Descifrando a Shakespeare

Navadian | Robaina

Introdución a la Ciencia de Datos

Julio 2023

Presentación

En este informe, se examinó la obra de William Shakespeare mediante el análisis de datos, en lugar de abordarla desde una perspectiva literaria. Se utilizó el dataset *The Open Source Shakespeare*, una base de datos relacional y abierta que consta de cuatro tablas. La tabla principal, 'Paragraphs', contiene todos los párrafos de las obras y tiene un total de 35,465 instancias. Cada instancia está asociada a un personaje en un capítulo específico.

La tabla 'Characters' proporciona un desglose de 1,266 personajes e incluye su nombre, una abreviación, una breve descripción y un ID que los relaciona con los párrafos. Los capítulos, que representan las escenas dentro de cada acto, se encuentran en la tabla 'Chapters', la cual incluye una breve descripción, la escena correspondiente y un ID de la obra.

Por último, las 43 obras de Shakespeare se encuentran en la tabla 'Works', donde se proporciona el título, la fecha de publicación y el género al que pertenecen. A continuación se presenta un esquema del dataset:

paragraphs	
id	int
ParagraphNum	int
PlainText	text
character_id	int
chapter_id	int

chapters	
id	int
Act	int
Scene	int
Description	text
work_id	int

characters	
id	int
CharName	varchar
Abbrev	varchar
Description	text

works	
id	int
Title	varchar
LongTitle	text
Date	int
GenreType	varchar

Calidad de los datos

En términos generales, el dataset no presentó datos faltantes, a excepción del campo *description* en la tabla *characters*, lo que se consideró irrelevante debido al rol secundario de dichos personajes. De manera similar, 5 personajes no contaban con el campo abreviatura.

Se investigó acerca del dominio encontrando información acerca de la completitud de la obra de Shakespeare, resultando que la cantidad total de obras que se le adjudican al autor es un debate aún abierto entre los académicos. Fue por esto que si bien no se contaba con *missings* en el resto de tablas del dataset, se puso en tela de juicio la calidad de datos de la tabla Works.

En la figura 1 se analizó la cantidad de párrafos por personaje, habiendo en el dataset un personaje llamado *stage directions*, al cual se le asignaba el 11 % del total de los párrafos. Estos eran acotaciones, los cuales no son considerados diálogos propios de la obra. De manera similar, en segundo lugar se encontraba *Poet*, que su descripción indicaba que era la voz poética de Shakespeare. Se tomó la decisión de no incluir a ambos en el análisis dado que no representaban personajes en escena.

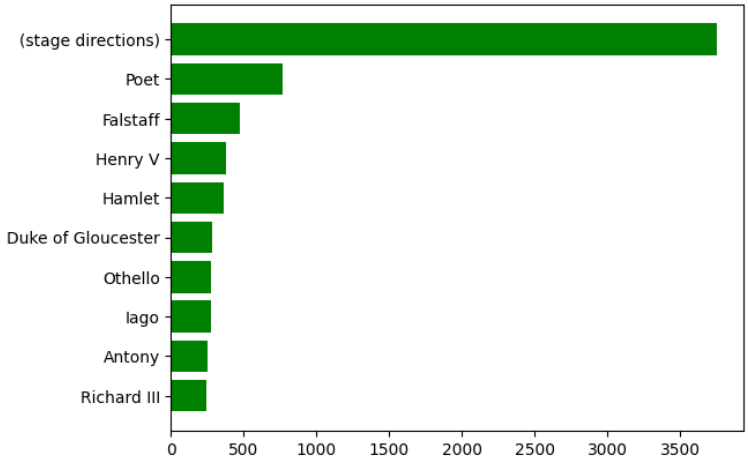


Figura 1: top 10 personajes según cantidad de párrafos.

La obra de Shakespeare

Luego de varias visualizaciones se optó por analizar la obra del autor en el tiempo incluyendo todos los años de carrera, ya que al agrupar en períodos se perdía información que se consideró relevante.

Géneros y publicaciones

En la figura 2, se visualizaron los géneros publicados respecto al total de publicaciones para cada año, destacándose que el autor publicó obras por más de dos décadas y con una amplia variedad de géneros. Se observó un punto de inflexión en el año 1603, el único año de su período de actividad donde las publicaciones se vieron interrumpidas y para el cual sería interesante indagar las causas, que no se abordaron en este trabajo.

Se tomó el año 1603 como punto de referencia, viendo que hay dos tendencias bien marcadas e interpretándose dos distintos "momentos" del autor. En la década y media que antecede dicho año, el autor realizó un número mayor de publicaciones que respecto a la década siguiente, con una mayor intercalación de géneros.

En el primer período predominaron los géneros comedia e historia, intercalados con algunos poemas y tragedias en menor medida. En este período también tuvo su máxima cantidad de publicaciones, con dos comedias, una tragedia y un poema en el año 1594.

Comenzando la década posterior al 1603, publicó mayoritariamente tragedias. No retomó la variedad de géneros hasta el año 1608, año en el que dejó de publicar tragedias y se avocó a publicar en todas las demás clases, incluso publicando por primera vez un soneto[1], sobre el final de su carrera.

Shakespeare en palabras

Uno de los aspectos que resultó de interés en este conjunto de datos fueron las palabras. Para poder llevar a cabo el análisis, antes era necesario normalizar el texto de los párrafos, por lo que se realizaron distintas transformaciones a los mismos. En primer lugar, se quitaron los signos de puntuación y todos caracteres que no eran letras, debido a que estos no forman palabras, y por lo tanto no aportaban información relevante de las mismas. Además, se convirtieron todas las letras mayúsculas a minúsculas, lo que evitó la diferenciación innecesaria de palabras sólo por diferir en un caracter que era la misma letra.

Realizado este filtrado, se consideró que se estaba en condiciones de visualizar las palabras de toda la obra por cantidad, para lo cual se eligió una nube de palabras. En este proceso se interpusieron

[1]Se entiende que esta afirmación puede someterse a la duda, pero nos basaremos en lo que dicen los datos suponiendo que el dataset representa correctamente la realidad del autor.

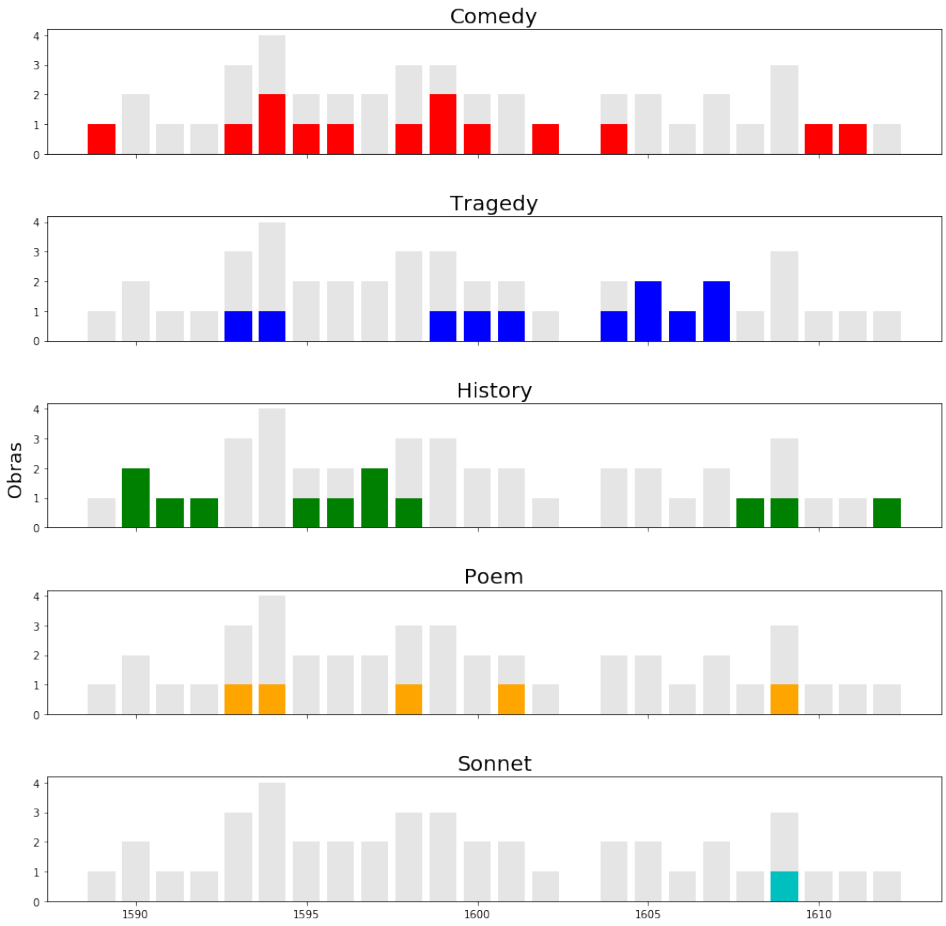


Figura 2: Géneros publicados por año.

las *stop words*, que son el conjunto de palabras usualmente habladas en cualquier idioma, tales como pronombres, conectores, etcétera. Se intentó crear una lista con las mismas pero era un trabajo bastante tedioso e inagotable, por lo que se recurrió a la librería **nltk**, la cual ya contaba con un conjunto bastante amplio de stop words en varios idiomas. Al quitarlas se obtuvieron mejores resultados ya que permitió identificar de manera más clara las palabras más frecuentes. Se visualiza la comparación en las figuras 3 y 4.

Se observó que el "filtrado"de stop words brindado por la librería no fue exhaustivo, y que en la nube de palabras predominaban "thou", "shall", "thy". Se tuvo en cuenta que la obra de Shakespeare fue escrita a fines del siglo 16 y comienzo del 17, y siendo el idioma un campo dinámico, era esperable que la librería no contemplara palabras de un inglés antiguo como el del autor. Una alternativa adicional que no se abordó, hubiera sido crear una lista con las stop words para filtrarlas de forma manual.

Otro aspecto que se tuvo en cuenta, fue que al trabajar con el idioma inglés habrían contracciones. Estas son versiones cortas de dos palabras, que podrían entenderse como "abreviaciones"de dos palabras en una, mediante la omisión de algunas letras. Por ejemplo la contracción de *She will* es *She'll*. En estos casos se podría extender a las dos palabras que compongan la contracción, pero se entendió que no aportaría mayor valor al análisis ya que en última instancia las palabras obtenidas de la contracción serían stop words.

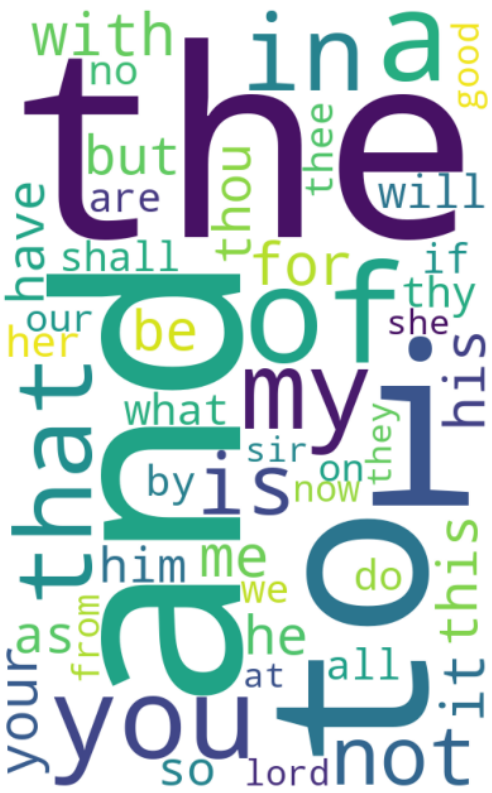


Figura 3: Nube de palabras con *Stopwords*

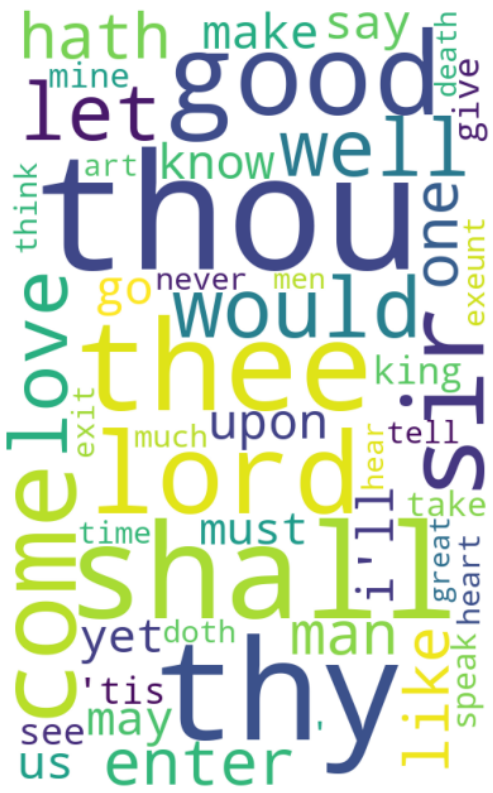
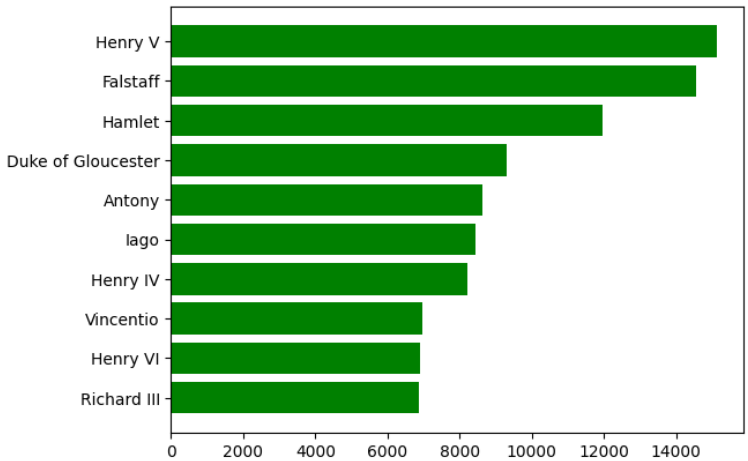


Figura 4: Nube de palabras filtradas

Quiénes dicen las palabras

En la línea del análisis de palabras, resultó de interés la intersección de las mismas con el personaje que las decía. Para ello se utilizó el *merge* entre ambas tablas proporcionado en el código base y se decidió proseguir con el gráfico de barras ya que era apropiado para representar cantidades. Sin embargo, se modificó la orientación de la visualización por barras horizontales, ya que permitían una mejor lectura del personaje en cuestión, además de la eliminación de los personajes *stage directions* y *poet* como se mencionó al principio del informe.

Top 10 personajes con más palabras



Descubriendo a los personajes

Se planteó como objetivo, poder determinar a partir de un párrafo el personaje que lo protagoniza. Este es un problema de aprendizaje supervisado y se utilizarán los datos anteriormente presentados para ajustar dos modelos de clasificación. Para simplificar la tarea, se decidió acotar el dataset y se centró en los *paragraphs* correspondientes a los personajes *Antony, Cleopatra y Queen Margaret*. El promedio de apariciones es de 208,6 por personaje en un total de 6 obras, aunque Antony tiene una cantidad superior de apariciones concentradas en dos obras, seguida por Cleopatra en una, y por último Queen Margaret que, si bien cuenta con más obras, es la que tiene menos párrafos. Estos resultados se resumieron en la siguiente tabla:

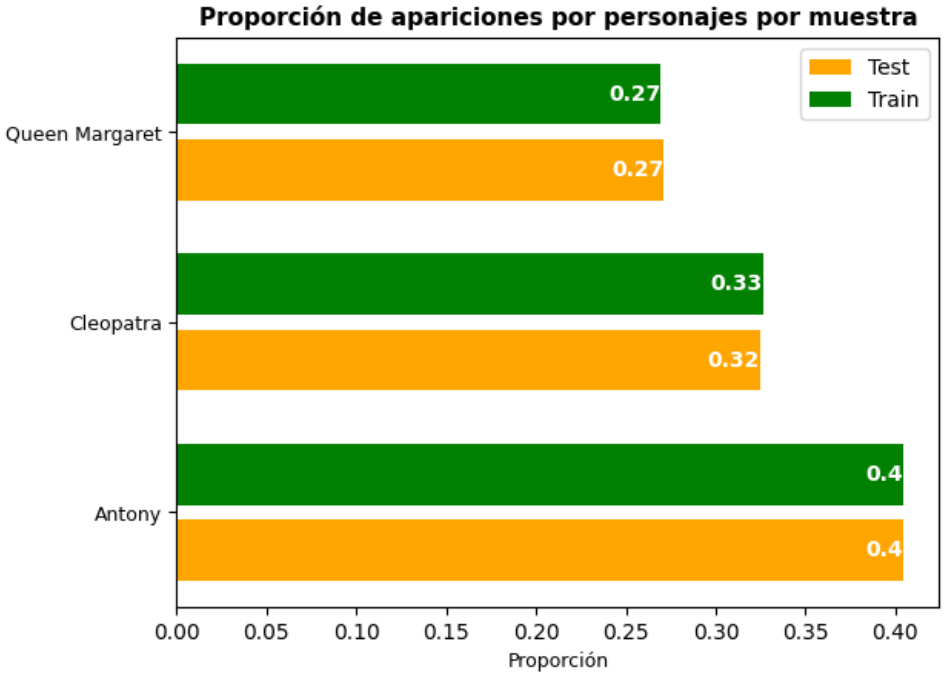
Personaje	**Obra**	**#**
Antony	Antony and Cleopatra	202
	Julius Caesar	51
Cleopatra	Antony and Cleopatra	204
Queen Margaret	Henry VI, Part I	22
	Henry VI, Part II	61
	Henry VI, Part III	53
	Richard III	33

Cuadro 1: apariciones de personajes por obra.

Muestreo estratificado

Dado el desbalance entre las apariciones para los distintos personajes seleccionados se realizó un muestreo estratificado para entrenar con el menor sesgo posible los modelos, dividiendo el total de la muestra en aproximadamente un 70 % para el entrenamiento, y un 30 % como muestra de testeo, obteniendo unos tamaños de muestra de 438 y 188 párrafos respectivamente. Para ello se utilizó el parámetro **stratify** en la función **train_test_split** de la librería **scikit-learn**, que dividió los conjuntos de entrenamiento y testeo respetando las proporciones de las etiquetas a predecir. En el siguiente gráfico se muestran las proporciones de cada personaje en ambos conjuntos.

Observado este balance entre ambas muestras, se continuó trabajando con la muestra de entrenamiento.



Representación numérica

En primer lugar, se procedió a transformar el texto del conjunto de entrenamiento en la representación numérica ***bag of words***, que consta de una transformación matricial $M_{a \times b}$ de los datos[2], con las filas indicando los párrafos y las columnas cada n-grama. Siendo un n-grama una subsecuencia de palabras consecutivas del documento[3], la entrada ij de la matriz indica el recuento de cada n-grama en el párrafo correspondiente. A continuación se ejemplifica cómo quedaría dicha matriz sin stop words:

	auto	**estac.**	**rojo**	**gris**
el auto es rojo	1	0	1	0
el auto en el estacionamiento	1	1	0	0
el estacionamiento gris	0	1	0	1

Al aplicar este procedimiento sobre el texto de los personajes el resultado fue una matriz esparza, es decir que tuvo 0 en la mayoría de sus entradas. Esto era esperable debido a que, incluso los párrafos más extensos en cantidad de palabras utilizaban una proporción ínfima del lenguaje. Quitando las *stop words*, tomando n-gramas de 1 y 2 palabras, y considerando solamente 3 personajes, se obtuvo una matriz de tamaño 469×8791, lo que resultó en un total de 4122979 entradas, de las cuales sólo 12731 eran no nulas[4]. Si se hubiera utilizado todo el dataset se hubiera obtenido una matriz de tamaño aún mayor, ocupando mucho espacio –un recurso finito–. Este método hizo el análisis difícil de escalar a más personajes.

Una transformación que se aplicó fue ***Term Frequency - Inverse Document Frequency***, que es una estadística utilizada en Procesamiento del Lenguaje Natural para evaluar la relevancia de un término dentro de un documento –que en este caso eran los párrafos– o dentro de un cuerpo –correspondiendo a la totalidad de las obras–. En primer lugar, se calcula el *Term Frequency* como:

$$\text{TF} = \frac{\text{número de ocurrencias del término en un documento}}{\text{número total de términos en el documento}}$$

Y por otra parte, el *Inverse Document Frequency* como:

$$\text{IDF} = \log\left(\frac{\text{número total de documentos en el cuerpo}}{\text{número de documentos que contienen el término}}\right)$$

Por último, el TF-IDF es el producto de estos dos: $\text{TF-IDF} = \text{TF} * \text{IDF}$

Si consideramos un término poco relevante como puede ser una palabra común como *food* es probable que aparezca varias veces en un documento, y esto le da un valor grande de TF pero al mismo tiempo también es probable que aparezca en una proporción grande de los documentos haciendo que su IDF sea chico. Palabras que aparecen mucho pero lo hacen en pocos documentos serán especialmente relevantes ya que son específicos, estos tendrán valores de TF-IDF altos.

En este caso los documentos tienen unas pocas palabras, por lo tanto es probable que IDF no agregue demasiada diferencia en los resultados debido a que es raro que una palabra aparezca en muchos párrafos.

Reducción de dimensionalidad

El conjunto de datos modelado se transformó previamente en una matriz de tamaño 469×8791 tal y como se comentó, esto quiere decir que cada párrafo es un punto en un espacio de 8791 dimensiones. La capacidad de graficar y visualizar está limitada a tres dimensiones como máximo, por lo que para lograr una idea de cómo estaban distribuidos los datos en el espacio se debió que recurrir a técnicas de reducción de dimensionalidad.

[2] Donde a es la cantidad de párrafos y b la cantidad de n-gramas.

[3] En la documentación se utiliza el termino documento para referirse a los conjuntos de texto que forman el corpus, en nuestro caso son los párrafos enunciados por cada personaje

[4] Un 0.3 % de todos los elementos.

En la figura 5 se presentó el resultado de aplicar la técnica *Principal Component Analysis* (PCA) sobre el conjunto de datos, que se basó en proyectar los puntos a un subespacio de menor dimensión perdiendo la menor cantidad de información posible. Una vez que los datos tuvieron dimensión 2 fue posible observarlos.

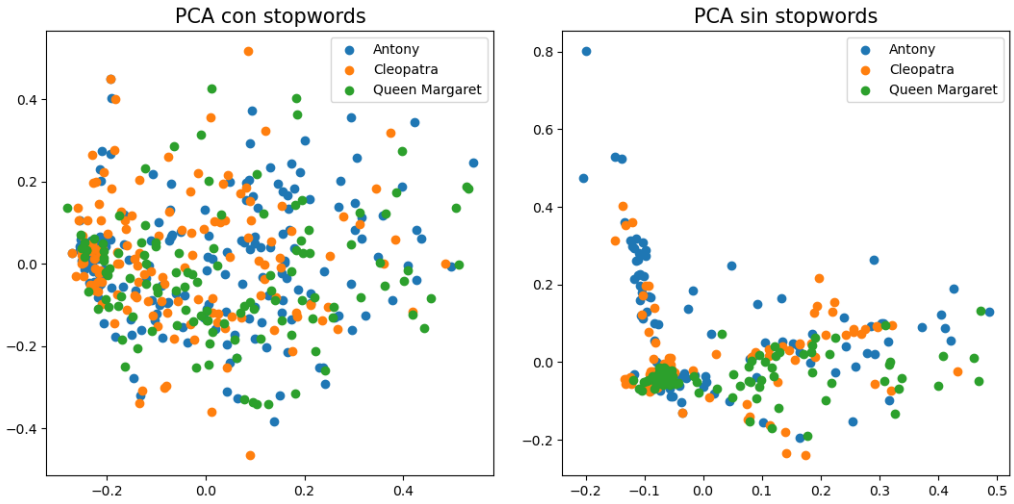


Figura 5: Primeras dos dimensiones del PCA

En la figura 5 se observa dos gráficos con los párrafos que le corresponden a cada personaje. En el gráfico de la izquierda se puede ver el resultado de aplicar PCA a la matriz de *bag of words* considerando n-gramas de una sola palabra, sin filtrar las *stopwords* y aplicando la transformación tomando solo TF (sin IDF). Se observa en estas condiciones cómo los puntos aparecen dispersos y sin una estructura clara, no es posible distinguir regiones donde predomine alguno de los personajes.

En cambio a la derecha tenemos PCA aplicado a los datos filtrando las *stopwords*, tomando n-gramas de una y dos palabras y además aplicando TF-IDF. En estas condiciones es claro que existen ciertas regularidades en los puntos. A pesar de que existe un cúmulo importante de puntos cerca del $(0, 0)$ se puede reconocer una zona vertical dónde los datos de Antony y Cleopatra se encuentran alejados de los datos de Queen Margaret, esto podría deberse a que estos dos tienen una obra en común. Es interesante notar que esto es una propiedad que la representación numérica parecería representar adecuadamente. Podríamos esperar que a la hora de entrenar modelos de clasificación este patrón se repita y obtengamos resultados en los que los datos de Antony y Cleopatra se confunden más entre sí que con los de Queen Margaret.

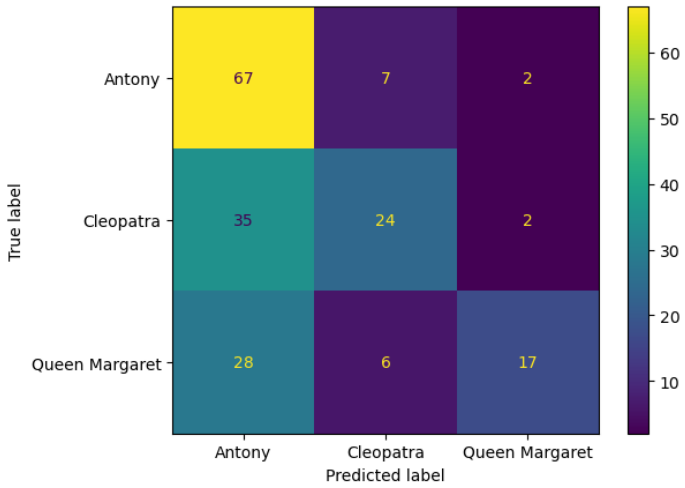
Varianza explicada

Como se mencionó anteriormente el objetivo del PCA es pasar los puntos a un subespacio de menos dimensión a fin de graficar. En este proceso se busca mantener la mayor cantidad de información y para esto la medida que se utiliza es la varianza de los datos. Cuando proyectamos en una componente la varianza de los datos es solamente una proporción de la varianza total, y a medida que agregamos componentes al análisis se obtiene cada vez más varianza y con ella más información. La figura a continuación nos muestra de que forma crece esa proporción hasta las primeras diez componentes. Por las condiciones del problema los datos tienen muchas dimensiones y por lo tanto la varianza explicada con solo dos dimensiones de PCA es escasa, pero de todos modos permite hacerse una idea general de como podrían ser los datos

Modelos de clasificación

Clasificador Multinomial Naive Bayes

Se entrenó con los parámetros por defecto el modelo *Multinomial Naive Bayes* para predecir el personaje según el texto dicho, obteniéndose un *accuracy* del **59 %** sobre la muestra de testeo. Esta métrica da cierta información sobre la calidad del modelo pero es importante considerar otras, ya que lo que indica es la proporción de las etiquetas que acertó el modelo. Esto no termina de definir la validez ya que en los casos en los que tenemos un desbalance en los datos, tener un buen desempeño en la clase mayoritaria se corresponde con un buen resultado y eso no necesariamente ilustra correctamente la calidad del modelo deseado. Por ejemplo predecir todos las etiquetas como la clase mayoritaria puede llegar a dar un buen resultado aunque claramente no es el modelo que se busca. En este caso el desbalance no es tan exagerado pero de todas formas es importante considerar otras métricas. Para tener una idea más holística de los resultados se presenta a continuación la matriz de confusión, comparando las etiquetas predichas por el modelo contra las etiquetas verdaderas.



El modelo predijo en mayor medida al personaje Antony, tanto cuando era lo esperado como cuando no. El resultado es esperable debido a que la cantidad de datos que el modelo tenía sobre Antony era mayor a los otros dos personajes, esto induce un sesgo.

Se incluye una tabla con los valores para cada personaje de precisión y recall. El primero nos indica que tan preciso fue el modelo al usar esa etiqueta y el segundo que tantas de las etiquetas que

se esperaban logró acertar. Se observa que Antony es el que tiene menor desempeño en precisión, solamente un 52 % de las predicciones sobre este personaje fueron acertadas. Esto se observa en la matriz al ver que en la columna de etiquetas predichas como Antony los números son los más altos incluso cuando lo correcto hubiera sido clasificar en alguno de los otros dos. Sin embargo si se observa la métrica de recall este personaje obtiene muy buenos resultados, la matriz tiene pocas ocurrencias de párrafos en los que se esperaba la etiqueta Antony y el modelo predijera otra.

Ocurre lo opuesto con Queen Margaret, la mayoría de sus párrafos se etiquetaron equivocadamente pero las pocas veces que el modelo interpretó que se trataba de un diálogo de este personaje lo hizo de forma precisa

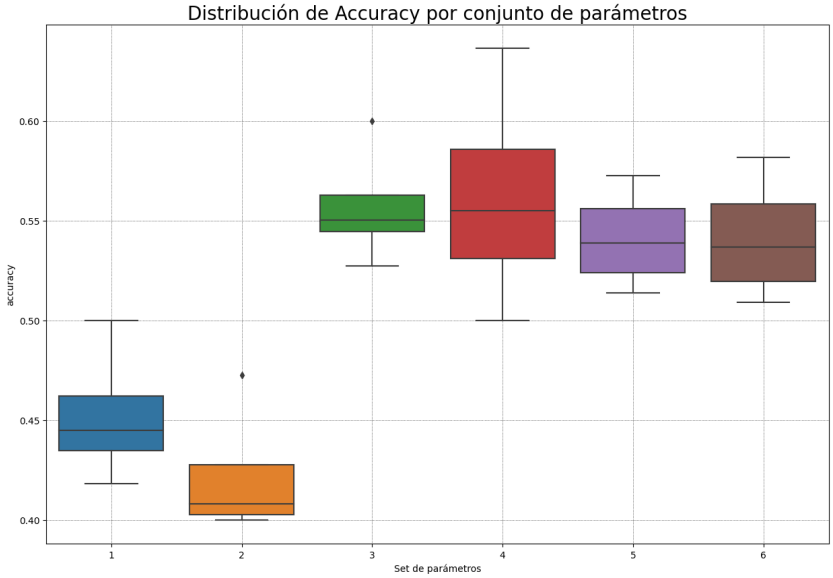
Por último con Cleopatra se observa que sus métricas de precisión y recall se encuentran entre los otros dos personajes.

Personaje	**Precisión**	**Recall**
Antony	0.52	0.88
Cleopatra	0.65	0.39
Queen Margaret	0.81	0.33

Elección de hiperparámetros

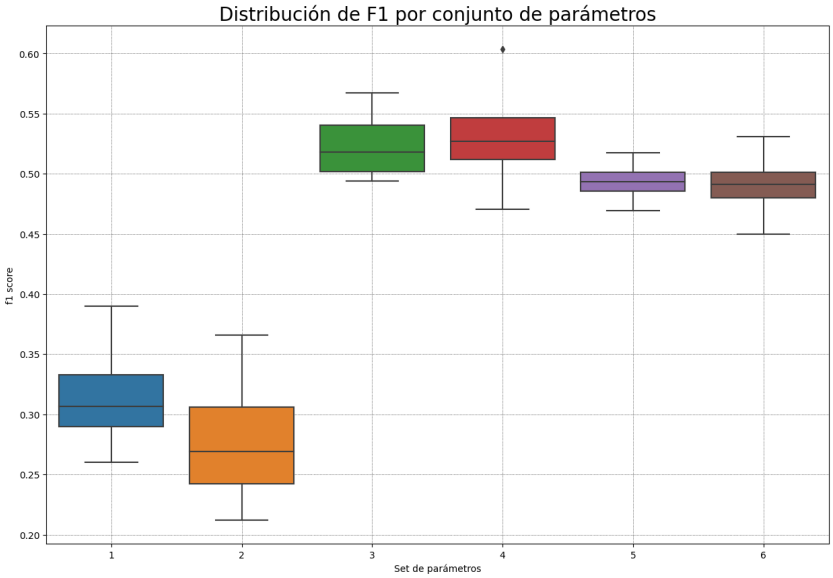
Para la elección de hiperparámetros, se utilizó la técnica de **validación cruzada**. Esta consta en tomar los datos de entrenamiento y separarlos en dos nuevos conjuntos, uno de entrenamiento, que se utilizará para probar los parámetros, y uno de validación que se utilizará para evaluar el rendimiento de estos parámetros. El problema que puede emerger de particionar una vez más el conjunto es que hay menos datos para entrenar el modelo y esto puede traer un rendimiento poco representativo de los parámetros que se buscan evaluar. Por esto lo que se hace para evitar sesgos que puedan aparecer por achicar la muestra es repetir el proceso de entrenamiento con los mismos parámetros cambiando los datos con los que se entrena y valida. Para esto se utilizó la función *StratifiedKFold* de *Sklearn* que lo que hace es partir el conjunto de entrenamiento en cuatro subconjuntos y definir uno de ellos para validar y el resto para entrenar. Se repetirá este proceso cuatro veces para cada set de parámetros que se busque probar. A continuación se incluye una tabla con los parámetros a probar y diagramas de caja con los resultados que se obtuvieron de cada set.

Set	**Stop Words**	**N-gram**	**IDF**
1	None	(1,2)	True
2	None	(1,1)	False
3	English	(1,1)	False
4	English	(1,1)	True
5	English	(1,2)	False
6	English	(1,2)	True



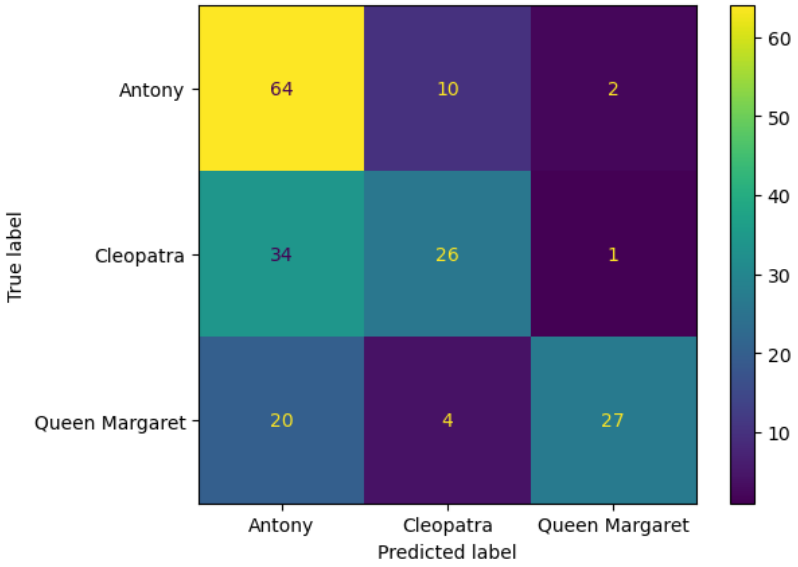
Como primera observación, se notó la clara diferencia entre mantener o excluir las stop words, arrojando la segunda decisión un mejor desempeño midiendo por accuracy. Por otra parte, de los sets de parámetros que excluían las stop words, los que mostraron desempeño levemente mejor fueron los que tuvieron n-grama (1, 1), sin mostrar diferencias relevantes en cuanto al IDF.

Se complementó la elección de hiperparámetros con el resultado del StratifiedKFold con la métrica F1[5] y se llegó a la conclusión de que el set de parámetros 4 era el que tendría el mejor desempeño.



Se incluye el resultado de la matriz de confusión usando los parámetros para los que se entendió tendrían el mejor desempeño, que eran no incluir las stop words, utilizar n-gramas (1,1) y realizar la transformación IDF. Estos últimos consiguieron una accuracy del **62 %**

[5]Esta métrica busca sintetizar los resultados de precisión y recall



Personaje	Precisión	Recall
Antony	0.53	0.89
Cleopatra	0.69	0.41
Queen Margaret	0.91	0.41

Como se pudo observar, se dieron ciertas mejorías respecto a los parámetros utilizados inicialmente, sin embargo los resultados aún no son los más precisos. Si bien se mejoró un poco en la predicción de Queen Margaret, persistió la confusión de Cleopatra por Antony, lo que a priori se adjudicó a la mayor cantidad de párrafos que tuvo este personaje en el entrenamiento del modelo.

Entrenar el modelo en base a *bag of words* parece mostrar limitaciones claras. En primer lugar la limitación de la cantidad de personajes que permite incluir. Y en segundo lo poco preciso de los resultados, resulta claro que el método no termina de capturar algunas de las características principales del texto y esto imposibilita al modelo de clasificación tener buenas métricas. Cabe la posibilidad que el problema tenga una dificultad intrínseca, y por ese motivo se entrenará un clasificador distinto para luego comparar resultados.

Clasificador Support Vector Machine

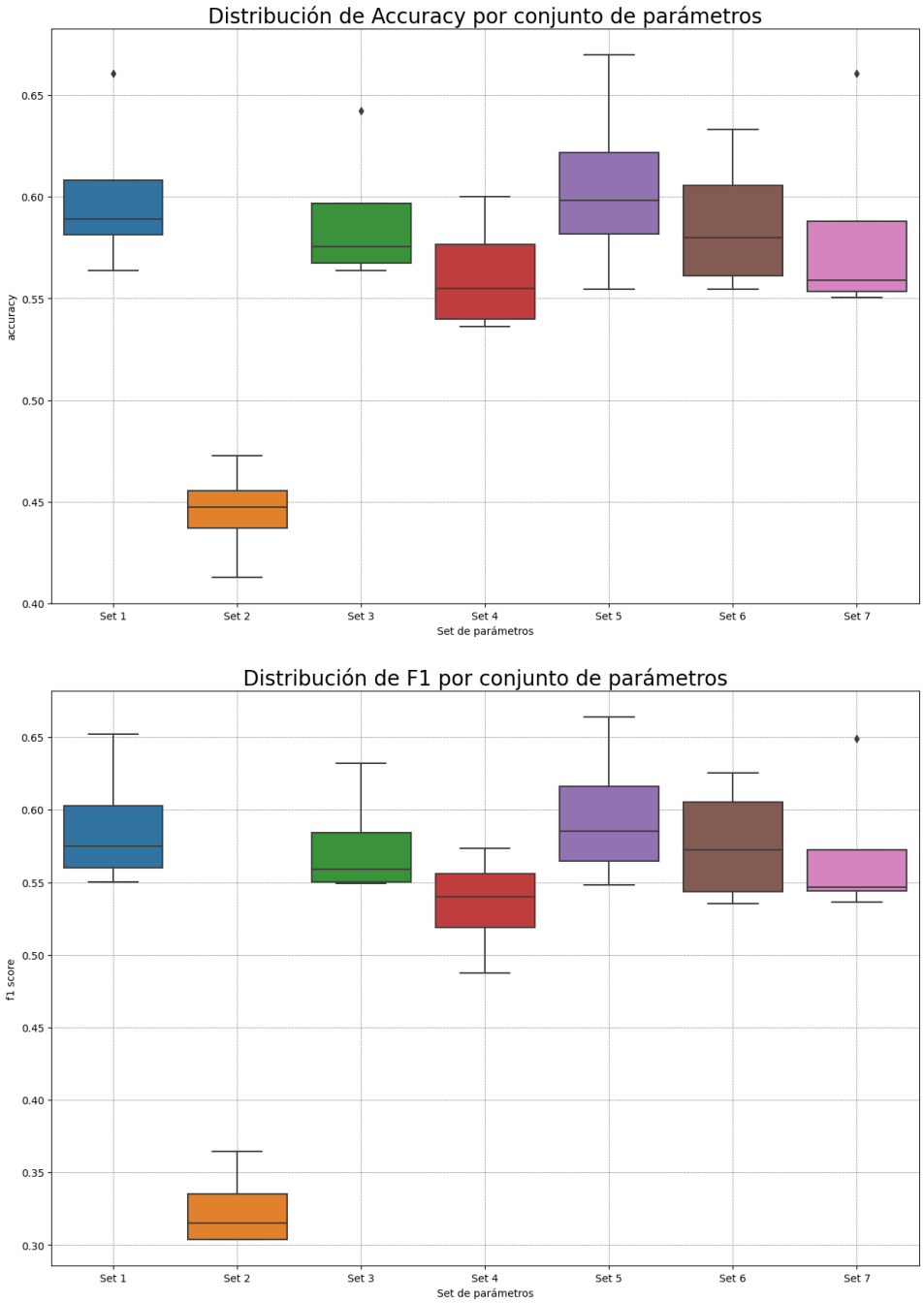
Support Vector Machine es un método de clasificación que se basa en encontrar en el espacio de datos los hiperplanos que separan las categorías a las que deben clasificarse. Se puede pensar que estos hiperplanos hacen de soporte a cada categoría, de aquí el nombre.

El problema que existe es que por lo general los datos no son linealmente separables, por eso, para hacer el clasificador más efectivo se utiliza una función kernel que mapea los datos a una dimensión mucho mayor (esperando que ahí sí sean separables por un hiperplano) y a través de las operaciones con estas funciones kernel podemos obtener los hiperplanos correspondientes.

Se repitió el procedimiento de validación cruzada buscando lo mejores hiperparámetros para este clasificador, a continuación aparecen los resultados obtenidos para cada set.

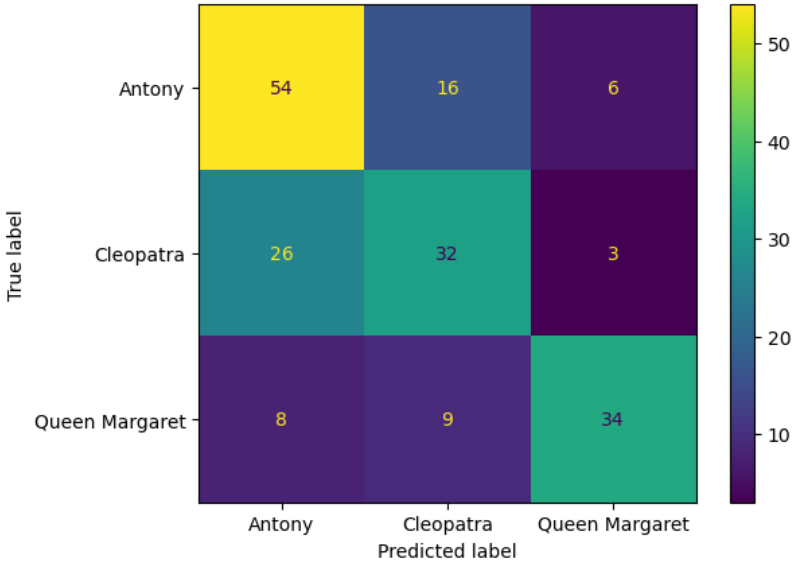
Set	Stop Words	N-gram	IDF	Kernel
1	English	(1,1)	False	Linear
2	English	(1,1)	False	Poly
3	English	(1,1)	False	Sigmoid
4	English	(1,1)	False	RBF
5	English	(1,1)	True	Linear
6	English	(1,2)	False	Linear
7	English	(1,2)	True	Linear

Se visualizó los boxplot de las métricas Accuracy y F1 para los distintos sets de parámetros, descartando en primer lugar el set 2 con el Kernel Polinomial. Se procedió directamente a observar los sets con mejor desempeño en ambas métricas, notando que se destacaban el set 1 y 5, ambos con Kernel lineal, con n-grama (1,1) y sin stop words, con la diferencia de que el quinto aplicaba la transformación IDF, siendo éste el finalmente electo.



Se entrenó el modelo SVM con el mejor set de parámetros (5) y los resultados fueron bastante más satisfactorios que en el modelo MNB. Se logró un valor de accuracy de **64 %**. Si se observan las métricas de precisión y recall incluidas en la tabla se observa un pequeño descenso en los valores de precisión pero compensados con una interesante mejoría en la métrica de recall. Solamente observando la matriz se puede apreciar el cambio viendo que la diagonal –que muestra las etiquetas que se predijeron correctamente– se puede apreciar que tiene color más claro que el resto de la matriz denotando el mejor desempeño. De todos modos el modelo tuvo dificultades a la hora de

predecir a Cleopatra, confundiéndola casi el 43 % de las veces con Antony.



Personaje	**Precisión**	**Recall**
Antony	0.61	0.71
Cleopatra	0.56	0.52
Queen Margaret	0.79	0.67

Cambio de personaje

Se evaluó el problema cambiando el personaje de Antony por el conocido Hamlet, buscando que fuera también del género masculino para observar si hay algún indicio de similitud en los resultados.

Personaje	**Obra**	#
Cleopatra	Antony and Cleopatra	204
Hamlet	Hamlet	358
Queen Margaret	Henry VI, Part I	22
	Henry VI, Part II	61
	Henry VI, Part III	53
	Richard III	33

Observando la tabla presentada vemos que el personaje introducido abarca casi el 50 % de los datos de entrenamiento. Esto genera que la métrica de accuracy sea aún menos representativa, ya que si se definiera un modelo que etiquetara todos los párrafos que recibe como Hamlet entonces se tendría un modelo con un desempeño no tanto peor que el primero que introduce el informe, con la diferencia que este modelo ficticio no tiene ninguna función más que responder Hamlet. Incluso entrenando cualquiera de los modelos anteriores es probable que los resultados de la matriz muestren un sesgo muy marcado hacia el personaje mayoritario.

Para mitigar este problema puede ser útil considerar técnicas de sobremuestreo o submuestreo (o ambas). Estos métodos buscan balancear las proporciones de los datos, uno con la premisa de quitar datos de la clase minoritaria y la otra aumentando la cantidad de datos de la o las clases minoritarias, ya sea dupicando o agregando versiones ligeramente modificadas. En este caso sería útil tomar solo una porción de los párrafos de Hamlet o repetir algunos párrafos de Queen Margaret o Cleopatra para incentivar al modelo a predecir otras etiquetas.

Feature extraction

En este trabajo se entrenaron los modelos de clasificación en base a los features del texto obtenidos por la técnica *bag of words*. Pero también existen otras técnicas, una de las más efectivas es convertir los tokens a embeddings, vectores típicamente de unas 150 dimensiones. Existen varios métodos para lograr este objetivo, una de los más populares es *Word2vec*. Esta herramienta logra representaciones vectoriales de los tokens entrenando una red neuronal para que, en base a un corpus de texto, prediga una palabra a partir de su contexto. Para lograr esto la red genera una representación vectorial de cada palabra, optimizada de modo que las palabras que aparecen en contextos similares tengan representaciones similares.

En cuanto a la tarea de predecir personajes a partir de un párrafo, si se intenta entrenar el embedding desde cero utilizando solo los datos de texto que se incluyen en esta tarea, es probable que no se obtengan buenos resultados. Esto se debe a que el entrenamiento de redes neuronales generalmente requiere una cantidad mayor de datos que la proporcionada por este conjunto si nos limitamos a tres personajes. Sin embargo, este método permitiría escalar el modelo para predecir una mayor cantidad de personajes, ya que cada token tiene una representación de longitud fija. Por lo tanto, incluir más párrafos en los datos de entrenamiento y prueba no presentaría las dificultades mencionadas con el enfoque de '*bag of words*'

De todos modos el entrenamiento de una red neuronal requiere una capacidad importante de computo por la cantidad de parámetros que se requieren aprender. Pero si se tuvieran preentrenadas las representaciones sería esperable que capturen mejor la información de los párrafos que el método que se utilizó en este informe, y por lo tanto podríamos esperar que se logren mejores resultados.

Conclusiones finales

Como comentarios finales, resultó interesante remarcar la mayor cantidad de apariciones de los personajes masculinos con respecto a los femeninos. Se vió que al contar con una mayor cantidad de párrafos en la obra de Shakespeare, se pueden generar problemas a la hora de entrenar modelos, ya que sesga las predicciones en favor de estos personajes más representados.

Puede resultar de interés para futuros trabajos realizar un etiquetado del género de los personajes, para determinar si definitivamente el género masculino se lleva más apariciones que el género femenino en las obras. A su vez, se podría entrenar un modelo en base a otro feature extraction que permita llevar este análisis de clasificación a toda la obra del autor, dadas las limitaciones analizadas con respecto a la capacidad de cómputo, y ver en este sentido qué tanto se confunde el genero femenino con el masculino en la totalidad de sus trabajos, o si ello fue un resultado para nuestros 3 personajes seleccionados.