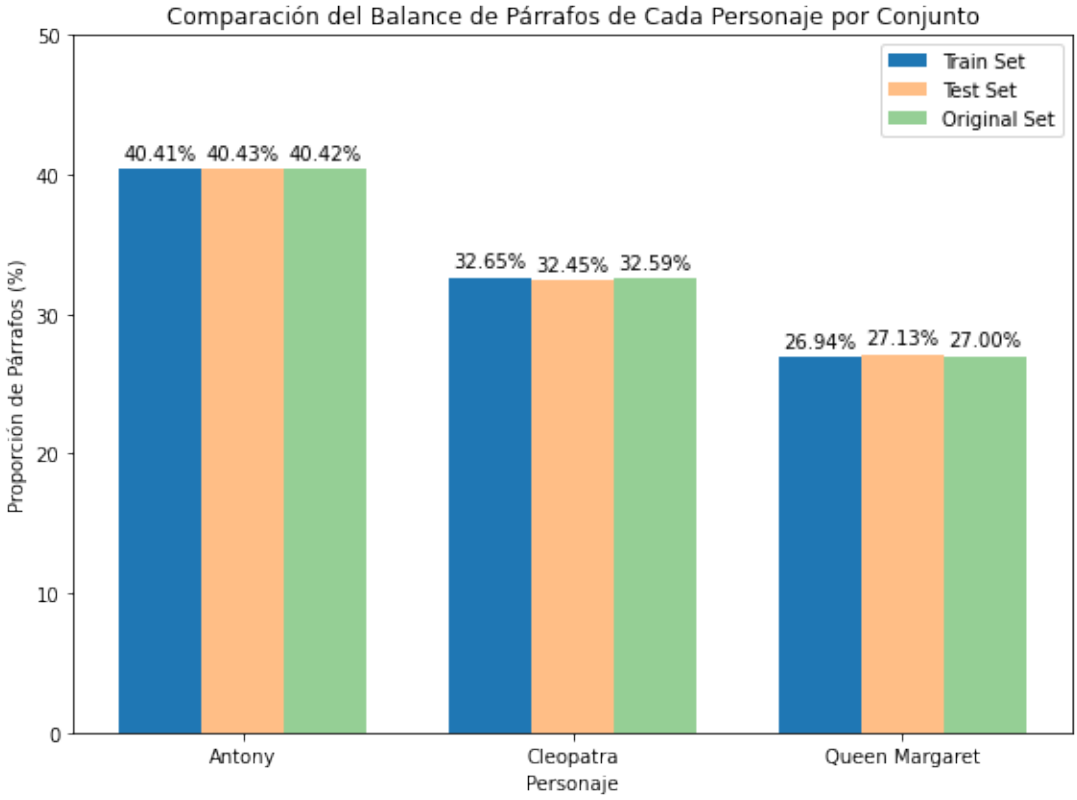

Figura 6: *Balance entre Cantidad de Párrafos por Personaje por Conjunto.*

Representación Numérica de Palabras

Bag of Words

Después de preparar los datos, se procedió a transformar el texto del conjunto de entrenamiento en una representación numérica conocida como "Bag of Words" o conteo de palabras. Este enfoque trata cada documento de texto como una "bolsa" de palabras, ignorando el orden y la estructura gramatical y cuenta la frecuencia de aparición de cada palabra en el documento.

El proceso para convertir el texto en una representación de "Bag of Words"implica los siguientes pasos:

1. Creación del vocabulario: Se construye un vocabulario único a partir de todas las palabras presentes en el conjunto de entrenamiento.

2. Vectorización: Cada documento de texto se convierte en un vector numérico, donde cada elemento del vector representa la frecuencia de una palabra en el documento. Por lo tanto, el tamaño del vector será igual al tamaño del vocabulario.

Veamos un ejemplo para ilustrar cómo funciona esta técnica. Supongamos que tenemos los siguientes tres documentos de entrenamiento:

- Documento 1: "El gato es negro"
- Documento 2: "El perro es marrón"
- Documento 3: "El gato y el perro son amigos"

Los pasos para convertir el texto a una representación BoF, son:

1. Construcción del vocabulario: El vocabulario único en este caso sería: ["El", "gato", "es", "negro", "perro", "marrón", "y", "son", "amigos"]
2. Vectorización: Representaremos cada documento en forma de vector contando la frecuencia de las palabras en el vocabulario.
    - Documento 1: [1, 1, 1, 1, 0, 0, 0, 0, 0]
    - Documento 2: [1, 0, 1, 0, 1, 1, 0, 0, 0]
    - Documento 3: [2, 1, 1, 0, 1, 0, 1, 1, 1]

El tamaño de la matriz resultante depende del número de documentos en el conjunto de entrenamiento y del tamaño del vocabulario. Si hay N documentos y M palabras únicas en el vocabulario, la matriz resultante tendrá una dimensión de N x M. Esta matriz resulta ser una matriz dispersa (sparse matrix). Esto se debe a que en la mayoría de los casos, cada párrafo contiene sólo una pequeña fracción del vocabulario total. Por lo tanto, la mayoría de las entradas de la matriz serán cero. Esta propiedad de esparsidad puede ser aprovechada para ahorrar memoria y cálculos computacionales, ya que no es necesario almacenar ni operar con todos los ceros en la matriz. Hay estructuras de datos y algoritmos especializados para trabajar con matrices esparsas, lo que es especialmente útil para cuando se trabaja con grandes conjuntos de texto.

n-grama

Una mejora posible al enfoque BoW es utilizar n-gramas en lugar de palabras individuales. Un n-grama es una secuencia contigua de n elementos de un texto, donde los elementos pueden ser caracteres, palabras o cualquier unidad definida.

Al analizar los n-gramas en un texto, se captura una mayor cantidad de información contextual y se tiene en cuenta el orden de las palabras en cierta medida. Esto permite obtener información sobre la frecuencia de aparición de ciertas combinaciones de palabras y ayuda a comprender mejor el significado y la estructura del texto. Utilizar n-gramas en lugar de palabras individuales puede ser especialmente útil en casos donde el orden de las palabras es relevante, como en el análisis de textos con estructuras específicas o en la detección de frases coloquiales. Esta técnica ampliada de representación del texto, conocida como "bag-of-ngrams", permite una comprensión más precisa y enriquecida de los textos al considerar las combinaciones de palabras en grupos de n elementos.

TF-IDF

La representación numérica Term Frequency-Inverse Document Frequency (TF-IDF) se utiliza para mejorar el enfoque de Bag of Words (BOW) al considerar la importancia relativa de las palabras en todos los documentos. La idea principal detrás de TF-IDF es que una palabra es más importante para un documento si aparece con frecuencia en ese documento pero es poco común en el resto de los documentos.

El cálculo de TF-IDF se realiza en dos pasos:

1. Term Frequency: Calcula la frecuencia de aparición de una palabra en un documento en relación con el total de palabras en el contexto del conjunto de documento. Esto refleja la importancia relativa de una palabra dentro de un documento específico.

2. Inverse Document Frequency: Calcula la importancia de una palabra en el conjunto de documentos. Se calcula como el logaritmo del cociente entre el número total de documentos y el número de documentos que contienen la palabra. Esto ayuda a penalizar las palabras que aparecen en muchos documentos y por lo tanto, tienen menos información distintiva.

La transformación final TF-IDF se obtiene multiplicando la Term Frequency (TF) por la Inverse Document Frequency (IDF) para cada palabra en cada documento.

Reducción de Dimensión: PCA

Para la reducción de dimensionalidad, se aplicó el análisis de componentes principales (PCA) a la representación numérica obtenida de los párrafos, utilizando la técnica de TF-IDF, filtrando las stop words y considerando un-igramas y bi-gramas. Consideramos, en primer lugar, las dos primeras componentes principales.

La Figura 7 muestra un gráfico de dispersión con la representación obtenida, luego de aplicar PCA. Cuando los puntos en un gráfico de reducción de dimensiones utilizando PCA están muy agrupados, indica que los datos presentan una alta similitud o coherencia entre ellos. En este contexto, la agrupación de los puntos refleja que los personajes seleccionados comparten características y no se pueden distinguir claramente unos de otros.

Como forma de ver si mejora la separación entre los personajes, se incorporaron un mayor número de componentes principales. En primera instancia, se prueba utilizando tres componentes con el mismo set de parámetros, obteniendo el gráfico que se muestra en la Figura 8.

Al incluir una tercera componente en el análisis y representarla en un gráfico en 3D, se añade varianza en esa dimensión adicional. Esto también implica que se incorpora más información y "profundidad.ª]l gráfico. Además, se observa que la clase dominante, Antony, comienza a separarse del resto de las clases en el espacio tridimensional.

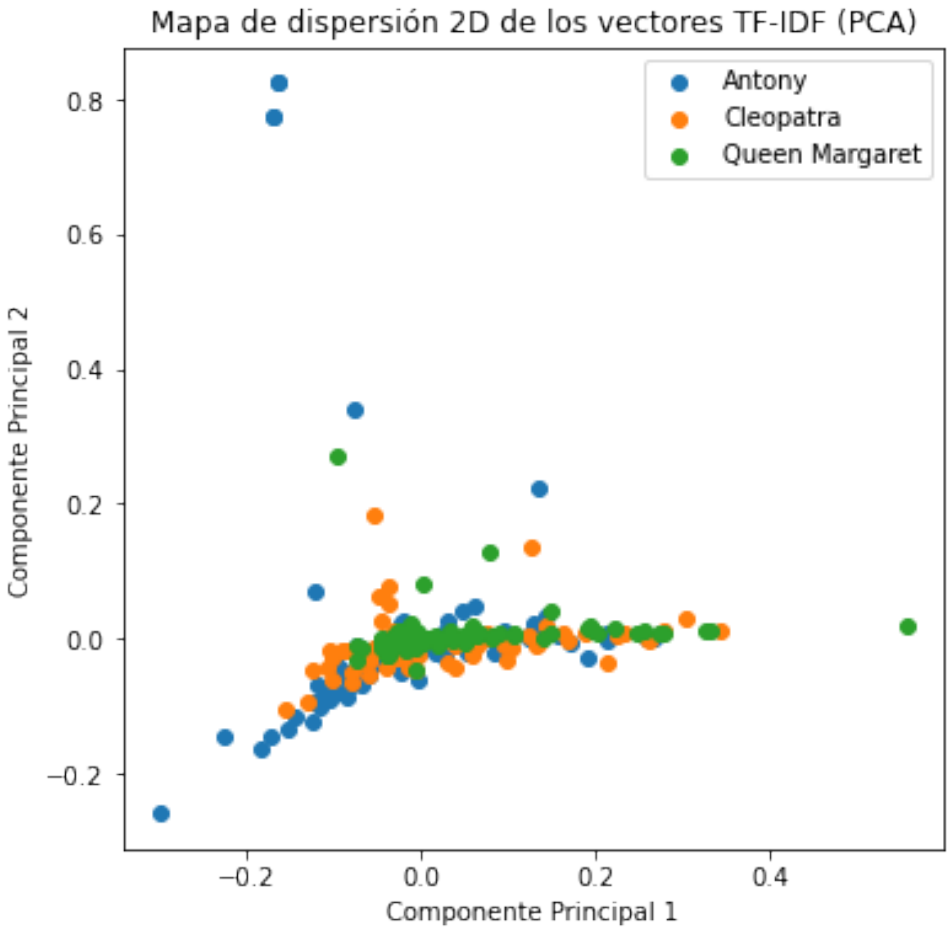

Figura 7: *Mapa de Dispersión de los vectores TF-IDF utilizando 2 componentes principales (PCA).*

Por último, en la Figura 9 se puede apreciar cómo varía la varianza explicada en porcentaje, al agregar componentes principales. A medida que se incorporan más componentes a la matriz, se observa un aumento progresivo en la varianza explicada.

Modelos de Clasificación

Multinomial Naive Bayes

Se entrenó el modelo Multinomial Naive Bayes sobre la transformación TF-IDF del conjunto de entrenamiento definido anteriormente, filtrando las stop words y considerando uni-gramas y bi-gramas. Se calcularon los valores de precision, recall y accuracy, los cuales se indican en la Tabla 1.

Mirar únicamente el valor de accuracy puede ser problemático en situaciones de desbalance de datos. En tales casos, el modelo puede tener un alto accuracy simplemente prediciendo la clase mayoritaria en todo momento, sin prestar atención a las clases minoritarias. Esto puede ocultar problemas de rendimiento en estas clases y dar una visión

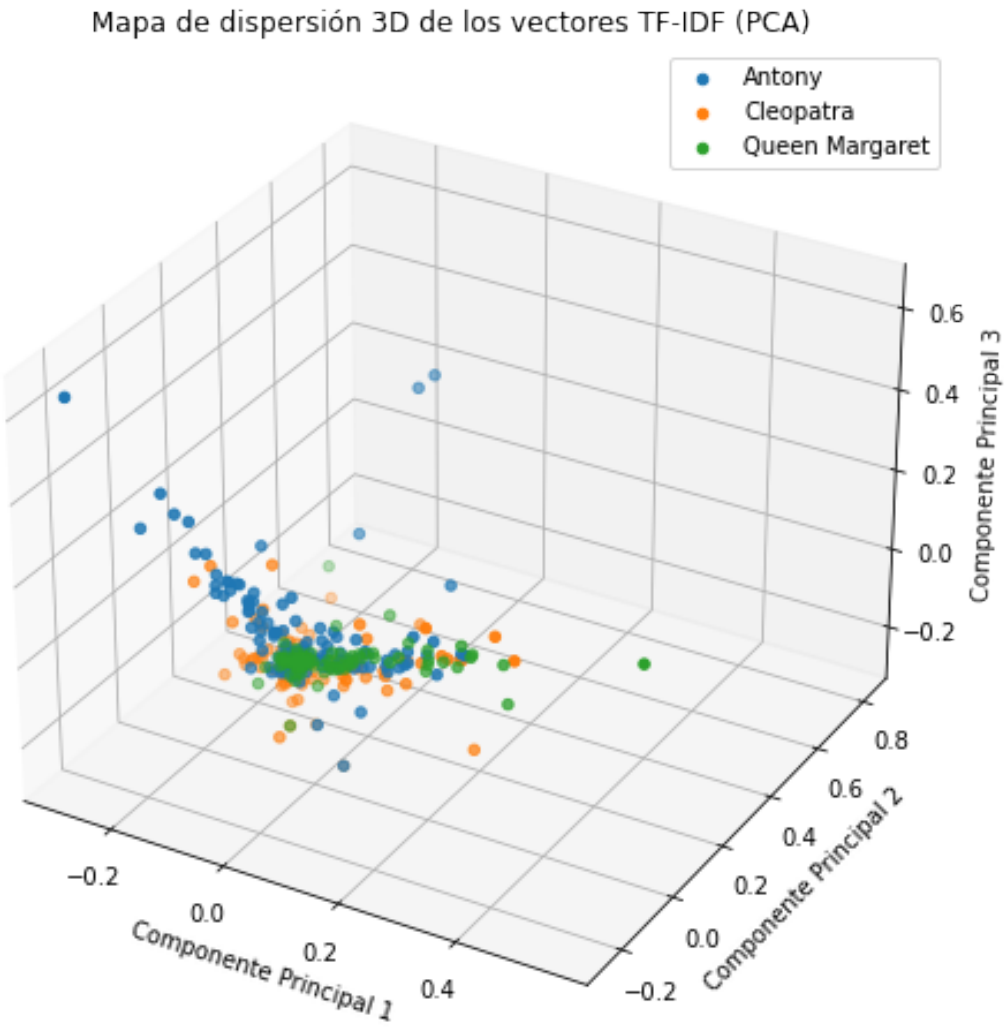

Figura 8: *Mapa de dispersión de los vectores TF-IDF utilizando 3 componentes principales (PCA).*

Accuracy	0.5691		
	Antony	**Cleopatra**	**Queen Margaret**
Precision	0.5	0.6774	1
Recall	0.9342	0.3443	0.2941

Tabla 1: *Métricas obtenidas con el modelo Multinomial Naive Bayes sobre los vectores TF-IDF utilizando mono-gramas y bi-gramas.*

incompleta del desempeño del modelo.

En tales casos, es importante examinar las métricas de precisión, recuperación y la matriz de confusión para obtener una comprensión más completa del rendimiento del modelo en cada clase individual y evaluar su capacidad para clasificar correctamente las clases minoritarias.

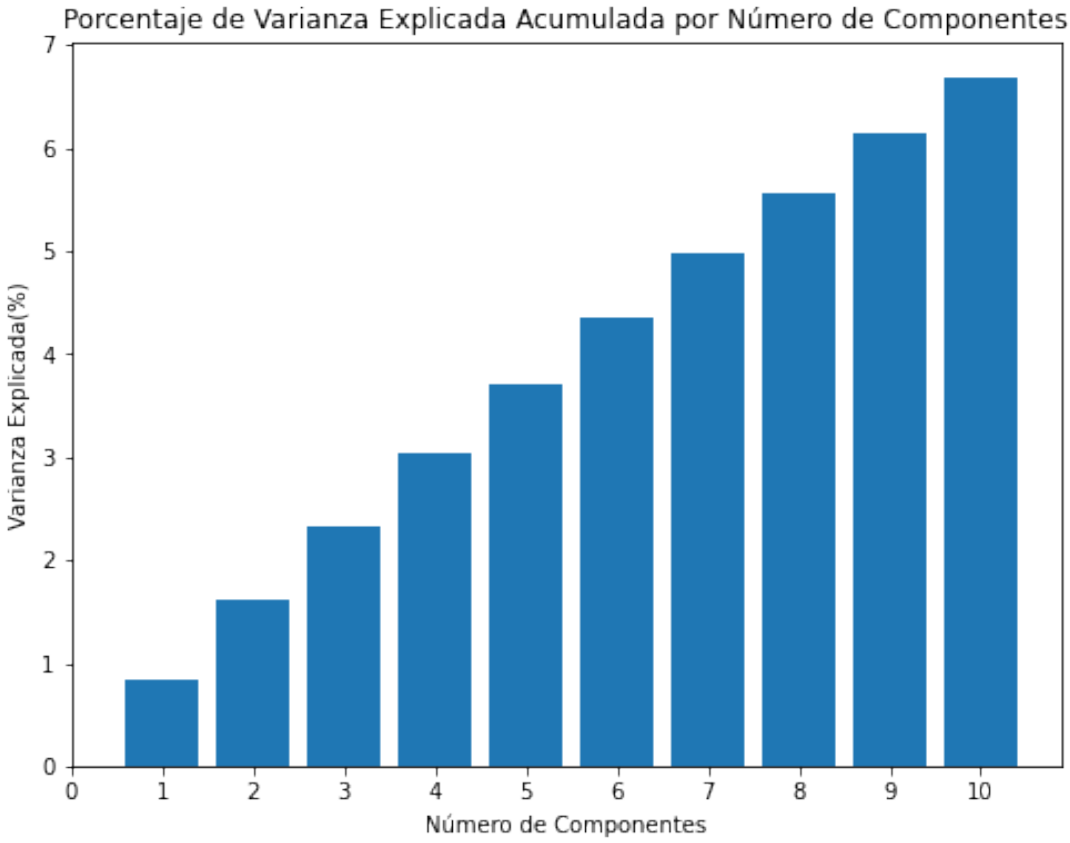

Figura 9: *Porcentaje de varianza explicada acumulada por cantidad de componentes (PCA).*

En la Figura 10 se muestra la matriz de confusión para el modelo indicado, donde podemos verificar lo antedicho con claridad. Notamos que el modelo tiende a predecir la clase mayoritaria, que en este caso corresponde al personaje Antony. Por este motivo, la métrica recall, es decir, la proporción de positivos existentes que encuentra, es alta. Sin embargo, si se observa la métrica de precisión se obtiene la misma cantidad de verdaderos positivos que de falsos positivos.

La segunda clase en orden de dominación corresponde al personaje de Cleopatra. Para este caso, se obtuvo una mejor precisión que en la clase anterior, pero una pésima recuperación, ya que gran parte de los Falsos Negativos se los "lleva" la clase dominante Antony.

Finalmente, para el personaje de Queen Margaret se obtuvo una precisión del 100 %, es decir, "donde pone el ojo, pone la bala", pero tenemos una mala recuperación. Es decir, la proporción de positivos que podemos predecir es baja, porque nuevamente se los "lleva"la clase dominante. Esto último se puede interpretar como que el modelo tiene una excelente precisión para predecir al personaje "Queen Margaret", lo que se puede interpretar como que el modelo logró aprender muy bien alguna característica que permite definir al personaje a partir de las palabras. Por un tema de dominación, no se logra una buena recuperación. Esto lo tomamos como que frente a la duda, el personaje es Antony.

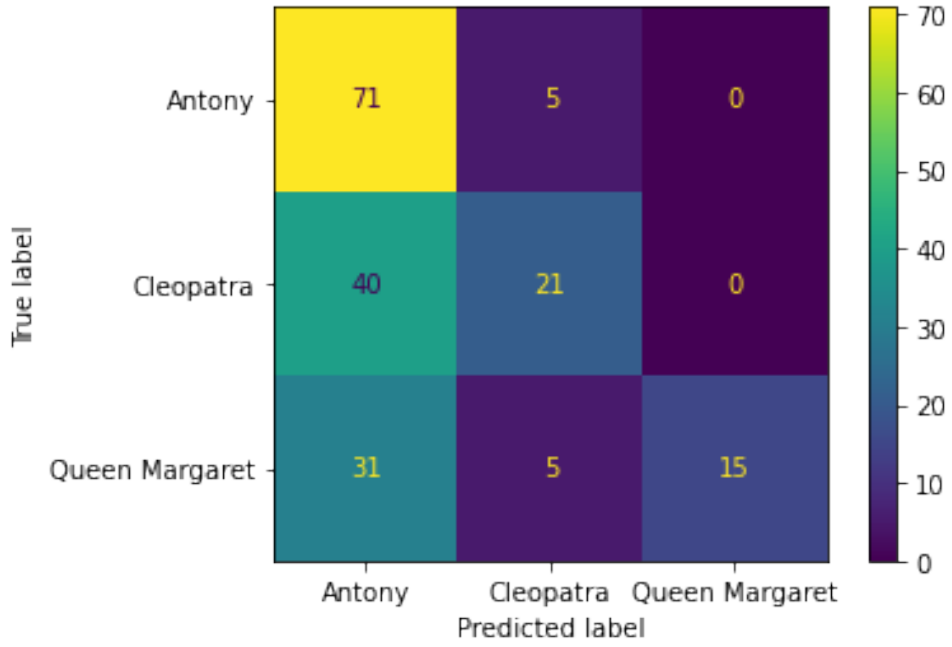

Figura 10: *Matriz de confusión para el modelo Multinomial Naive Bayes sobre los vectores TF-IDF considerando uni-gramas y bi-gramas.*

Cross-Validation

La validación cruzada (cross-validation en inglés) es una técnica utilizada para evaluar y seleccionar modelos de aprendizaje automático de manera más robusta y confiable. En lugar de dividir los datos en un único conjunto de entrenamiento y un único conjunto de prueba, la validación cruzada divide los datos en múltiples subconjuntos llamados "folds" o "pliegues". Luego, se entrena y evalúa el modelo varias veces, utilizando diferentes combinaciones de folds como conjunto de entrenamiento y prueba. Esto permite obtener una estimación más precisa del rendimiento del modelo.

La validación cruzada permite obtener una evaluación más confiable del modelo al utilizar todos los datos disponibles para el entrenamiento y la evaluación. Esto es especialmente útil cuando se dispone de un conjunto de datos limitado y se desea tener una idea más precisa del rendimiento del modelo en diferentes configuraciones.

Para la búsqueda del hiper-parámetro, se recopilaron todas las variantes de parámetros consideradas relevantes en una variable. A continuación, se llevó a cabo el procedimiento mencionado anteriormente y se generó una visualización que permita comparar la accuracy de los distintos modelos entrenados.

Tras observar la Figura 11, el modelo que tiene mayor precisión y menor variabilidad corresponde a aquel que utiliza una representación numérica de los párrafos con la técnica de TF-IDF, filtrando las stop words y considerando únicamente palabras.

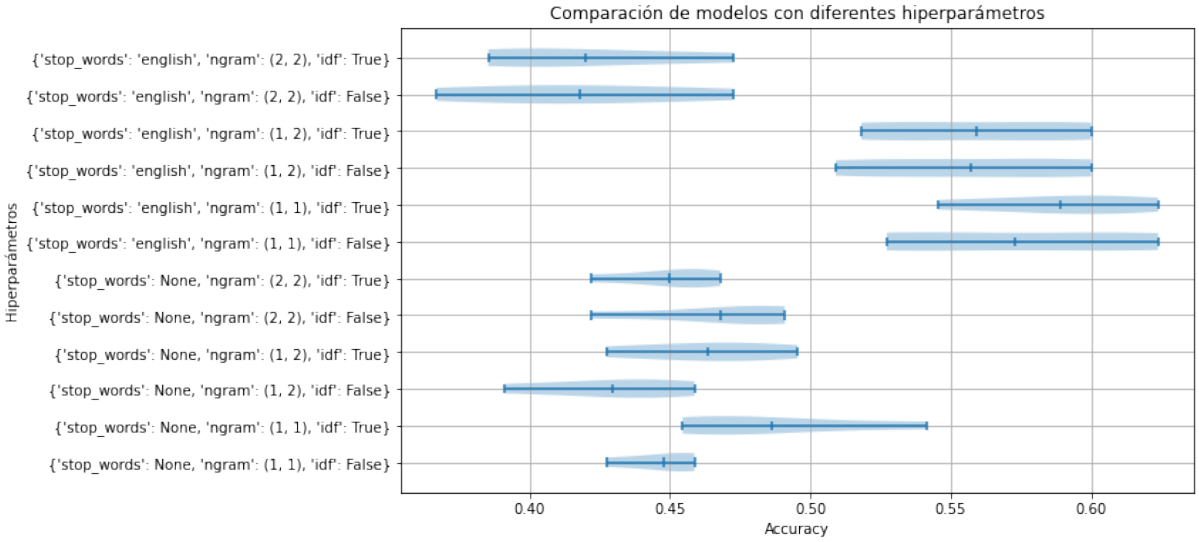

Figura 11: *Cross Validation sobre modelo Multinomial Naive Bayes para la búsqueda de hiperparámetros.*

Consideramos este modelo y volvemos a entrenar sobre todo el conjunto de entrenamiento disponible. Como resultado, obtuvieron los valores de accuracy, precisión y recall que se aprecian en la Tabla 2 y la matriz de confusión de la Figura 12.

Accuracy	0.9105		
	Antony	**Cleopatra**	**Queen Margaret**
Precision	0.8498	0.9371	1
Recall	0.9842	0.8774	0.8402

Tabla 2: *Métricas obtenidas con el modelo Multinomial Naive Bayes al hacer Cross-Validation.*

Se observa que la métrica recall de los tres personajes sube con respecto al modelo anterior, sobre todo para Cleopatra y Queen Margaret.

Modelo Support Vector Machines (SVM)

SVM es un algoritmo de aprendizaje supervisado que se utiliza ampliamente en tareas de clasificación. Este busca encontrar un hiperplano óptimo que pueda separar las diferentes clases. Utiliza vectores de soporte para definir el hiperplano y maximizar el margen entre las clases.

Se entrena el modelo SVM sobre todo el conjunto de entrenamiento disponible. Se obtuvieron los valores de accuracy, precisión y recall que se muestran en la Tabla **??**. Mientras que en la Figura 13 se observa la matriz de confusión obtenida.

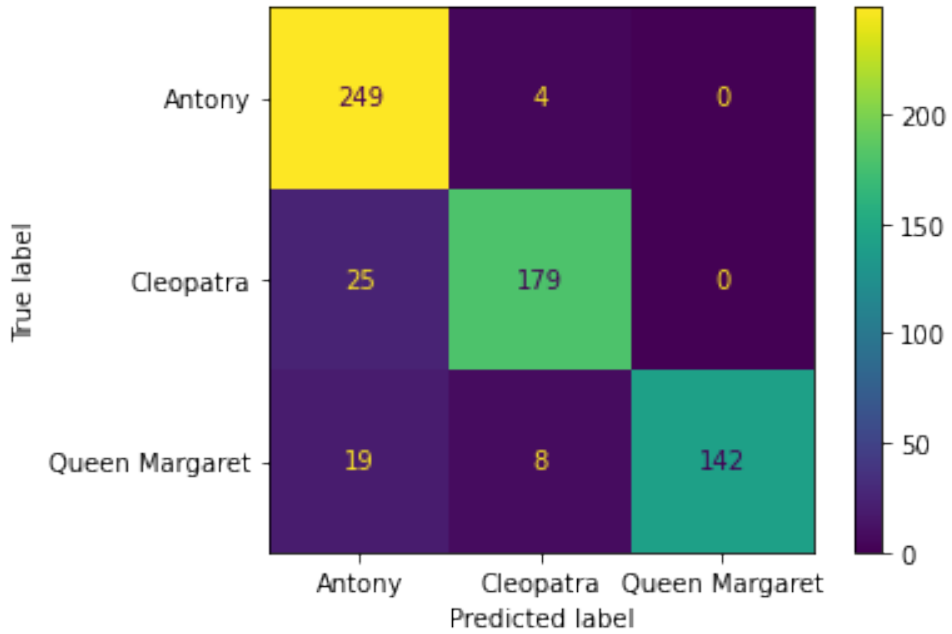

Figura 12: *Matriz de Confusión para el modelo Multinomial Naive Bayes al hacer Cross-Validation.*

Accuracy	0.9744		
	Antony	**Cleopatra**	**Queen Margaret**
Precision	0.9838	00.9481	0.9940
Recall	0.9644	0.9853	0.9763

Tabla 3: *Métricas obtenidas con el modelo SVM.*

En comparación con el modelo Multinomial Naive Bayes, el modelo SVM presenta una mejoría significativa en los valores de accuracy, precisión y recall. Esto sugiere que el modelo SVM tiene una mayor capacidad de clasificación.

Cambio de Personajes

En esta instancia, se procedió a entrenar nuevamente el modelo Multinomial Naive Bayes, pero cambiando uno de los personajes. De acuerdo al análisis hecho anteriormente de la cantidad de párrafos asociados a cada personaje y dado que el objetivo de este punto era observar el desbalance ente clases y los problemas que esto pueda generar, se decidió reemplazar al personaje Antony por Falstaff, que es quien tiene mayor cantidad de párrafos.

Al igual que el primer modelo, se separara un conjunto de prueba que corresponde al 30 % del total de datos disponibles, utilizando muestreo estratificado. La comparación de balances se observa en la Figura 17 del Anexo.

Figura 13: *Matriz de Confusión para el modelo SVM.*

El siguiente paso fue la reducción de dimensionalidad, utilizando PCA. El gráfico de dispersión de la representación obtenida se puede apreciar en la Figura 18 del Anexo. Al igual que antes, se agregan componentes y se observa la varianza explicada en porcentaje. Encontraran más detalles en el Anexo. Lo importante es que no se evidencian resultados o conclusiones distintas a las indicadas anteriormente.

A continuación, se entrena el modelo Multinomial Naive Bayes utilizando la técnica de cross-validation, ya mencionada anteriormente para la búsqueda de hiper-parámetros. El mejor modelo resulta el que utiliza una representación TF-IDF, filtra las stop words y considera únicamente palabras. Los detalles del cálculo los podrán encontrar en el Anexo. Como conclusión importante, los hiper-parámetros que definen el mejor modelo coinciden con los definidos antes de cambiar de personaje. Se obtuvieron los siguientes valores de accuracy, precisión y recall:

Accuracy	0.5944		
	Falstaff	**Cleopatra**	**Queen Margaret**
Precision	1	0.5802	0.75
Recall	0.1147	0.9929	0.0588

Tabla 4: *Métricas obtenidas con el modelo Multinomial Naive Bayes, cambiando de personaje.*

La matriz de Confusión se puede apreciar en la Figura 14. Para este modelo, no solo se mantienen los comentarios anteriormente hechos, sino que al elegir un personaje con

una diferencia aun mayor en la cantidad de párrafos y por tanto en el desbalance, el desempeño del modelo en este caso es aún peor que antes.

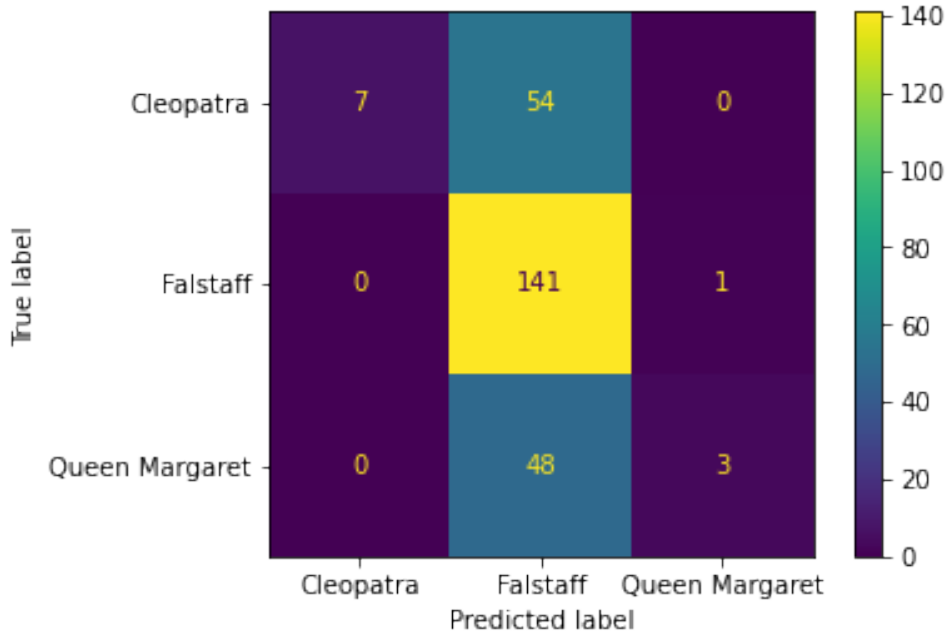

Figura 14: *Matriz de Confusión para el modelo Multinomial Naive Bayes, cambiando de personaje.*

Técnicas alternativas para extraer Feature de Texto

Aunque los modelos basados en bag-of-words o TF-IDF son ampliamente utilizados y pueden ofrecer buenos resultados en muchas tareas de análisis de texto, también presentan ciertas limitaciones que es importante tener en cuenta:

1. Pérdida de la estructura del texto: Estos enfoques tratan el texto como una colección de palabras independientes y no tienen en cuenta la estructura sintáctica y semántica del lenguaje.

2. Ignorancia del contexto: Estos modelos no consideran el contexto en el que se utilizan las palabras. Cada palabra se trata de manera independiente, lo que puede llevar a la pérdida de significado y ambigüedad en ciertos casos.

3. Sensibilidad al ruido y a las palabras poco frecuentes: Estos enfoques son sensibles a palabras poco frecuentes o raras, lo que puede afectar negativamente el rendimiento del modelo.

4. Incapacidad para capturar la semántica: Estos modelos no tienen en cuenta el significado real de las palabras. Como resultado, pueden tener dificultades para capturar matices semánticos, como sinónimos, polisemia, ironía o sarcasmo.

5. Problemas con textos largos: Los modelos basados en bag-of-words o TF-IDF asignan mayor peso a las palabras más frecuentes, lo que puede sesgar el análisis hacia textos más largos. Esto puede ser problemático cuando se trabaja con textos de longitud variable, ya que la información importante en textos cortos puede perderse.

6. Vocabulario limitado: Estos enfoques requieren la creación de un vocabulario fijo antes de la fase de entrenamiento. Esto implica que las palabras fuera del vocabulario no serán consideradas en el modelo, lo que puede ser problemático si se encuentran palabras nuevas o específicas en los datos de prueba.

7. Problemas de desequilibrio de clases: Si hay un desequilibrio significativo en la distribución de las clases objetivo, pueden tener dificultades para capturar la información de las clases minoritarias.

Es importante tener en cuenta estas limitaciones y considerar enfoques más avanzados como Word2Vec, GloVe o FastText, dependiendo de los requisitos y las características específicas del problema de análisis de texto.

En particular, describiremos muy brevemente la técnica de FastText. Este se un algoritmo desarrollado por Facebook Research que se utiliza para el procesamiento de texto y la clasificación de palabras y documentos. A diferencia de otros métodos basados en palabras completas, FastText se basa en la representación de subpalabras (n-gramas) para capturar características y relaciones semánticas.

El algoritmo FastText se basa en dos componentes principales:

1. Modelado de subpalabras: FastText divide cada palabra en subpalabras más pequeñas llamadas n-gramas. Por ejemplo, la palabra .ªpple"se dividiría en los n-gramas .ªp", .ªpp", "ppl", "ple", "le". Esto permite capturar tanto las características de palabras completas como las características compartidas entre palabras similares.

2. Representaciones vectoriales: FastText asigna a cada subpalabra y palabra completa una representación vectorial densa. Cada subpalabra tiene su propio vector, y el vector de una palabra se calcula sumando los vectores de sus subpalabras. Esto permite capturar la información semántica de las palabras y subpalabras en un espacio vectorial.

Una vez entrenado, el modelo FastText se puede utilizar para clasificar palabras o documentos en diferentes categorías. Dado un nuevo texto, FastText descompone las palabras en subpalabras, calcula la representación vectorial de cada subpalabra y luego combina estas representaciones para obtener una representación del texto completo. Luego, el modelo clasifica el texto utilizando estas representaciones y las relaciones aprendidas durante el entrenamiento.

En el Anexo, podrá encontrar la aplicación de este algoritmo sobre el último dateset considerado y los resultados obtenidos. Ya le adelantamos que los resultado son mejores, como era de esperar.

Anexo

Modelo k-NN

El algoritmo k-NN clasifica nuevos puntos de datos basándose en la proximidad a los puntos de datos de entrenamiento. Dado un nuevo punto de datos, k-NN busca los k puntos de datos más cercanos en el espacio de características y determina su clase basándose en la mayoría de las clases de los vecinos más cercanos.

A partir de lo anterior, se entrena el modelo sobre todo el conjunto de entrenamiento disponible. Como resultado, obtuvieron los siguientes valores de accuracy, precisión y recall:

Accuracy	0.9744		
	Antony	**Cleopatra**	**Queen Margaret**
Precision	0.9838	0.9481	0.9940
Recall	0.9644	0.9853	0.9763

Tabla 5: *Métricas obtenidas con el modelo SVM.*

A continuación se presenta el resultado de la matriz de Confusión:

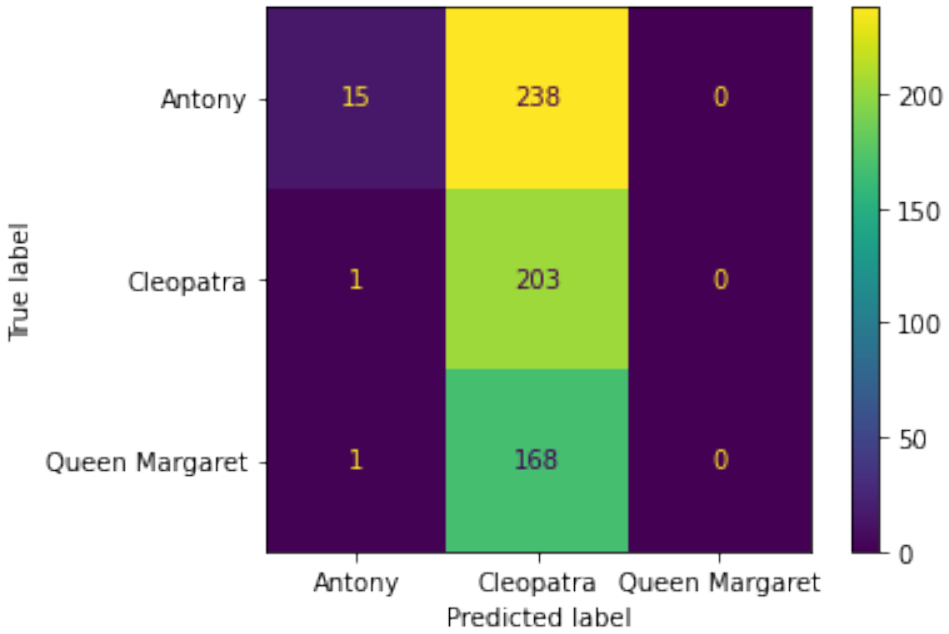

Figura 15: *Matriz de Confusión para el modelo k-NN.*

Este modelo a diferencia de los anteriores, tiene un desempeño considerablemente inferior. A partir de sus valores de accuracy, recall y precisión, sumado al resultado de la matriz de confusión, se deduce que el modelo no está clasificando correctamente las instancias.

Modelo Random Forest

Random Forest es un algoritmo de aprendizaje supervisado que combina múltiples árboles de decisión para realizar predicciones.

A partir de lo anterior, se entrena el modelo sobre todo el conjunto de entrenamiento disponible. Como resultado, obtuvieron los siguientes valores de accuracy, precisión y recall:

Accuracy	0.9776		
	Antony	**Cleopatra**	**Queen Margaret**
Precision	0.9878	0.9528	0.9940
Recall	0.9644	0.9902	0.9822

Tabla 6: *Métricas obtenidas con el modelo Random Forest.*

A continuación se presenta el resultado de la matriz de Confusión:

Figura 16: *Matriz de Confusión para el modelo Random Forest.*

El desempeño de este modelo fue igualmente destacado, al igual que el modelo SVM. Esto se debe principalmente a la mejora significativa en los valores de accuracy, precision y recall. Esto demuestra una mayor capacidad de generalización y un mejor rendimiento del modelo en la tarea de clasificación.

Cambio de Personaje

Comparación de balance de párrafos para cada personaje por los tres conjuntos, prueba, testeo y original.

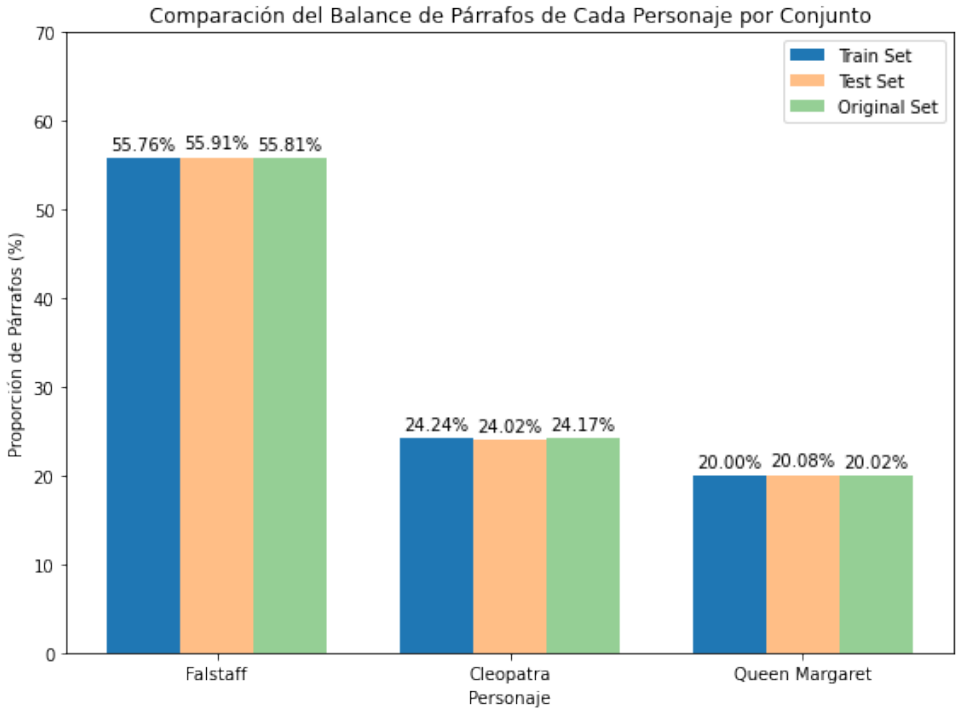

Figura 17: *Balance por cantidad de párrafos por personaje por conjuntos.*

Se observa una notable diferencia entre la proporción de párrafos correspondientes a Falstaff en comparación con los otros dos personajes.

Luego se aplica PCA y se grafica la representación obtenida con dos componentes principales, según se aprecia en la Figura 18

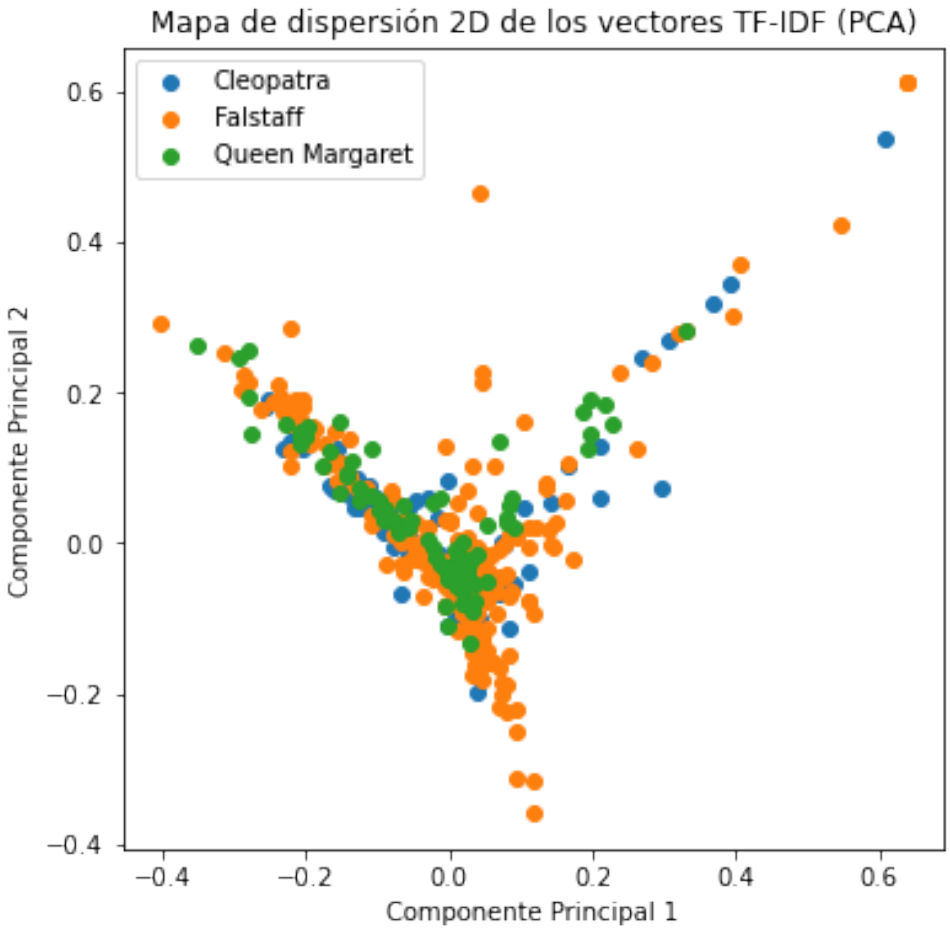

Figura 18: *Mapa de dispersión de los vectores TF-IDF utilizando 2 componentes principales (PCA).*

Posteriormente, se incorporaron un mayor número de componentes principales. En primera instancia, se prueba utilizando tres componentes con el mismo set de parámetros 19.

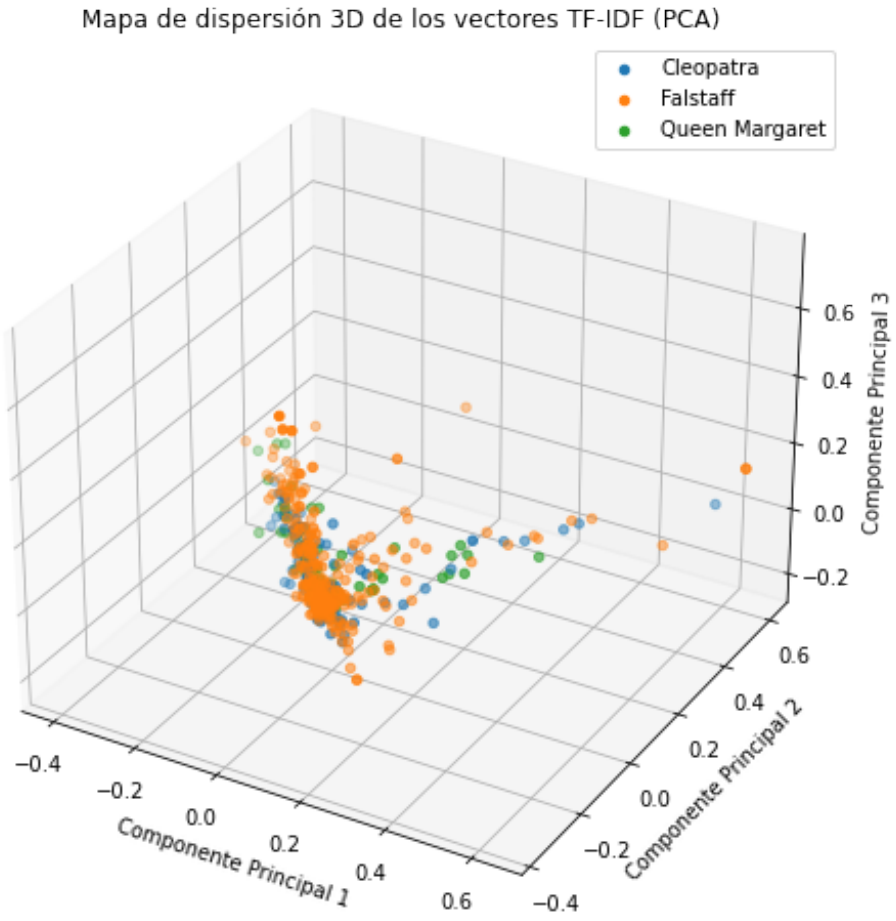

Figura 19: *Mapa de dispersión de los vectores TF-IDF utilizando 3 componentes principales (PCA).*

Por último, se llevó a cabo una visualización con el objetivo de comprender cómo varía la varianza explicada al agregar componentes principales, la cual se muestra en la Figura 20.

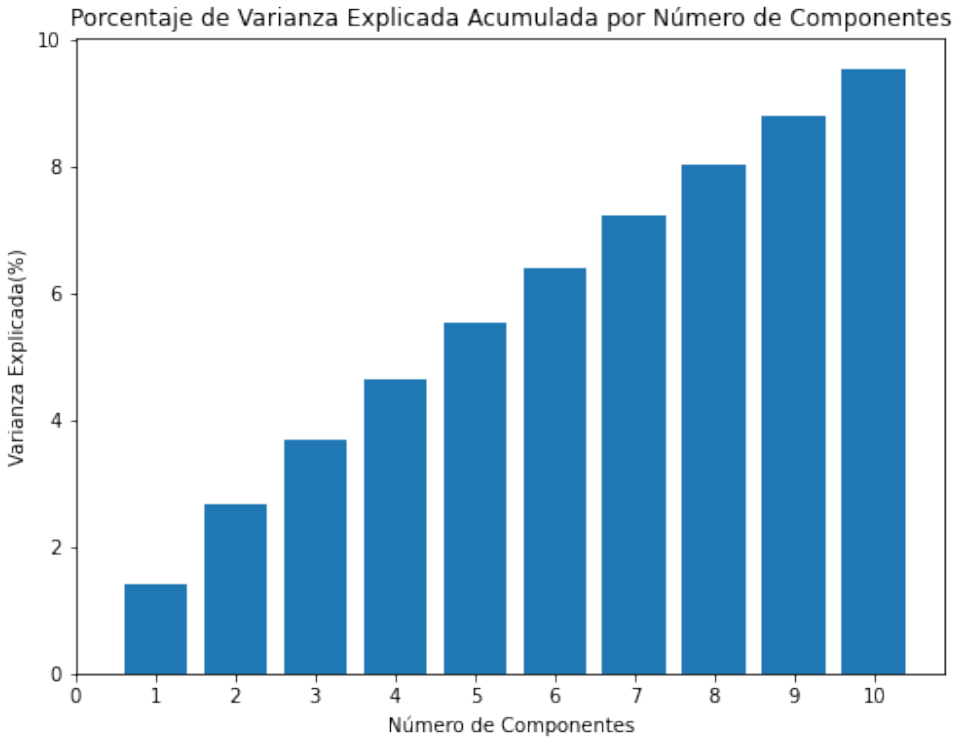

Figura 20: *Mapa de dispersión de los vectores TF-IDF utilizando 3 componentes principales (PCA).*

Como se observa, a medida que se incorporan más componentes a la matriz, se observa un aumento progresivo en la varianza. Este modelo agrega incluso una mayor varianza por cada componente que se agrega en comparación con el primero.

FastText

FastText aplicado sobre el último dataset, es decir, considerando los personajes: Falstaff, Cleopatra y Queen Margaret. En la Figura 21 se puede apreciar la matriz de confusión.

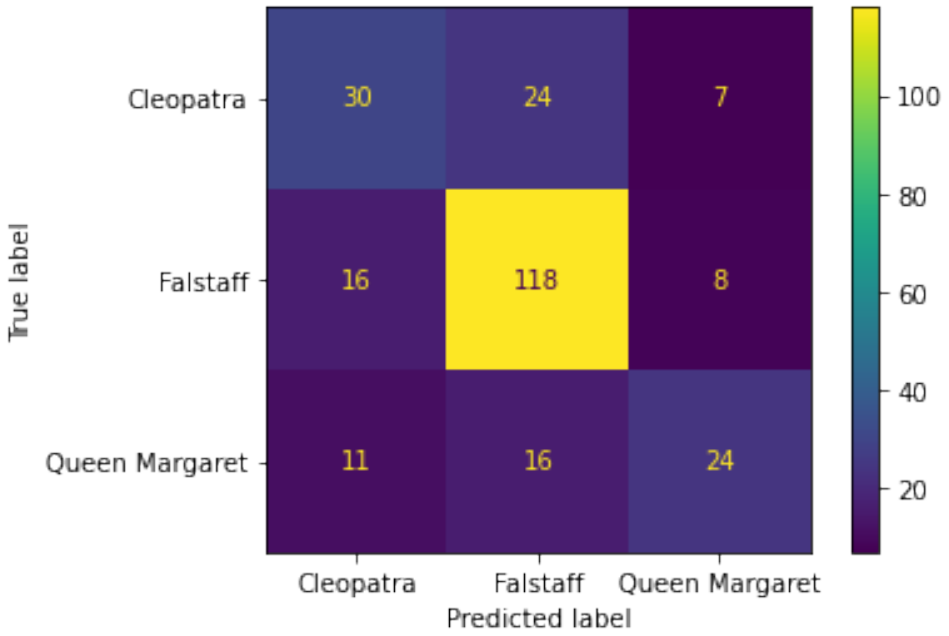

Figura 21: *Matriz de confusión del modelo FastText*