2 de julio de 2024

Introducción a la Ciencia de Datos

Tareas - Grupo 5

Agustín Ghazarian
Sofía Gervaz

Tarea 1

La Tarea consiste en el análisis de una base de datos que contiene información sobre la obra completa de William Shakespeare. La misma está conformada por las siguientes 4 tablas:

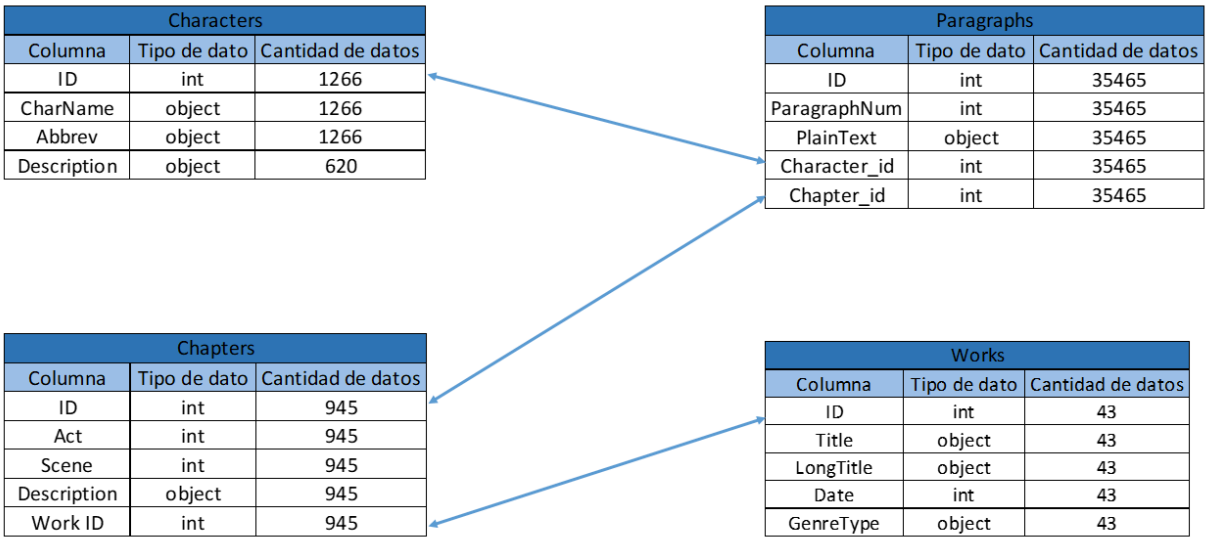
- *Characters*. Tiene los personajes de todas las obras de Shakespeare. Tiene un ID, el nombre y una abreviación para cada personaje. Además posee una columna dedicada a la descripción del mismo. Tiene 1266 entradas (filas), que en general se encuentran casi completas, salvo la columna descripción donde faltan 646 valores.

- *Chapters*. Tiene los actos y escenas de todas las obras de Shakespeare. Tiene un ID para cada acto-escena de cada obra. Otros dos campos numéricos indican por separado el número de acto y de escena. Además posee una descripción de cada escena que menciona la ubicación de la misma y un work_Id, que debe relacionarse con la tabla *Works* indicando de qué obra se trata. Tiene 945 entradas, y no falta ningún dato.

- *Works*. Tiene el nombre de las obras de Shakespeare, en versión corta y larga. Tiene un ID para cada obra que debe relacionarse con el que apareció en la tabla *Chapters*. Además especifica el año y el género para cada obra. Tiene 43 entradas, y no falta ningún dato.

- *Paragraphs*. Tiene información de todos los párrafos de las obras de Shakespeare, incluyendo todo el texto y tanto un ID como un número de párrafo para cada uno. Además, muestra un ID para cada personaje que se debe relacionar con los datos de *Characters* y para cada capítulo que se deben relacionar con los datos de *Chapters*. Tiene 35465 entradas, y no falta ningún dato.

La Figura 1 resume la información relevada:



Characters

Columna	Tipo de dato	Cantidad de datos
ID	int	1266
CharName	object	1266
Abbrev	object	1266
Description	object	620

Paragraphs

Columna	Tipo de dato	Cantidad de datos
ID	int	35465
ParagraphNum	int	35465
PlainText	object	35465
Character_id	int	35465
Chapter_id	int	35465

Chapters

Columna	Tipo de dato	Cantidad de datos
ID	int	945
Act	int	945
Scene	int	945
Description	object	945
Work ID	int	945

Works

Columna	Tipo de dato	Cantidad de datos
ID	int	43
Title	object	43
LongTitle	object	43
Date	int	43
GenreType	object	43

Figura 1 - Descripción de base de datos de la obra de Shakespeare.

Respecto a la calidad de los datos, se observa que las tablas están en general completas, salvo la columna *Description* en la tabla *Characters* en la que faltan más de la mitad de los datos. Los tipos de datos de cada columna son los adecuados, los ID de las tablas son únicos y hay consistencia en las referencias entre ellas (por ejemplo: los *Work_ID* en *Chapters* existen en *Works* y así sucesivamente). No hay filas duplicadas en las tablas.

En la tabla *Characters*, aparece *(stage directions)* como un personaje (siendo además el personaje con más párrafos). Además, existen casos de personajes como *First Apparition*, *First citizen*, *First Gentleman*, etc., que pueden referir a personajes distintos dependiendo de la obra.

En cuanto al análisis de las obras, se observa que las mismas se ubican entre el año 1589 y 1612. El máximo de obras escritas en este periodo fue de 4 para el año 1594, existiendo varios años de 3 obras (1593, 1598, 1599 y 1609) y un solo año en el periodo donde no publicó ninguna obra, 1603. En la Figura 2 se presenta un histograma con la distribución de la cantidad de obras publicadas por año.

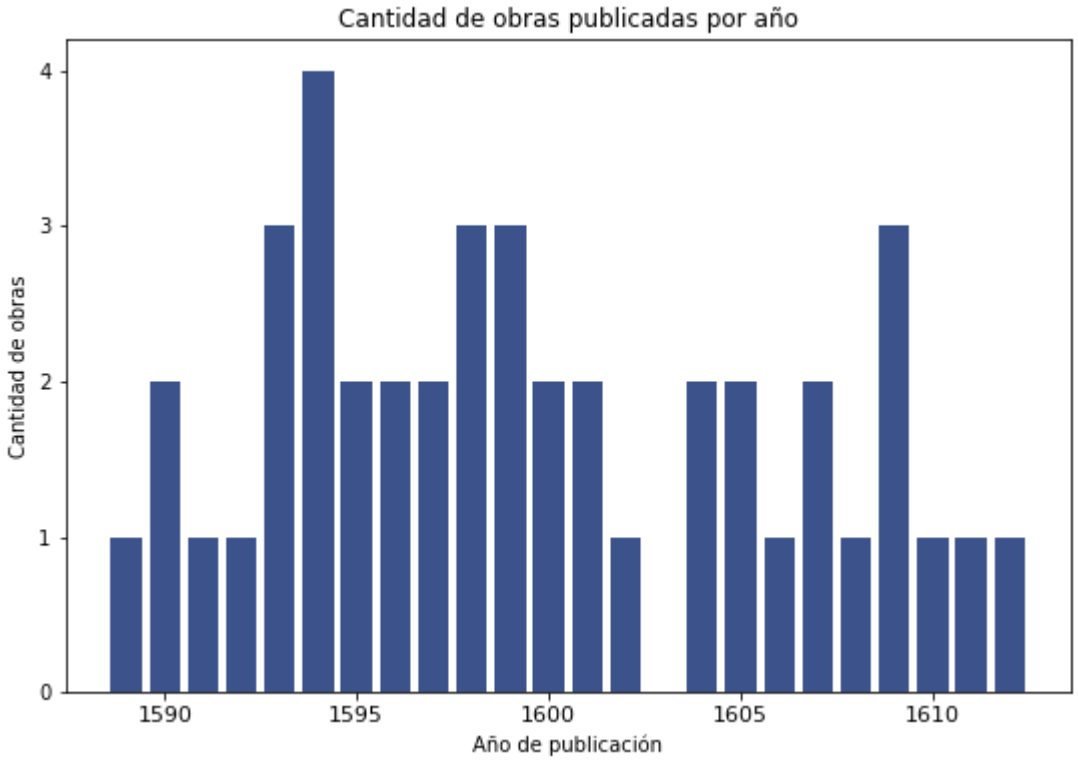


Figura 2 - Histograma de cantidad de obras de Shakespeare publicadas por año.

Al analizar la evolución por género (ver Figuras 3), se observa que las comedias fue lo que escribió de forma más constante a lo largo de toda su carrera (publicando 14 en total), mientras que sólo escribió un soneto. En la primera mitad de su carrera escribió tres veces más de novelas históricas que en la segunda mitad (9 y 3 respectivamente), mientras que las tragedias tienen un comportamiento opuesto.

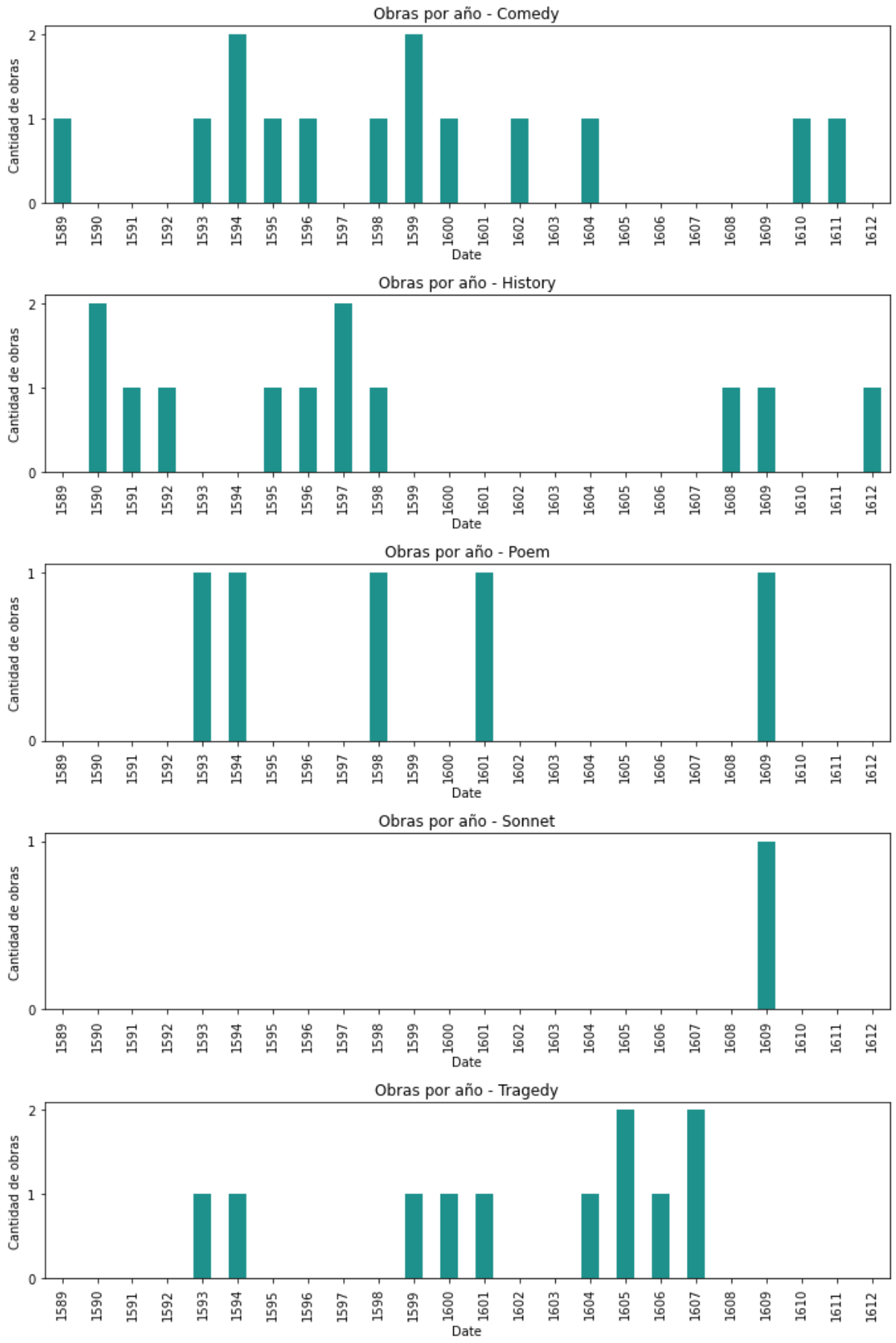


Figura 3 - Histograma de obras de Shakespeare por año para cada género.

Se realizaron además distintas visualizaciones que resumen los gráficos presentados previamente. La gráfica de la Figura 4 permite comparar rápidamente el total de obras en periodos de 4 años y da una idea de cómo se distribuyeron esas obras según el género, pero puede dificultar la comparación de la evolución temporal por género. Por ejemplo, se dificulta obtener de forma rápida la evolución de tragedias a lo largo del tiempo por encontrarse “apilado” encima del resto de los géneros. lo mismo sucede con todos los géneros, salvo la comedia que se encuentra debajo.

A partir de la Figura 8 se puede observar que entre 1593 y 1600, el autor publicó 21 de sus 43 obras. De estas 21, la mayoría son comedias seguidas por obras históricas, tragedias y poemas, en ese orden. Se observa también que al inicio de su carrera los géneros preponderantes eran las comedias y las obras históricas, que fueron dando lugar a las tragedias. En particular, entre 1605 y 1608, de las 6 obras que publicó, 5 fueron tragedias. En el último tramo de su carrera, sin embargo, la producción estuvo repartida entre comedias, género histórico y poemas y, además, publicó su único soneto.

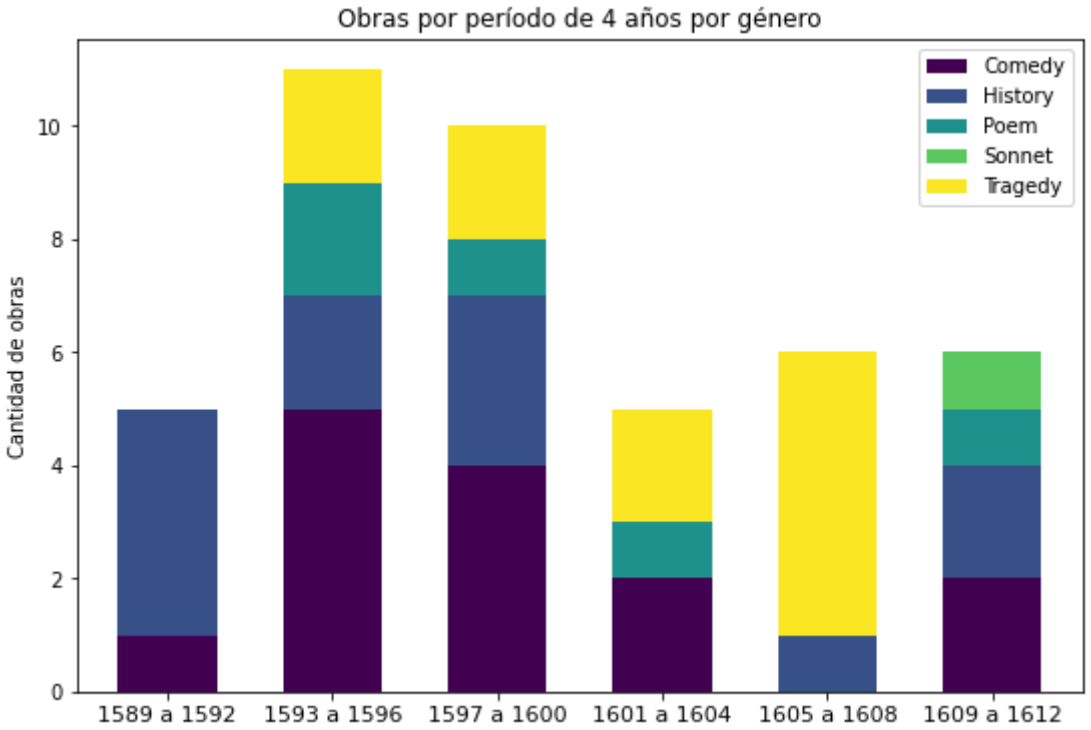


Figura 4 - Histograma obras de Shakespeare por año, columnas apiladas según género.

Se realizó la siguiente visualización en modo de mapa de color (ver Figura 5), se cree que esta alternativa es eficiente para analizar tanto evoluciones temporales por género como distribución por géneros en un periodo de tiempo, no así para analizar la evolución del total de obras presentados por periodo.

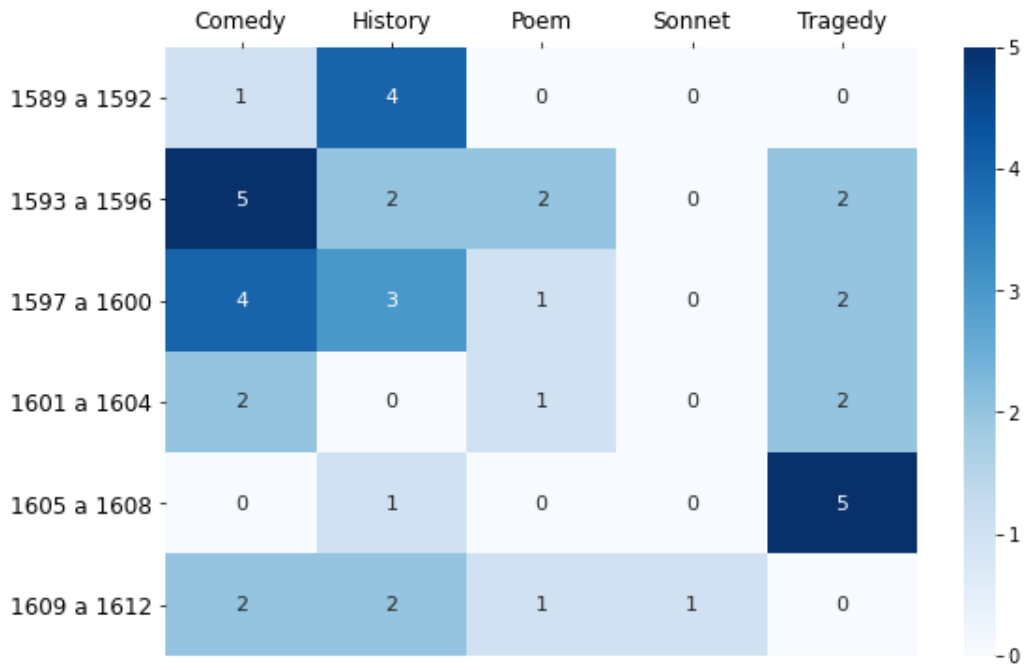


Figura 5 - Cantidad de obras publicadas por Shakespeare por periodos de 4 años y por género.

Para poder analizar el texto de las obras de Shakespeare, se debe procesar el texto de forma de eliminar mayúsculas y signos de puntuación. Para esto se utilizó la función *CleanText*, que pone todo el texto en minúscula y elimina los signos de puntuación ",",";",":","!","?","-","""," ". Tras realizar esta transformación, analizamos las palabras más populares en la obra de Shakespeare, que se muestran a continuación.

Para analizar las palabras más populares por género o personaje se podría agregar una columna género y otra columna personaje al dataframe. La visualización de mayor cantidad de palabras por género y personaje se podría observar en un mapa de calor como la imagen previa, donde en un eje estuvieran los distintos géneros, en el otro eje las palabras más populares y se muestre la cantidad de apariciones de cada una de esas palabras para cada género.

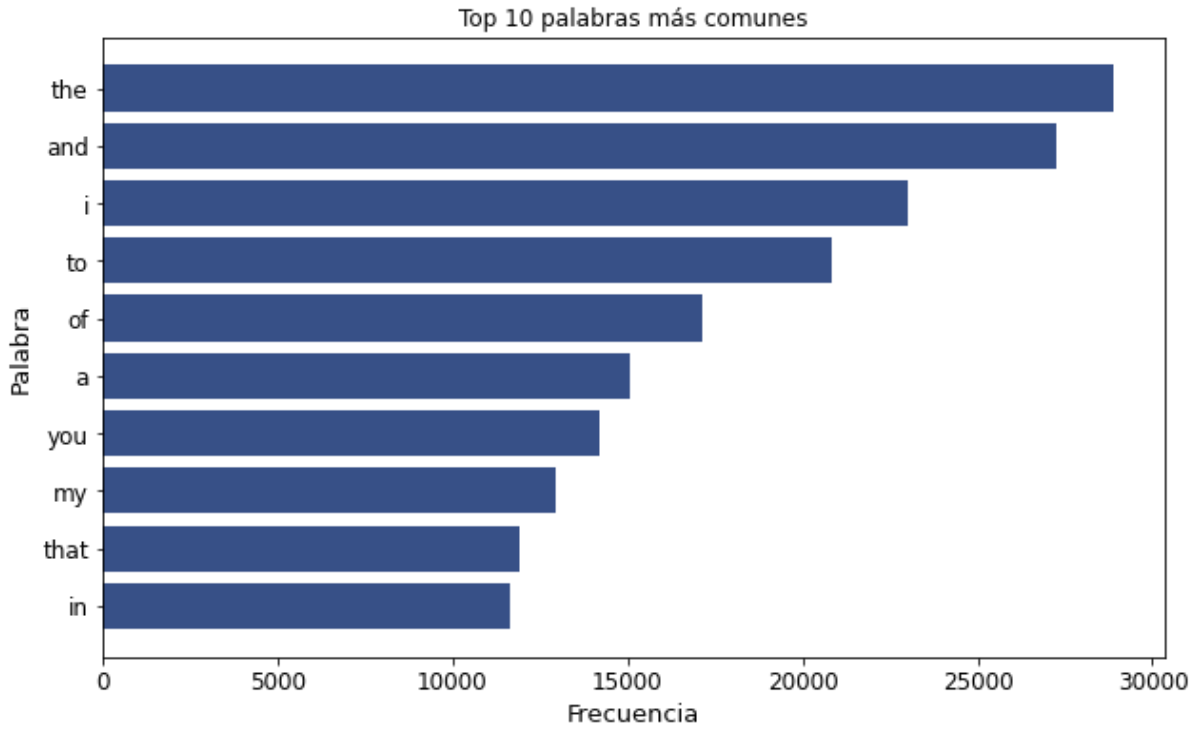


Figura 6 - Cantidad de menciones a las palabras más frecuentes en la obra de Shakespeare.

En caso de sólo querer hacer un análisis cualitativo de las palabras usadas con más frecuencia se puede optar por presentarlas en forma de *wordmap* (ver Figura 7).

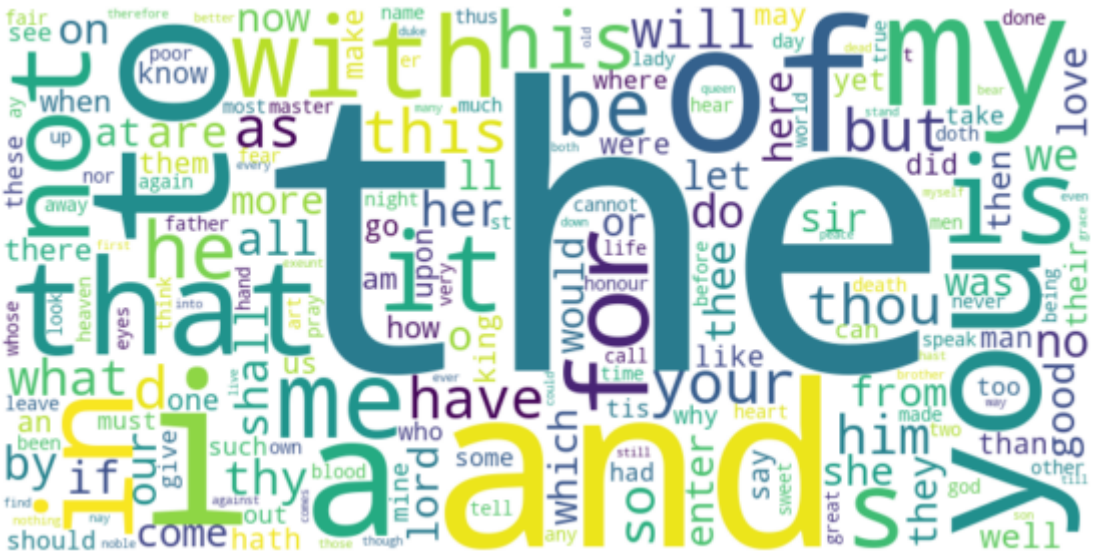


Figura 6 - Palabras más frecuentes en la obra de Shakespeare.

Si bien el personaje que aparece con más palabras es “Poet”, se entiende que el mismo puede referirse a poetas distintos lo que además explicaría la amplia diferencia que presenta frente a “ Henry V” y “Falstaff”. Por lo tanto se considera que el personaje con mayor cantidad de palabras es Henry V. Para ver efectivamente los 10 personajes con más palabras se podría eliminar la primera entrada del dataframe.

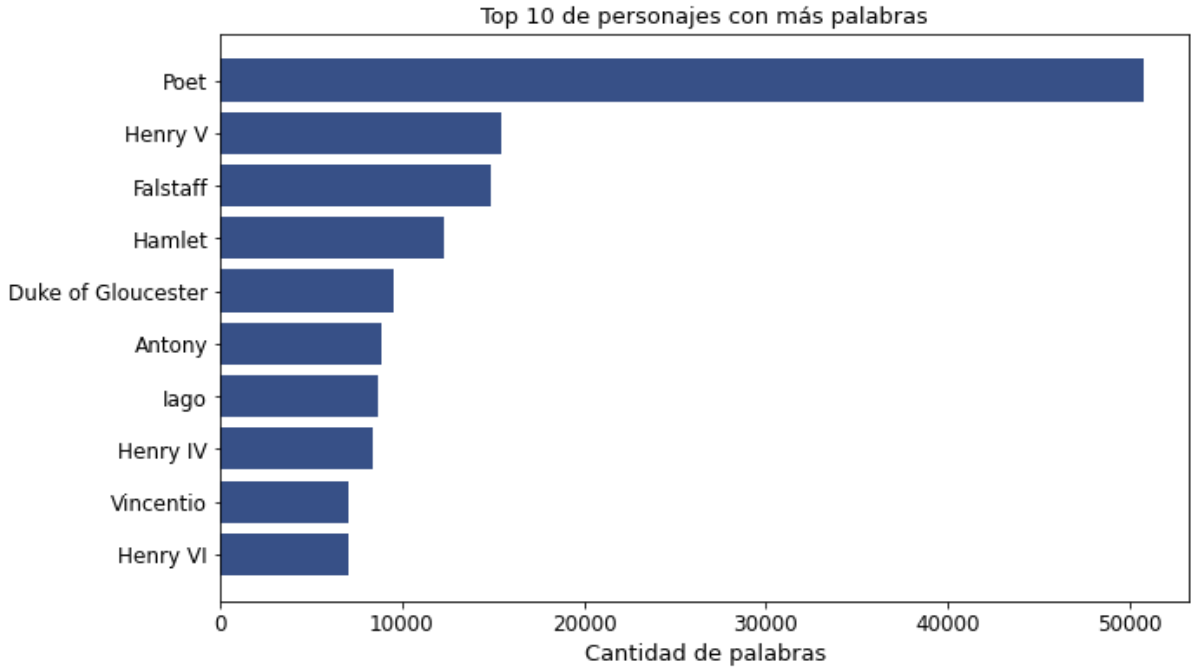


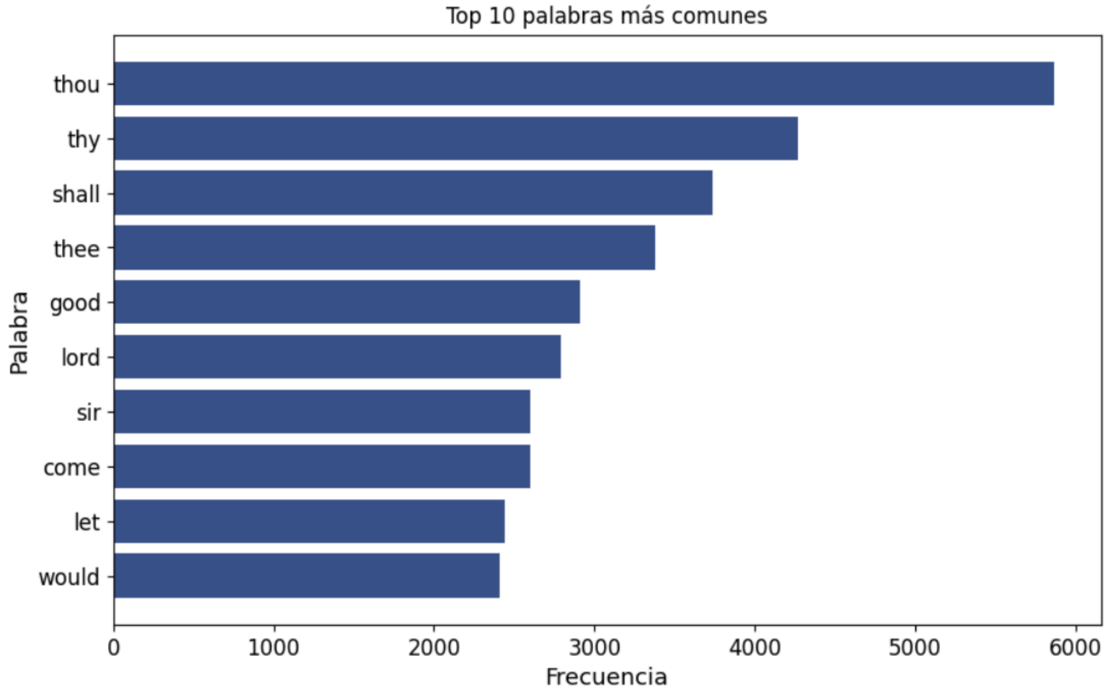
Figura 11 - Cantidad de palabras por personaje en las obras de Shakespeare.

Más allá del análisis realizado, el set de datos permite extraer otras conclusiones como:

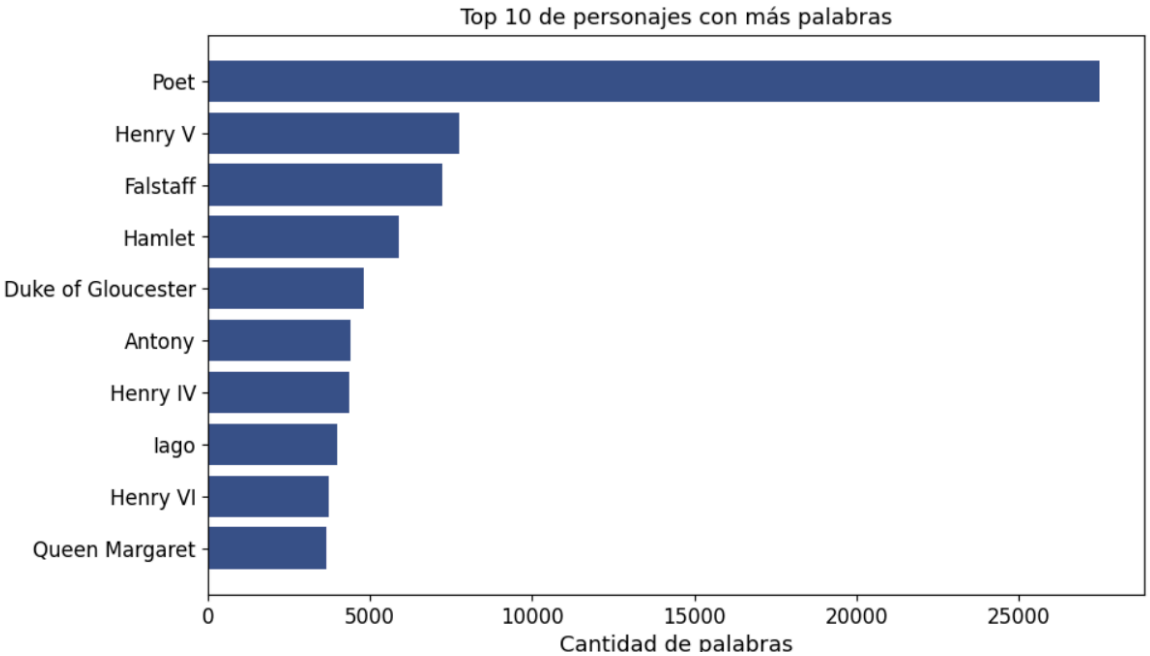
- ¿Cuál es la obra más larga de Shakespeare? ¿Y la más corta?
- ¿Qué género presenta el mayor largo por obra promedio? ¿Y el menor?
- Análisis de la estructura de las obras ¿existe alguna tendencia temporal en cuanto al largo de sus obras? ¿Y en cuanto a la cantidad de escenas y actos?¿Y según el género?
- ¿Qué personaje aparece en más párrafos? ¿Y en más capítulos?
- Para alguna obra en particular ¿cuáles son los personajes principales?¿Cómo es la evolución de la participación de los personajes?
- Comparación de palabras más recurrentes diferenciando por género y/o por año de publicación.

Correcciones a la Tarea 1

De acuerdo a las sugerencias realizadas en la devolución de la tarea 1, se han filtrado las stop words. Los nuevos resultados se muestran a continuación, comparadas con los resultados anteriores


Top 10 palabras más comunes
thou
thy
shall
thee
good
lord
sir
come
let
would
Palabra
0
1000
2000
3000
4000
5000
6000
Frecuencia


thou
thy
shall
thee
would
love
good
sir
lord
come
let
well
must
may
say
hath
like
upon
enter
man
see
one
know
make
king
go
yet
mine
tis
made
never
hand
honour
heart
speak
take
give
time
true
tell
think
father
death
night
life
day
lady
sweet
great
art
eyes
fear
hear
look
us


Top 10 de personajes con más palabras
Poet
Henry V
Falstaff
Hamlet
Duke of Gloucester
Antony
Henry IV
Iago
Henry VI
Queen Margaret
0
5000
10000
15000
20000
25000
Cantidad de palabras

Tarea 2

En esta etapa se entrenará un modelo de lenguaje natural utilizando la base de datos presentada anteriormente. Se utilizará una versión reducida de la misma, considerando los personajes “Antony”, “Cleopatra” y “Queen Margaret”. En la Tabla 1 se presenta la cantidad de párrafos para cada uno de estos personajes.

Para estos personajes se dividieron los párrafos en *train* y *test*, siendo los mismos el 70% y 30% respectivamente. El tamaño de *train* alcanza los 433 párrafos mientras que para *test* es de 188. La proporción de párrafos de cada personaje en *train* y *test* se observa en la Figura 1.

Tabla 1 - Cantidad de párrafos por personaje.

Personaje	Total
Antony	253
Cleopatra	204
Queen Margaret	169

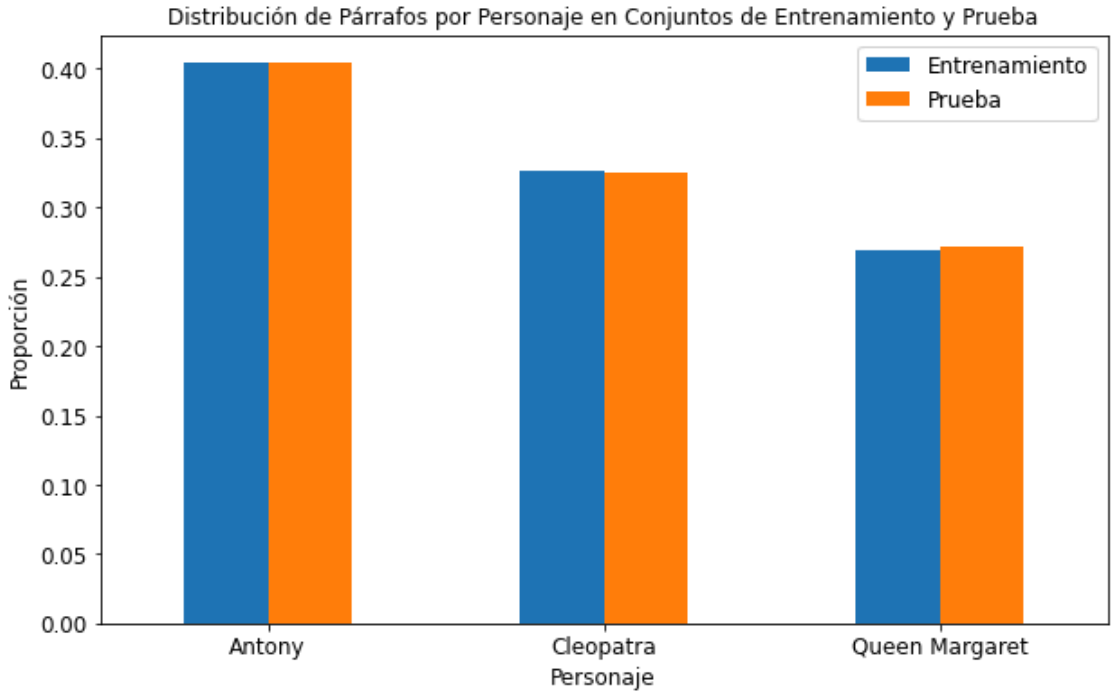


Figura 1 - Balance de párrafos por personaje en train y test.

Se procede a transformar el texto de entrenamiento a representación numérica utilizando *bag of words*. Este método asigna una fila a cada párrafo y una columna a cada palabra que se encuentra. Cada posición i,j de la matriz es completada con la cantidad de veces que aparece la palabra j en el párrafo i.

A modo de ejemplo, se presenta el siguiente texto formado por dos párrafos:

Párrafo 1:
El gato negro duerme en la silla. La casa está en silencio.

Párrafo 2:
La luna brilla en el cielo. El perro ladra en el jardín.

Con estos párrafos, la matriz *bag of words* es la que se presenta en la Tabla 2.

Tabla 2 - Matriz *bag of words* para caso de ejemplo.

	brilla	casa	cielo	duerme	el	en	está	gato	jardín	la	ladra	luna	negro	perro	silencio	silla
Párrafo 1	0	1	0	1	1	2	1	1	0	2	0	0	1	0	1	1
Párrafo 2	1	0	1	0	3	2	0	0	1	1	1	1	0	1	0	0

Como los textos suelen tener muchos párrafos, lo habitual es que cada párrafo tenga pocas de las palabras que aparecen en todo el conjunto de entrenamiento. Por lo tanto, la matriz *bag of words* resultante es mayoritariamente ceros. La representación de matriz dispersa aprovecha esta característica, guardando únicamente aquellos valores no nulos y su posición. Con este mecanismo se ahorra un gran uso de memoria que se destinaría a almacenar valores nulos.

De igual modo al método *bag of words*, el método *Term Frequency-Inverse Document Frequency* (TF-IDF) es una representación numérica de un determinado texto, pero donde cada término recibe un peso basado en su frecuencia de ocurrencia en el párrafo y su rareza en el conjunto de párrafos. Esta transformación tiene en cuenta tanto la importancia local (frecuencia del término en el párrafo) como la importancia global (inversa de la frecuencia del término en todos los párrafos). De esta forma, las palabras que se repiten mucho en cada párrafo pero además se repiten en el conjunto de párrafos (por ejemplo *stop words*), pierden relevancia. Dando entonces más peso a aquellas palabras “claves” al momento de diferenciar personajes.

Un N-grama es una secuencia contigua de N elementos (palabras o caracteres) de un texto dado. Los N-gramas se utilizan comúnmente en procesamiento de lenguaje natural para modelar el contexto local dentro del texto.

El método de componentes principales (PCA) sirve para reducir la dimensionalidad de problemas complejos. En este caso, se tiene un conjunto de 438 párrafos formados por 2,807 palabras. Se trata de encontrar los ejes que expliquen de mejor modo la nube de puntos existente. En las Figuras 2 y 3 se presentan las gráficas resultantes al reducir la dimensionalidad del problema a 2 componentes; en la Figura 2 para el conjunto de entrenamiento original y en la Figura 3 para el conjunto de entrenamiento donde se filtraron las *stop words,* se usó IDF y se consideraron bigramas.

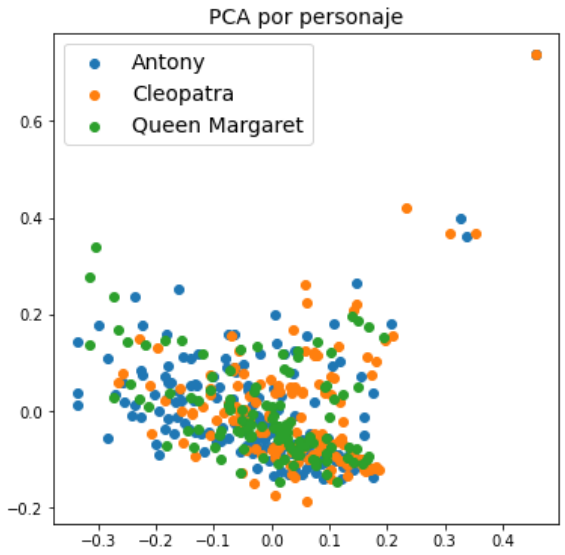


Figura 2 - PCA. dos componentes

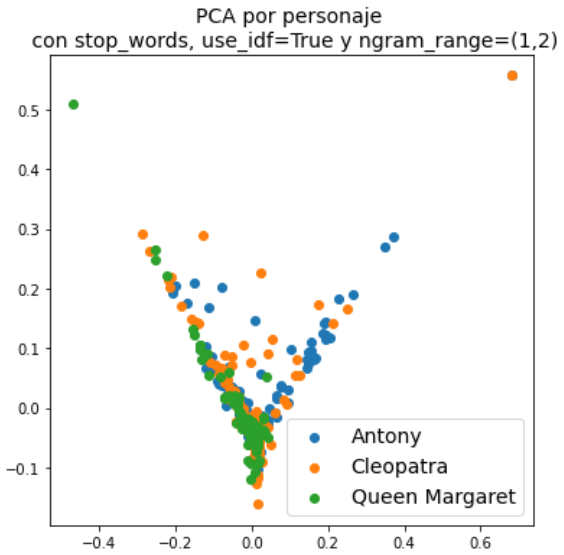


Figura 3 - PCA dos componentes, stop_words=english, use_idf=True, ngram_range(1,2)

Debido a que la reducción de dimensionalidad es muy elevada, pasando de un problema de dimensión 2,807 a un problema de dimensión 2, se pierde gran parte de la información por lo que la varianza explicada por el nuevo modelo es baja (ver Figura 4).

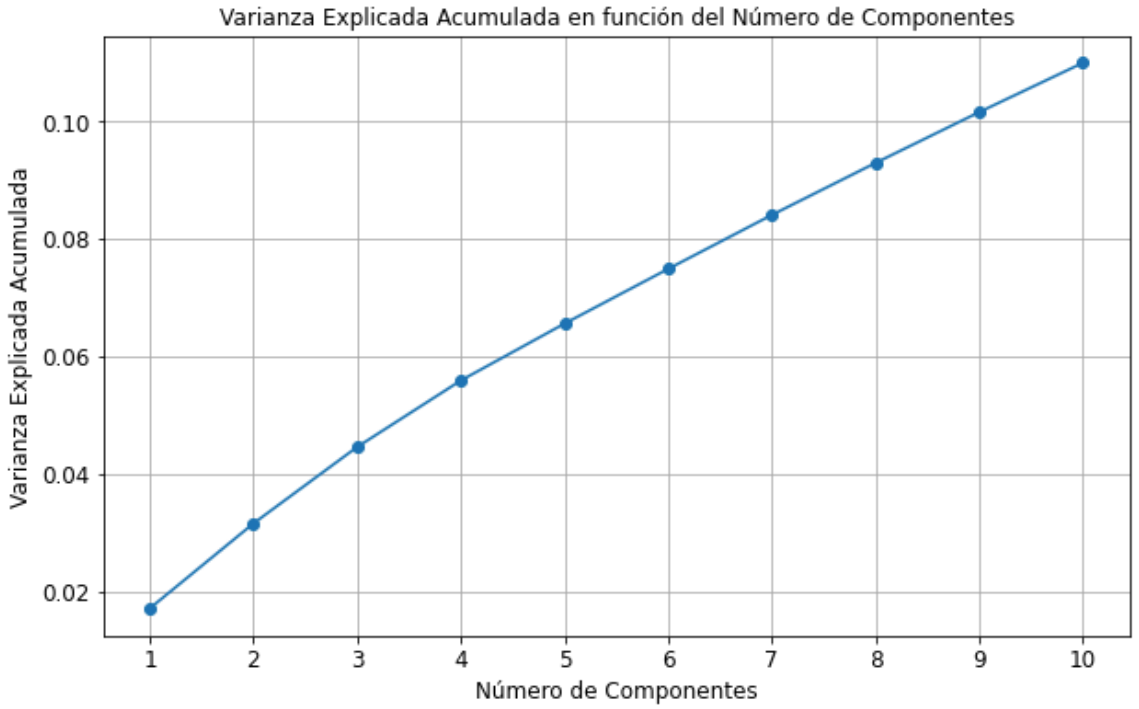


Figura 4 - Varianza explicada vs Número de componentes de PCA utilizados.

Al aplicar el filtrado de *stop words*, el uso de IDF y considerar bigramas, la representación TF-IDF se vuelve más útil para discriminar entre personajes, con una mayor separación en los puntos entre personajes y un menor solapamiento (ver Figs. 2 y 3).

Se entrenó el modelo Multinomial Naive Bayes y se evaluarán los resultados en términos de *accuracy*, matriz de confusión, precisión y *recall*.

Accuracy: El valor de *accuracy* mide la proporción de predicciones correctas sobre el total de predicciones. Sin embargo, no evalúa los resultados para cada personaje ni evalúa si los mismos se encuentran balanceados.

Matriz de Confusión: La matriz de confusión muestra las verdaderas etiquetas frente a las etiquetas predichas, proporcionando una visión detallada de los errores de clasificación para cada clase.

Precision y *Recall*:

- Precisión: Mide la proporción de predicciones correctas sobre el total de predicciones realizadas para un personaje específico.
- Recall: Mide la proporción de predicciones correctas sobre el total de párrafos realmente asignados al personaje.

El valor de *accuracy* hallado es 0.5 mientras que la siguiente tabla muestra tanto el parámetro precisión como *recall* para cada personaje y la matriz de confusión correspondiente.

Observar solamente el valor de *accuracy* (variable que considera la totalidad de las predicciones, no cada personaje en particular) puede llevar a error en el nivel de confiabilidad en la predicción en cada personaje. Por ejemplo, si la muestra fuera extremadamente desbalanceada, el modelo puede tender a asignar todos los párrafos al personaje con mayor peso, y eso daría como resultado un alto valor de *accuracy*, aunque el modelo falla en predecir realmente los párrafos correspondientes a otros personajes.

Tabla 2 - Parámetros de precisión y recall obtenidos.

	Precisión	*Recall*
Antony	0.91	0.20
Cleopatra	0.63	0.20
Queen Margaret	0.64	0.95

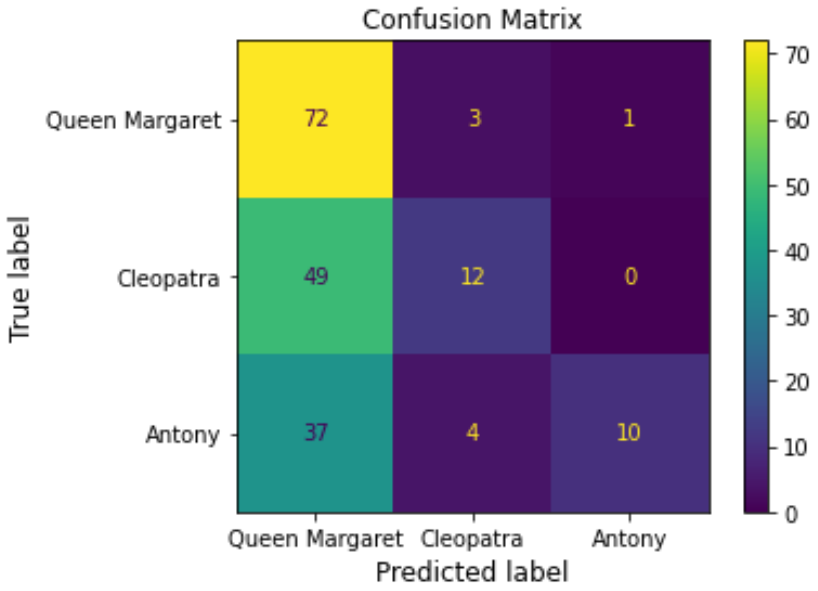


Figura 5 - Matriz de confusión.

La técnica de validación cruzada es un método para evaluar el rendimiento de un modelo. Consiste en dividir el conjunto de datos en múltiples subconjuntos o *folds*. El modelo se entrena en una combinación de estos *folds* y se evalúa en el *fold* restante. Este proceso se repite varias veces, y los resultados se promedian para obtener una estimación más confiable del rendimiento del modelo.

Del análisis anterior se desprende que para nuestro caso el mejor modelo resulta con los parámetros (ngram_range=(1,1), use_idf=False, alpha=0.01) (ver Figura 6). Utilizando este modelo se alcanza un valor de *accuracy* de 0.63, y los siguientes valores de precisión y *recall* (ver Tabla 3).

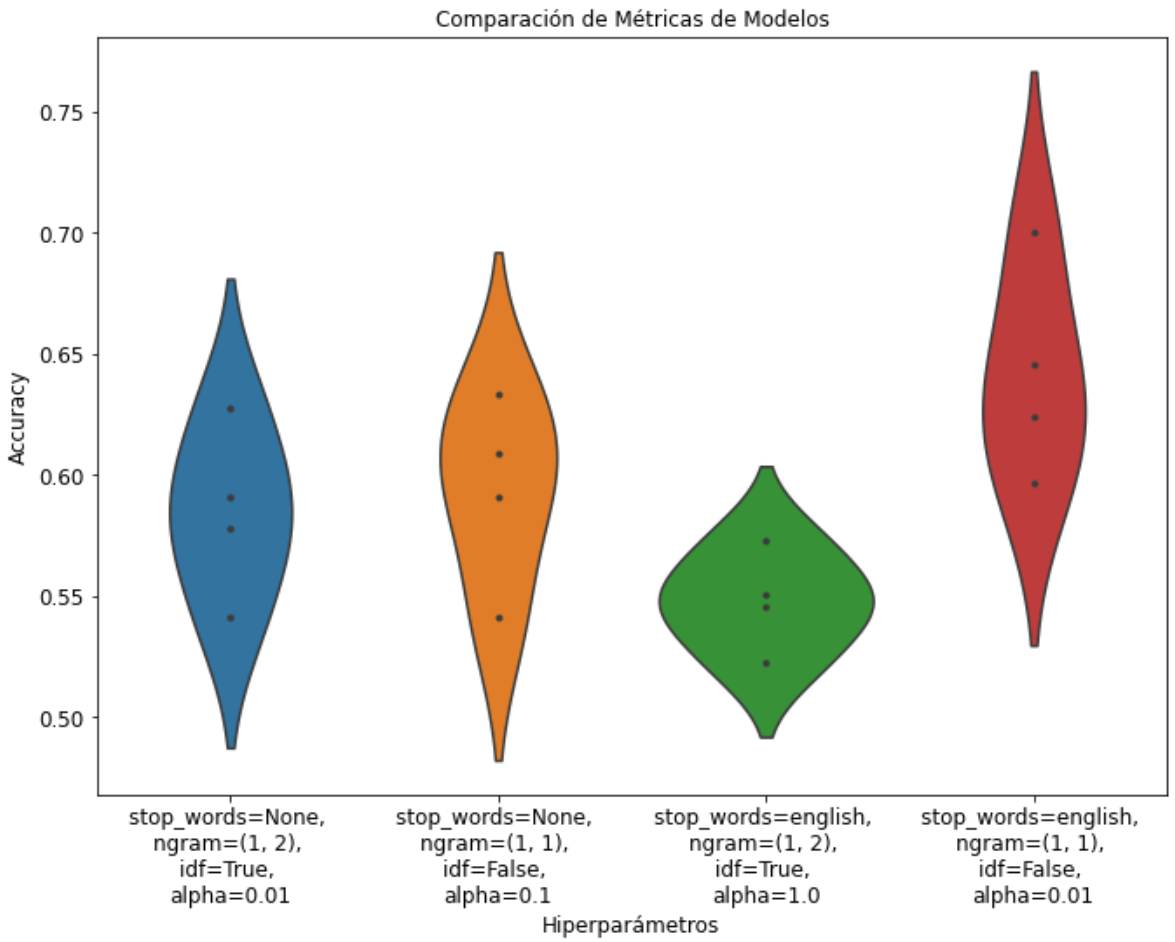
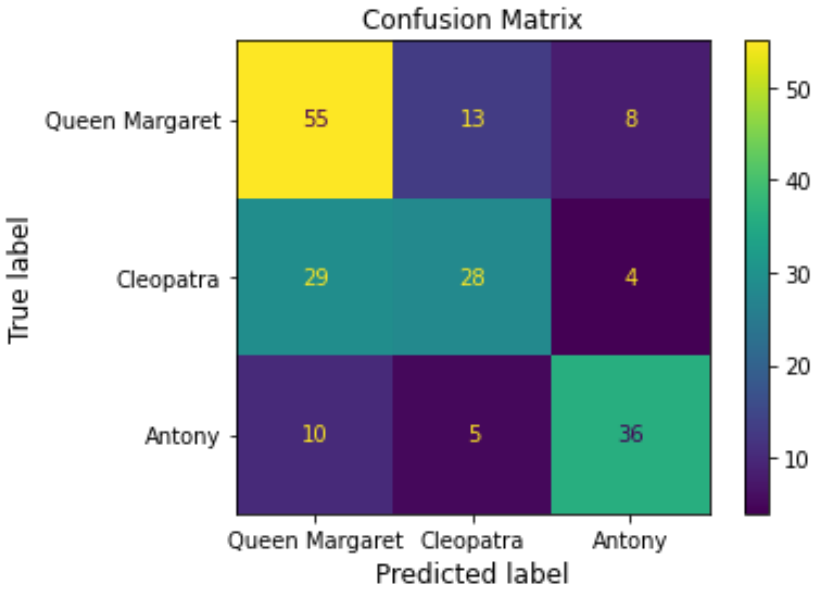


Figura 6 - Gráfico de violín para distintos parámetros de entrenamiento de modelo.

Tabla 3 - Parámetros de precisión y *recall* obtenidos.

Personaje	Precisión	*Recall*
Antony	0.75	0.71
Cleopatra	0.61	0.46
Queen Margaret	0.59	0.72



Una de las limitaciones asociadas al uso de un modelo basado en *bag of words* es la abordada en el método TF-IDF y es que con *bag of words* no se considera la importancia relativa de las palabras, resultando que no se tenga en cuenta que hay palabras más informativas que otras. Además, el modelo no tiene en cuenta el orden de las palabras en el texto, haciendo que se pierda contexto y que frases con distinto significado puedan tener igual representación. Por el contrario, al no capturar relaciones semánticas, palabras que son en realidad sinónimos van a representarse por dimensiones distintas. Además, el modelo suele llevar a representaciones de dimensionalidad alta y podría ser difícil de manejar cuando se trabaja con grandes volúmenes de datos.

Al igual que el bag of words, TF-IDF no considera el orden de las palabras, por lo que se pierde contexto y sintaxis. Tampoco captura relaciones semánticas, haciendo que palabras que son sinónimos se representen como dimensiones separadas. Esta técnica también suele llevar a representaciones de dimensionalidad alta que pueden ser difíciles de manejar. Por otro lado, a diferencia de *bag of words,* al considerar la importancia relativa de las palabras, hace que la técnica pueda ser sensible a la longitud del documento a estudiar.

Para evaluar otro modelo además del Multinomial Naive Bayes, vamos a utilizar el clasificador de Máquinas de Soporte Vectorial (*Support Vector Machine*, SVM). Este algoritmo de clasificación busca encontrar el hiperplano que mejor separa las diferentes clases en el espacio de características. El objetivo del SVM es maximizar el margen, es decir, la distancia entre el hiperplano separador y los puntos de datos más cercanos de cada clase, que se llaman vectores de soporte.

El *accuracy* resultante es de 0.62 y los valores de precisión y *recall* son los que se presentan en la Tabla 4.

Tabla 4 - Parámetros de precisión y *recall* obtenidos.

Personaje	Precisión	Recall
Antony	0.85	0.65
Cleopatra	0.54	0.43
Queen Margaret	0.57	0.76

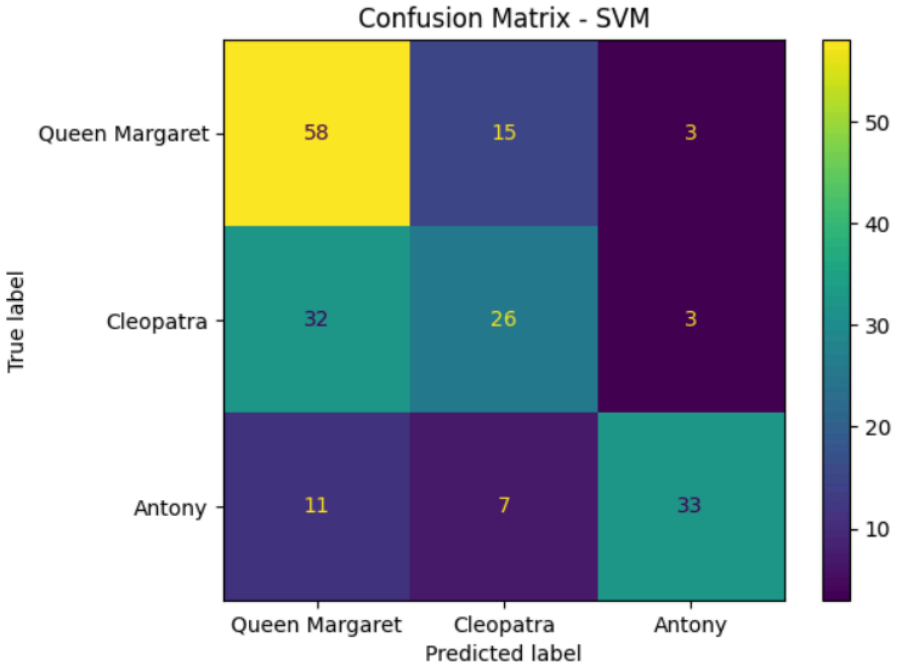


Figura 7 - Matriz de confusión SVM.

Se procede a realizar el análisis cambiando el personaje “Antony” por “Henry V”. Este personaje tiene más participación que “Cleopatra” y “Queen Margaret”. Este desbalance puede llevar a que el modelo tienda a predecir más veces el resultado “Henry V”, dado que esto resultaría en un alto nivel de *accuracy*.

Tabla 5 - Cantidad de párrafos por personaje.

Personaje	Total	Train	Test
Henry V	377	264	113
Cleopatra	204	143	61
Queen Margaret	169	118	51

Al entrenar el modelo y hacer la predicción, se obtiene un *accuracy* de 0.51 pero se observa que el modelo adjudica los párrafos casi exclusivamente a “Henry V”. Esto da como resultado valores malos de *recall* y precisión para cada personaje.

En el caso de “Cleopatra” la precisión es 1 ya que sólo una vez predijo este resultado y fue acertado, pero el recall es de 0.02 ya que falló en predecir 60 párrafos, los que fueron asignados a “Henry V”. Para “Queen Margaret” no realizó predicciones, por lo que tanto el *recall* como la precisión fueron nulos. Por otra parte, para “Henry V” el modelo tuvo una precisión de 0.5 y un *recall* de 1.

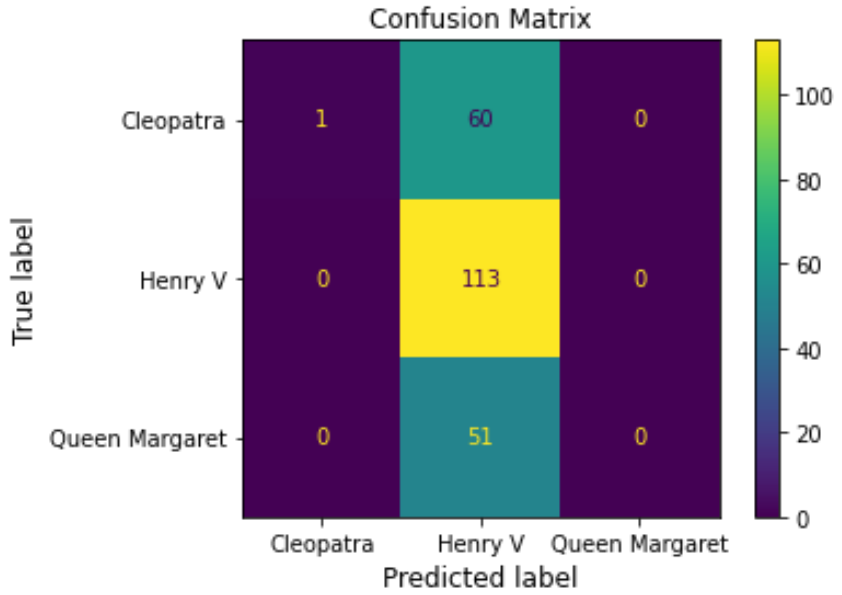


Figura 7 - Matriz de confusión variando un personaje.

Existen técnicas más avanzadas que abordan las limitaciones mencionadas para *bag of words* y TF-IDF. Un ejemplo es el modelo de *word embedding Word2Vec,* una técnica de *deep learning* desarrollada por Google que se basa en generar representaciones vectoriales (o *embeddings*) de palabras. Así, a diferencia de los modelos trabajados en la tarea, que representan palabras de manera independiente y sin contexto, *Word2Vec* crea *embeddings* que capturan relaciones semánticas entre palabras.

Para hacer esto, la técnica utiliza dos arquitecturas principales:

1. *Continuous bag of words:* se predice una palabra dado su contexto (ventana de palabras).
2. *Skip-gram:* se predicen las palabras circundantes (dentro de la ventana definida) dada una palabra central.

Estos modelos aprenden los *embeddings* de las palabras usando una red neuronal simple, que busca predecir los vecinos de una palabra.

Para usar la técnica, se limpian los datos igual que para los modelos trabajados en la tarea, se define una ventana de palabras que determina la extensión del contexto y se entrena la red neuronal para aprender los *embeddings* de las palabras. En *continuous bag of words* la entrada es el contexto y la salida la palabra objetivo, en *skip-gram* es el inverso. La red ajusta los pesos de la capa oculta y estos pesos ajustados son los *embeddings*.

Como esta técnica capta las relaciones semánticas entre las palabras, palabras con significados similares estarán cercanas en el espacio vectorial. Lo que entendemos que sería beneficioso para adjudicar párrafos a personajes ya que facilita la identificación del estilo y vocabulario específico de cada personaje. El hecho de que considere el contexto también puede ayudar a distinguir entre personajes, sobretodo si son de obras diferentes.