Tarea 2:
Introducción a la Ciencia de Datos

Julio 2023

Integrantes:

Federico Matonte (3.832.168-8)

Luciana Vidal (5.147.188-8)

Introducción

El objetivo del presente documento es conectarse a una base de datos relacional y analizar los datos de la misma, realizando la limpieza correspondiente en caso de ser necesario.

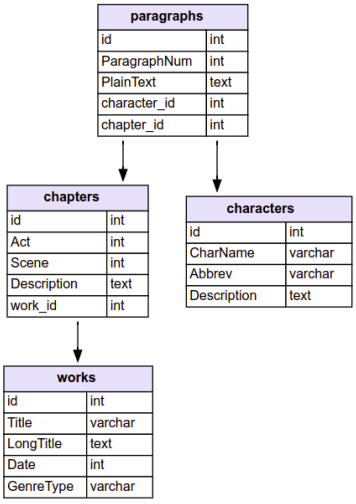
La base de datos contiene las obras de William Shakespeare [1], por lo que el objetivo es utilizar los conocimientos adquiridos en el curso de Introducción a la ciencia de datos para analizar dichos datos. Para esto, se trabajó con dos notebooks, uno para cada tarea, subidos en el repositorio público de GitHub llamado introcd_grupo_17 [5].

El documento se organiza de la siguiente forma: en la Sección 1 se explica el procesamiento de datos, detallando la carga y calidad de los mismos, y en la Sección 2 el proceso de análisis de datos: estudiando el personaje con más párrafos, visualizando la obra a lo largo del tiempo, trabajando sobre el conteo de palabras en distintos escenarios, y generando una lista de interrogantes que sería interesante analizar en un trabajo a futuro. En la Sección 3 se entrenan distintos modelos evaluando la performance de cada uno para un subconjunto de datos.

1. Procesamiento de datos

La base de datos a analizar contiene todas las obras del dramaturgo William Shakespeare [1], está formada por cuatro tablas relacionadas entre sí que contienen la información de las obras, el texto de cada una de ellas y los personajes. En la Figura 1 se muestra el esquema de dicha base de datos.

Figura 1. Diagrama de la base de datos de las obras de William Shakespeare [1].



La tabla de párrafos (**`paragraphs`**) contiene el texto de cada párrafo de las obras, indica el personaje que lo interpreta y el capítulo al que pertenece dicho pasaje. Un párrafo pertenece a un capítulo y es interpretado por un único personaje.

La tabla de capítulos (**`chapters`**) contiene la información de cada capítulo: descripción que indica en su mayoría al escenario, el número de acto al que pertenece y el número de escena, así como la referencia a la obra a la que pertenece. Por lo que la tabla posee una jerarquía de pertenencia de tres niveles: empezando por la obra, siguiendo por el acto y finalizando por la escena. Un capítulo pertenece a una única obra y puede estar relacionado a varios párrafos.

La tabla de obras (**`works`**) contiene la información sobre la obra: título, título largo, fecha y género literario. Una obra tiene un único género y puede tener varios capítulos.

Finalmente, la tabla de personajes (**`characters`**) contiene la información sobre el personaje: un nombre, una abreviación y una descripción. Un personaje puede interpretar varios párrafos, por lo que puede estar en varios capítulos y obras.

1.1. Carga de datos

Siguiente el ejemplo provisto por los docentes se utiliza la biblioteca sqlalchemy para crear una conexión a la base de datos Shakespeare [1], alojada en relational.fit.cvut.cz con la que se creará un motor de Base de datos, que resolverá la consulta sql que traerá las tablas para ser almacenadas en variables a utilizar como dataframes.

1.2. Calidad de datos

Para analizar la calidad de los datos se revisó si las tablas tenían datos faltantes. En la tabla de personajes se descubrió que más de la mitad de los personajes no tienen una descripción (51.029% tienen una descripción vacía). Por otro lado, cinco personajes no tienen abreviatura (0.395).

2. Análisis de datos

En la presente sección se realiza un análisis sobre los datos de las obras de Shakespeare, en donde se estudian los personajes con mayor cantidad de párrafos y se realiza un conteo de palabras.

En el Anexo 1. Preguntas interesantes, se presenta una lista de preguntas con posibles interrogantes para extender el análisis de datos. Por otro lado, el análisis de las obras a lo largo de los años se puede encontrar en el Anexo 2. Obras a lo largo de los años.

2.1. Personaje con más párrafos

Se utiliza como línea base una versión recortada de A LIST of the 20 longest Shakespearean roles incluida en el libro The Book of Lists 2 (1980) obtenida en la pregunta de Quora “Which character has the most lines in a Shakespeare play?” [2].

Tabla 1. Top 10 de los personajes según sus intervenciones.

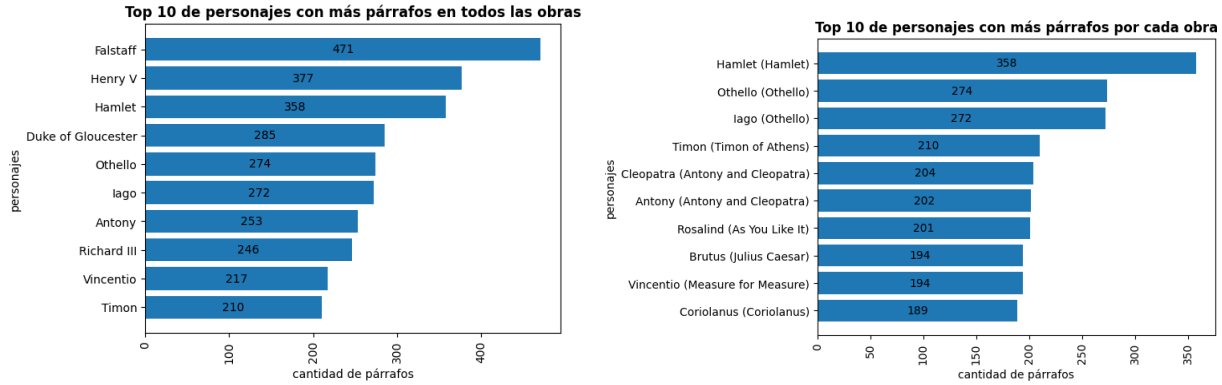
Posición	Personaje	Cantidad de líneas	Obra
1	Hamlet	1,422	Hamlet
2	Falstaff	1,178	Henry IV (parts one and two)
3	Richard III	1,124	Richard III
4	Iago	1,097	Othello
5	Henry V	1,025	Henry V
6	Othello	860	Othello
7	Vincentio	820	Measure for Measure
8	Coriolanus	809	Coriolanus
9	Timon	795	Timon of Athens
10	Antony	766	Antony and Cleopatra.

La pregunta no solo brinda una línea base sino también plantea que algunos de los personajes aparecen en más de una obra.

Se preparan dos listas de los 10 personajes con mayor cantidad de párrafos, considerando en la primera que los personajes puede aparecer en más de una obra (y sus nombre puede identificarlos) y en la segunda que los personajes son locales a las obras (y es necesario tanto sus nombres como los de las obras para identificarlos).

Si consideramos que los personajes son los mismos en distintas obras Falstaff es el personaje con más párrafos, lo que es consistente con el hecho de que aparece en tres obras distintas presentes en el juego de datos: Henry VI Part I, Henry VI Part II y Merry Wives of Windsor. En cambio si diferenciamos los personajes por obras, Hamlet es el personaje con más párrafos en el juego de datos, lo que es consistente con el hecho de que es el personaje principal de la obra más extensa del dramaturgo.

Figura 2. Listas de personajes con más párrafos. A la derecha se hace distinción de los personajes por obra, y a la izquierda no.



2.2. Conteo de palabras

2.2.1. Normalización del texto

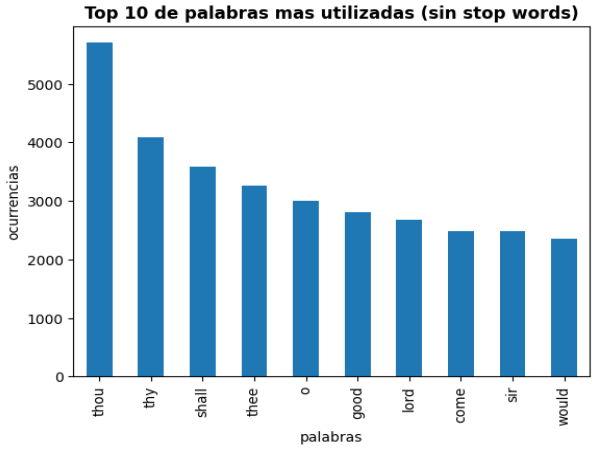
Antes de proceder a realizar el conteo de palabras, se debió normalizar el texto, por ejemplo para que una palabra en mayúscula representa lo mismo que en minúscula. Sobre el mismo se hicieron distintas transformaciones:

- Conversión de letras a minúscula.
- Eliminar los indicadores de escena, y aquellas frases que estén dentro de las misma dado que es una descripción utilizada para explicarle al lector sobre el contexto en el que se desarrolla la escena.
- Reemplazar los signos de puntuación, por espacios: ("\n", ",", ";", ".", "?", "!", ":", "--", "\"", "(", ")","&c" y "[","]").
- Expandir las palabras contraídas con una pequeña lista de ejemplos.
- Eliminamos las contracciones no presentes en la línea y su apóstrofe.
- Eliminar las palabras denominadas stop words dado que no aportan un significado relevante, para así enfocarse en las palabras que son importantes.

2.2.2. Conteo de palabras

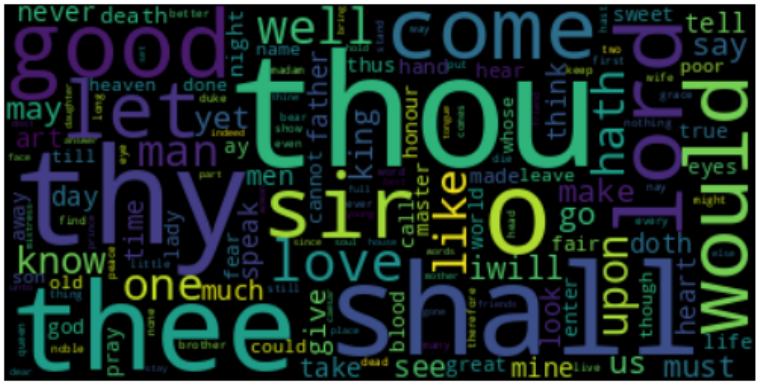
Con el fin de analizar cuáles son las palabras más utilizadas en las obras, primero se graficaron las diez más mencionadas y se constató que son las denominadas stopwords (no aportan significado por sí mismas). Por lo que se procedió a eliminar las stopwords presentes en las listas de NLTK [3] y se graficó en la Figura 3 el top 10. Finalmente, con el fin de mejorar la visualización en la Figura 4 se graficó una nube de palabras (en la que el tamaño de la palabra representa la cantidad de ocurrencias).

Figura 3. Top 10 de las palabras más utilizadas en las obras, eliminando las stopwords.



Buscando encontrar diferencias entre géneros o personajes, se podría utilizar una gráfica de barras apiladas con las palabras en el eje de las x, la cantidad de ocurrencias de una palabra en el eje de las y, usando los géneros o personajes para separar la cantidad de personajes (colores en las barras).

Figura 4. Nube de palabras con las más utilizadas en las obras, eliminando las stop words.



Una opción más refinada sería establecer que cada sujeto (personaje o género) es representado como un vector en n dimensiones, con n la cantidad de palabras, pero al no poder graficar en tantas dimensiones se utilizará un algoritmo de reducción de dimensiones (por ejemplo PCR) para así convertir esta presentación en un vector que sí pueda ser graficado: en dos dimensiones sería graficarlos como puntos; o en cuatro dimensiones utilizar dos coordenadas, una para el radio (haciendo que un círculo sustituya al punto) y una cuarta dimensión que represente un color en una paleta de colores.

2.2.3. Personajes con mayor participación

Al analizar los personajes con mayor cantidad de palabras, se pudo constatar que hay párrafos dichos por un personaje llamado “Poet” que hace referencia a la representación del yo lírico y por lo tanto no es un personaje. Por otro lado, también se encontró indicaciones de dirección (personaje: “(stage directions)”), que nuevamente son palabras que no corresponden a un personaje.

Finalmente, eliminando al yo lírico y a las indicaciones, se observó que hay personajes que tienen diálogos en más de una obra por lo que su participación es mayor aunque no necesariamente lo es dentro de cada obra. En la Figura 5 se graficaron los 10 personajes con más palabras inter obra y en la Figura 6 los de intra obra.

Analizando ambas gráficas, se constata que si bien un personaje puede tener mayor cantidad de palabras a los largo de todas las obras, esto no asegura que su participación intra obra también sea mayor, este es el caso de Henry V que si bien en el análisis inter obra está primero, si se lo analiza a lo largo de las obras en las que participa no está en la misma posición. El personaje aparece en tres obras, pero únicamente en una está como personaje con mayor cantidad de palabras, esto se puede observar en la Tabla 2.

Figura 5. Top 10 de los personajes con mayor cantidad de palabras inter obra.

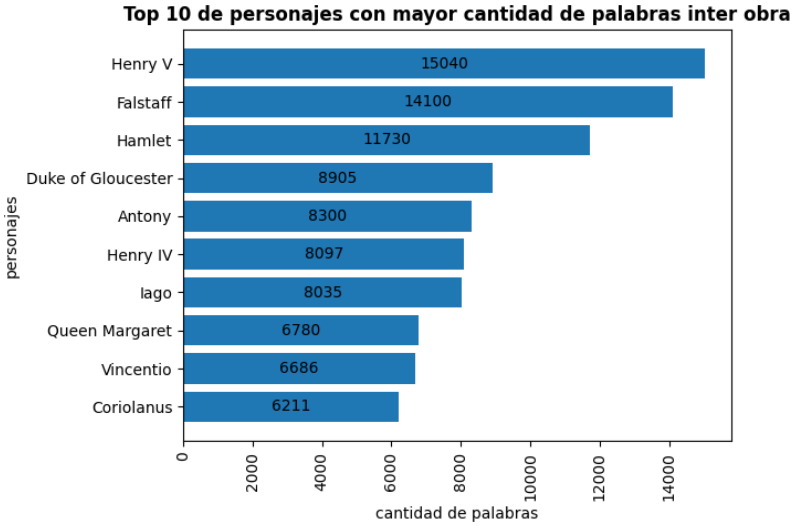


Figura 6. Top 10 de los personajes con mayor cantidad de palabras intra obra.

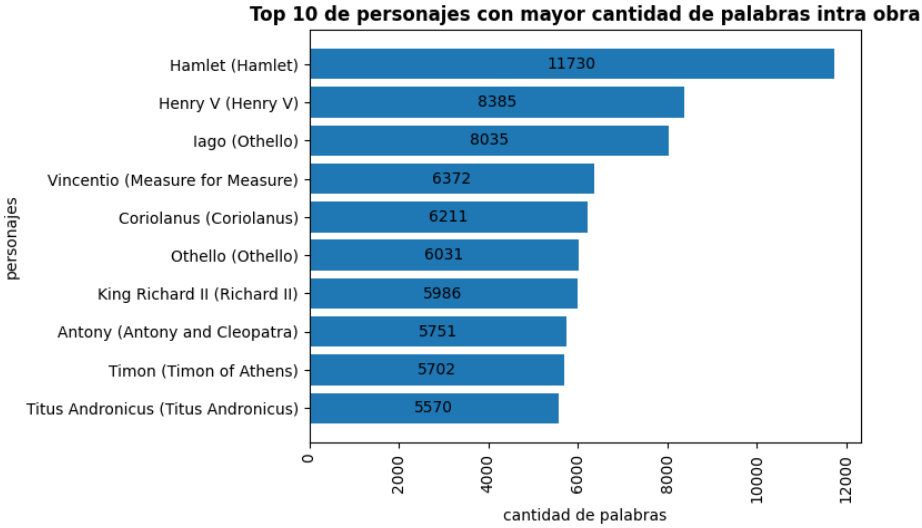


Tabla 2. Ranking de las apariciones de Henry V en sus obras.

Obra	Ranking
Henry IV, Part I	2
Henry IV, Part II	3
Henry V	1

3. Modelo de texto

Para la segunda tarea se trabajó con un subconjunto de los datos, armando un corpus en el que solo se contemplan los párrafos de los personajes: Antony, Cleopatra y Queen Margaret, y su relación con dichos personajes. Considerando los párrafos como el dato (x) relacionado con la categoría personajes (y).

3.1. Separación del corpus

Partiendo del corpus, y normalizando los textos siguiendo los pasos descritos en la Sección 2.3.1., se utiliza la función `train_test_split` para realizar una separación del mismo entre un conjunto de entrenamiento (30%) y uno de test (70%). Sabiendo que los tres personajes tienen una cantidad de párrafos desproporcionada, y siguiendo la sugerencia de la tarea, se utiliza el parámetro `stratify` para que las particiones mantengan las proporciones de párrafos para cada personaje del corpus original. Corroborando esto con una función de conteo y visualizándolo mediante un gráfico treemap (Anexo 3. Proporciones en los corpus).

3.2. Cambio de representación

Contando hasta ahora con un array compuesto de la versión normalizada de cada párrafo, se utiliza la función `CountVectorizer` para crear una estructura de datos con la que representarlos. Esta nueva representación utiliza una matriz en la que a cada párrafo se le asigna una fila y a cada nueva palabra presente en los párrafos se le otorga una columna. En esta matriz la celda `[i,j]` representa cuán "presente" (según un criterio definible) se encuentra la palabra `j` en el párrafo `i`.

Un ejemplo de esta representación, donde la presencia se define como cantidad de ocurrencias de la palabra, puede generarse con los siguientes párrafos:
`p1 = "hola a todos"` y `p2 = "hola amigos"`.

Entre los dos párrafos se genera un diccionario como el conjunto de palabras: `(hola, a, todos, amigos)`. Produciendo una matriz convencional de dimensión (2, 4) por contar con 2 párrafos y 4 palabras distintas, en una matriz convencional se representaría como se muestra en la Tabla 3.

Tabla 3. Ejemplo de representación de las palabras.

Párrafos \ Palabras	hola	a	todos	amigos
p1	1	1	1	0
p2	1	0	0	1

Este tipo de representación, llamado **conteo de palabras o bag of words** tiende a generar matrices dispersas ya que no toda palabra del diccionario se encuentra presente en cada párrafo. Esto es aprovechado al momento de implementar la estructura de datos que almacenará la matriz, utilizando un vector en el que se almacena sólo el valor y los índices de las celdas distintas a cero. Siguiendo el ejemplo anterior la estructura de conteo de palabras se implementa como un vector: `[(1,1,1), (1,2,1), (1,3,1), (2,1,1), (2,4,1)]` que corresponde a las ocurrencias de palabra párrafo: `(p1, hola)`, `(p1, a)`, `(p1, todos)`, `(p2, hola)`, y `(p2,amigos)`.

Esta implementación es preferible ya que al agregar un nuevo párrafo sólo requiere que se agreguen nuevos elementos al vector ocurrencias, mientras que en una representación de matriz convencional requeriría no solo agregar una fila para el nuevo párrafo, sino también una nueva columna para cada una de las nuevas palabras que aporte al diccionario (con ceros en todas las filas salvo la del nuevo párrafo).

La representación puede ser mejorada no solo tomando las palabras, sino también conceptos compuestas de varias palabras. Para esto se utiliza el concepto de **n-grama**,

donde los uno-gramas permiten almacenar las palabras, los bi-gramas permiten almacenar 2 palabras continuas, pudiendo generalizarse esta idea para n-gramas en que se almacenan n palabras continuas. Gracias a esto se puede reconocer conceptos que no pueden ser definidos en una sola palabra, donde antes podíamos representar los conceptos *pájaro* y *azul* por separado, con los bi-gramas podemos reconocer el concepto de *pájaro azul*.

Además, la matriz de conteo de palabras puede ser mejorada ignorando los stopwords.

Otra forma de mejorar la representación anterior es, en vez de utilizar la medida cantidad de ocurrencias, calcular la importancia de la palabra en un párrafo y en el corpus. Para esto, la representación **Term Frequency - Inverse Document Frequency (TF-IDF)** se basa en la frecuencia del término (palabra) y la frecuencia inversa del documento (párrafos):

- Frecuencia del término (TF): mide la importancia de un término en un documento específico. Por lo que se calcula como la frecuencia con la que aparece un término en el documento. La fórmula para calcular el valor TF de un término t en un documento d se muestra en la Ecuación 1.
- Frecuencia inversa del documento (IDF): mide la importancia del término en los documentos. Por lo tanto, pondera aquellos términos que son más raros en el corpus. La fórmula para calcular el valor IDF de un término t en un documento d se muestra en la Ecuación 2.

Entonces, la matriz TF-IDF utiliza las dos medidas antes definidas para hallar el peso que indica la importancia relativa de un término en el documento y en el corpus. Para esto multiplica el valor TF de un término en un documento por su valor IDF.

Ecuación 1. Fórmula para calcular el valor TF para el término t en el documento d.

$$TF(t,\ d)\ =\ \frac{Cantidad\ de\ veces\ que\ aparece\ t\ en\ d}{cantidad\ de\ términos\ en\ d}$$

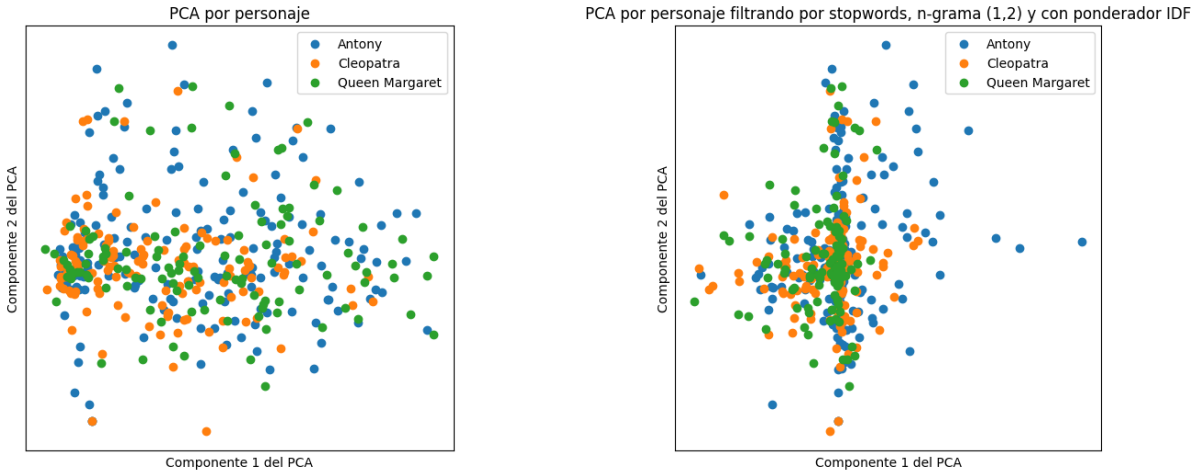
Ecuación 2. Fórmula para calcular el valor IDF para el término t en el documento d.

$$IDF(t,\ d)\ =\ log\left(\frac{Cantidad\ de\ documentos\ en\ el\ corpus}{cantidad\ de\ documentos\ en\ los\ que\ aparece\ t}\right)$$

Utilizando la medidas TF por un lado y la medida IDF y el filtrado de stopwords se crean dos nuevos modelos. Utilizando el algoritmo PCA (Principal Component Analysis) para reducir los vectores a solo 2 dimensiones con las que podrán ser visualizadas en una nube de puntos en la que se colorea según su personaje.

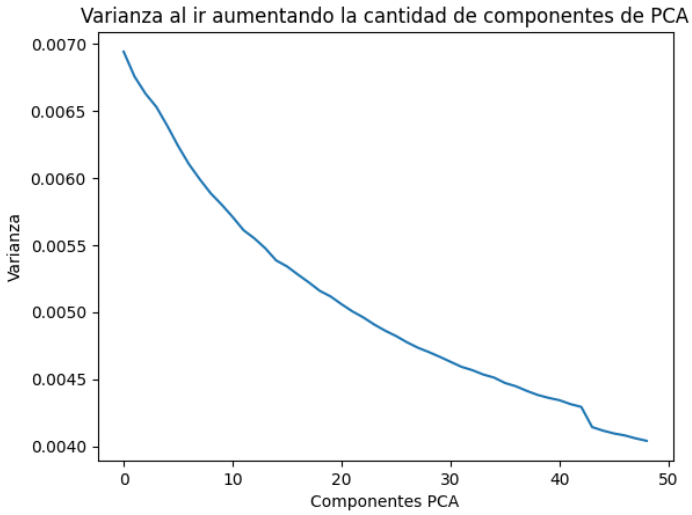
En la Figura 7 se grafican los PCA por personaje para dos componentes y lo mismo pero filtrando los stopwords, con bi-grama y usando el ponderador IDF. En ninguna de las visualizaciones se encuentran patrones que permitan separar a los personajes por su ubicación en el espacio de dos dimensiones.

Figura 7. Gráficas de PCA. A la izquierda PCA por personaje para dos componentes, y la derecha lo mismo pero filtrando por stopwords, con bi-grama y utilizando el ponderador IDF.



Se toma el modelo TF y se ejecuta el algoritmo PCA modificándolo de 1 a 50 dimensiones. En la Figura 8 se puede notar que ante un incremento en la cantidad de componentes del PCA, la varianza se achica, por lo que no se pueden descartar.

Figura 8. Variación de la varianza al aumentar la cantidad de componentes de PCA.



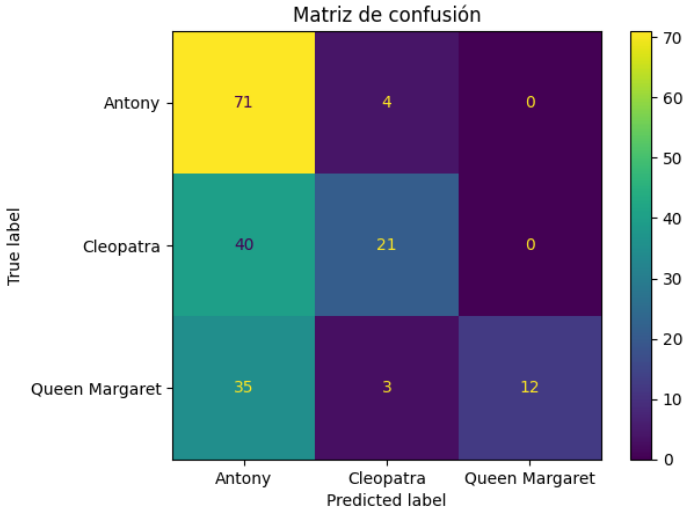
3.3. Cambio de Entrenamiento y Evaluación del Modelo

Se continúa trabajando con la representación que utiliza la medida IDF y filtra las stopwords, entrenando un modelo **Multinomial Naive Bayes**. Este clasificador probabilístico, asigna la clasificación con mayor probabilidad de ser correcta, utiliza el teorema de Bayes ($P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)}$) para inferir las probabilidades de cada clase (B) según las ocurrencias presentes en el conjunto de entrenamiento (A).

A partir del clasificador se predice sobre el conjunto de test, obteniendo un accuracy ($\frac{TP+TN}{TP+FP+TN+FN}$) de 0,559. Indica que de las 186 predicciones, sólo 104 fueron correctas. Esto no nos da una idea de cómo están distribuidas las clasificaciones correctas ni las incorrectas, por lo que puede causar problemas cuando existen categorías menos representadas, ya que la importancia de su clasificación depende de cuántas ocurrencias posean en el conjunto evaluado.

La matriz de confusión revela un primer problema con el clasificador. Tiene una importante predilección por Antony, ya que le adjudica el 146 de los 186 párrafos donde solo 71 son verdaderos positivos.

Figura 9. Matriz de confusión para el clasificador Multinomial Naive Bayes.



Utilizando la función `clasification_report` se obtienen las principales medidas de desempeño. Donde pueden apreciarse el valor de la precisión de las categorías, recall y f1-score (sus definiciones matemáticas se detallan en la Ecuación 3):

- Precisión: medida de cuántas veces la predicción fue correcta. Se puede corroborar el problema notado en la matriz de confusión, en donde la predicción

de Antony sólo es correcta un 49% de las veces, y como caso opuesto las predicciones de Queen Margaret fueron todas correctas.

- Recall: medida de cuántos de los elementos (párrafos) que pertenecen a una categoría (personaje), se lograron identificar. Se observa la relación inversa a la precisión, en donde entre más cauto sea al momento de predecir, se obtiene mayor precisión pero menor recall.
- F1-score: dada la relación inversa entre la precisión y el recall, el f1-score logra un balance entre la necesidad de una predicción correcta y un alto nivel de recuperación.

Ecuación 3. Definición de precisión, recall y f1-score.

$$precision = \frac{TP}{TP+FP} ; recall = \frac{TP}{TP+FN} ; f1 - score = \frac{2\,x\,precision\,x\ recall}{precision + recall}$$

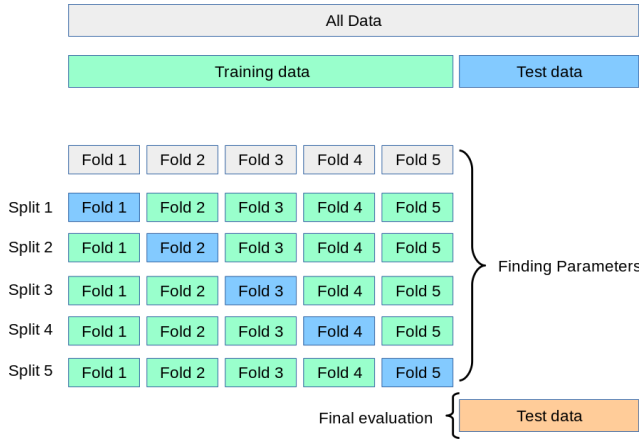
Tabla 4. Desempeño del modelo Multinomial Naive Bayes.

Personaje	Precision	Recall	F1-score	Support
Antony	0.49	0.95	0.64	75
Cleopatra	0.75	0.34	0.47	61
Queen Margaret	1	0.24	0.39	50

Habiendo identificado 3 parámetros que pueden definir el modelo: eliminación de stopwords, uso de n-gramas de una y dos palabras, y uso del algoritmo IDF. Se definen 8 posibles configuraciones para la paramétrica (presentadas en el Anexo 4. Configuraciones de los parámetros).

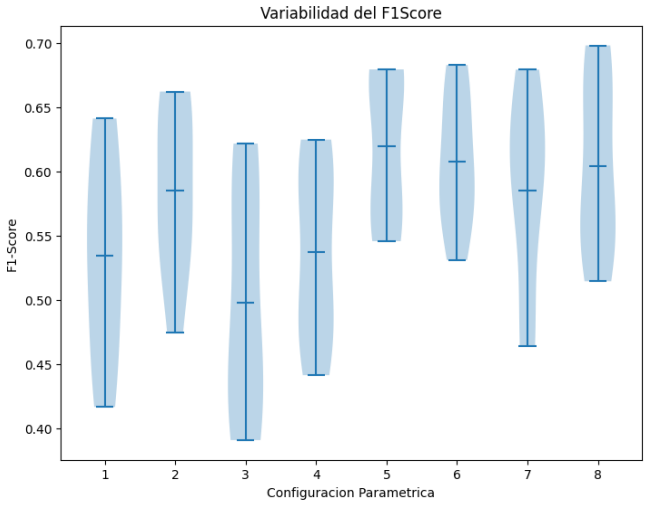
Para definir cuál de los conjuntos paramétricos se selecciona, se recurre a la técnica de **validación cruzada**. La validación cruzada es una técnica de remuestreo que permite extender la utilidad del conjunto de entrenamiento permitiendo la reutilización de sus elementos sin incurrir en sobreajuste. Para esto se particiona el conjunto de entrenamiento en k dobleces (folds en inglés), y a partir de estos se crean k separaciones (splits en inglés) en la que cada una de estas usará un fold distinto como conjunto de validación y el resto de estos como conjuntos de entrenamiento. En cada separación se podrá estudiar el desempeño de una configuración paramétrica al realizar un proceso de entrenamiento independiente por cada k-1 fold, validándolo contra el fold de validación. Pudiendo seleccionar la configuración paramétrica que reporte el mejor promedio de resultados según la métrica deseada (precisión, recall o f1-score). Para luego entrenar usando el conjunto de entrenamiento original (compuesto de todos los folds) y la mejor configuración paramétrica, evaluando contra el conjunto de test.

Figura 10. Diagrama de la técnica validación cruzada.



Para cada una de las configuraciones paramétricas, por cada split se entrena con cada fold de prueba y evaluando cada instancia contra el de fold de validación, obteniendo para estos su: accuracy, precision, recall y F1-Score (Anexo 4. Configuraciones de los parámetros). Decidiendo priorizar el F1-Score, por balancear tanto la precision como el recall, se grafica en un diagrama de violín los resultados de las distintas configuraciones paramétricas.

Figura 11. Variabilidad del F1-score para cada paramétrica.



Tomando en cuenta la variabilidad del f1-score (Figura 11) y el promedio de las métricas (Tabla 5) se decide utilizar la configuración paramétrica 5 por contar con un mejor promedio de la métrica F1-Score. La configuración 5 quita las stopwords, sólo utiliza palabras y no utiliza el algoritmo IDF.

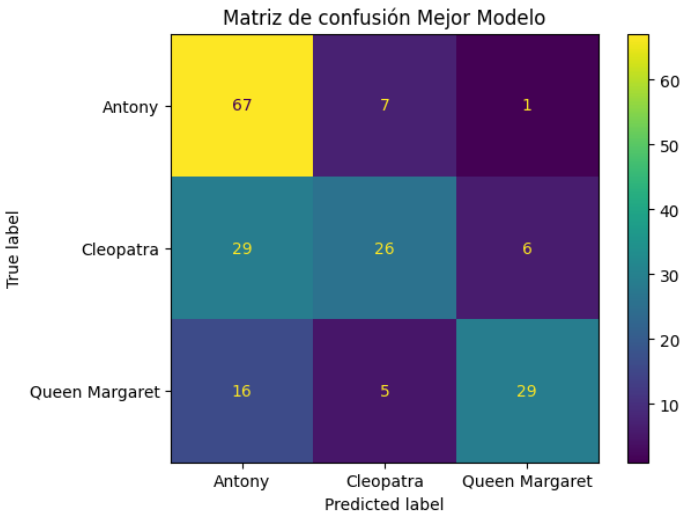
Tabla 5. Promedio de las métricas para cada configuración.

Configuración	Accuracy	Precision	Recall	f1-score
1	0.559948	0.602985	0.559948	0.520601
2	0.594340	0.628558	0.594340	0.575592
3	0.562544	0.633999	0.562544	0.510936
4	0.565239	0.610483	0.565239	0.528903
5	0.639114	0.664973	0.639114	0.629814
6	0.623290	0.653825	0.623290	0.618393
7	0.604772	0.628811	0.604772	0.582732
8	0.619048	0.657035	0.619048	0.602885

Vale mencionar que la configuración 6 obtiene resultados muy cercanos a la 5 y sería una alternativa viable.

Se entrena con la configuración 5 contra todo el conjunto de entrenamiento y se predice el conjunto de evaluación obteniendo la matriz de confusión representada en la Figura 12 y en la Tabla 5 las métricas de desempeño.

Figura 12. Matriz de confusión para el clasificador Multinomial Naive Bayes con la configuración 5.



En la Figura 12 puede apreciarse un mejor desempeño, que en la Figura 9, en las clases menos representadas dentro del corpus (Cleopatra y Queen Margaret).

Tabla 5. Desempeño del modelo Multinomial Naive Bayes con la configuración 5.

Personaje	Precision	Recall	F1-score	Support
Antony	0.60	0.89	0.72	75
Cleopatra	0.68	0.43	0.53	61
Queen Margaret	0.81	0.58	0.67	50

El principal problema, a nivel de implementación, del modelo Bag of Words, es la llamada maldición de la dimensionalidad (efecto Hughes) ya que para tener un modelo con un léxico adecuado se requiere aumentar el tamaño del vector en que se representa el documento. Generando espacios con demasiadas dimensiones, lo que dificulta su tratamiento y almacenamiento.

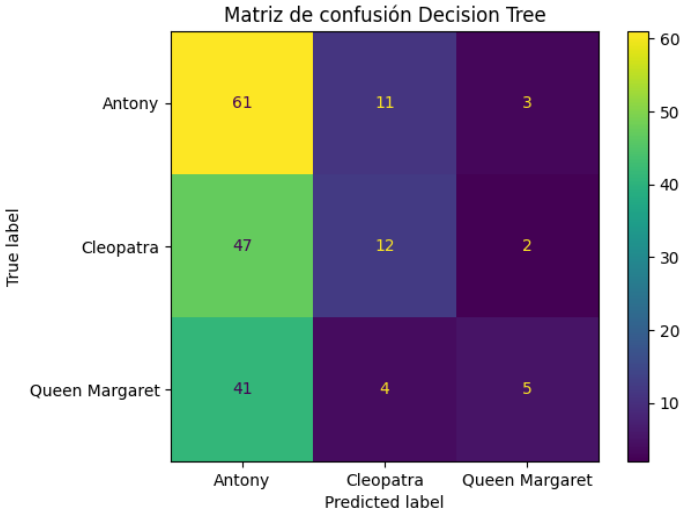
Tanto la técnica Bag of Words como la medida TF-IDF comparten el defecto de no ser capaces de modelar y mantener la información semántica incluida en un documento, un ejemplo de esto se da con la frase “Si Luciana viaja y Federico cocina, entonces ella viaja”, que sería representada igual que la frase “Si Federico viaja y Luciana cocina, entonces ella viaja”. Si bien es posible atenuar levemente esto con el uso de n-gramas donde al representar “Luciana Viaja” sería un solo bigama, no resolvería la igualdad semántica entre “Luciana viaja” y “ella viaja”.

3.4. Cambio de Clasificador

Se entrena un clasificador de Árbol de Decisión [7] provisto por Scikit Learn. Este tipo de clasificadores realiza preguntas sobre las features del elemento a clasificar moviéndose en el árbol de preguntas según sus respuestas. Partiendo del nodo inicial (raíz) hasta llegar a un nodo final (hoja) en el que ya no tenga más preguntas para hacer. Al momento de entrenar buscará ubicar las preguntas que aporten más información en los primeros niveles, valiéndose para esto de distintos criterios (como la entropía o el índice de gini). Un ejemplo popular de un uso intuitivo de los árboles de decisión se presenta en el juego Cara a Cara al comenzar con la pregunta: “¿Tu personaje es mujer?” (lo que descarta la mitad de los personajes) para luego continuar preguntado por las características que descarten más personajes, hasta que solo quede uno.

Luego de entrenar el clasificador con el conjunto de entrenamiento, se predice el de validación, obteniendo la matriz de confusión presentada en la Figura 13 y en la Tabla 6 las métricas de performance.

Figura 13. Matriz de confusión para el clasificador Árbol de decisión.



En la Figura 13 puede notarse un peor desempeño que en la Figura 9 (Naive Bayes inicial) y en la Figura 12 (Naive Bayes mejorado).

La diferencia con la Figura 12 puede deberse a que la selección de la configuración paramétrica que define la representación de los datos fue realizada para mejorar el desempeño para el Naive Bayes. No se encuentran motivos para la diferencia con la Figura 9.

Tabla 6. Desempeño del clasificador Árbol de decisión.

Personaje	Precision	Recall	F1-score	Support
Antony	0.52	0.61	0.56	75
Cleopatra	0.49	0.54	0.52	61
Queen Margaret	0.68	0.42	0.52	50

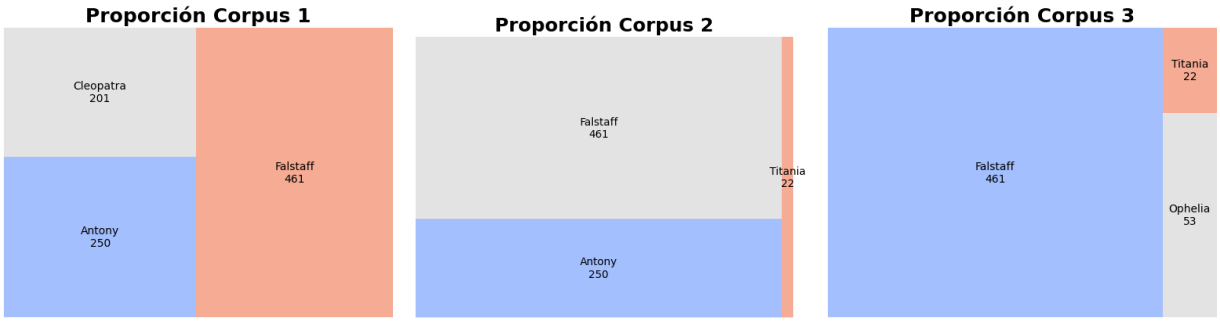
3.4. Cambio de Personajes

Se varía el conjunto de entrenamiento, cambiando los personajes dueños de los párrafos que lo integran. Preparando 3 conjuntos de personajes para observar cómo se altera el desempeño del clasificador a medida que el corpus de entrenamiento se vuelve más desbalanceado. Se utiliza el coeficiente de gini como una medida de desbalance de los datos.

Considerando el corpus 1 integrado por Falstaff (461 párrafos), Antony (250 párrafos) y Cleopatra (201 párrafos) relativamente balanceado, con un valor de gini 0.19. El corpus 2 compuesto de Falstaff (461 párrafos), Antony (250 párrafos) y Titania (22 párrafos)

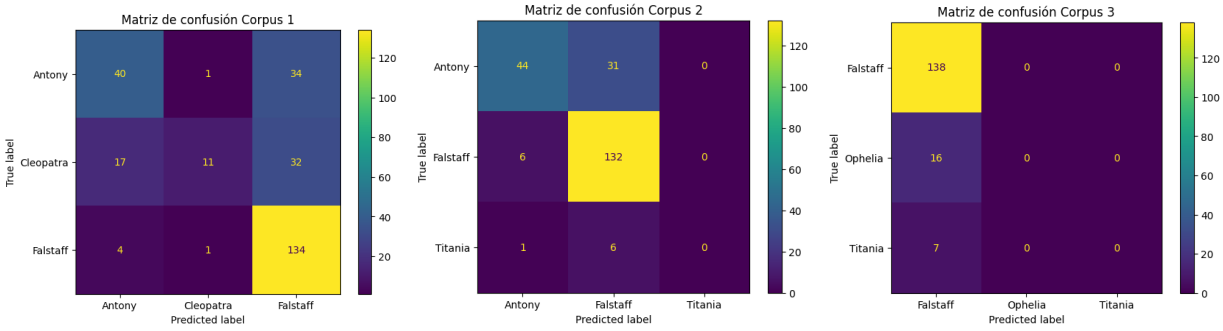
con un desbalanceado escalonado, con un valor de gini 0.39. Y el corpus 3 constituido por Falstaff (461 parrafos), Titania (22 párrafos) y Ophelia (53 párrafos) con dos categorías significativamente más pequeñas que la mayor, con un valor de gini 0.54.

Figura 14. Diferencia de balance entre los tres Corpus.



Entrenando y evaluando sobre cada uno de estos corpus, pudiendo observarse en la Figura 17 cómo se degrada la capacidad del clasificador entre más desbalanceado sea su corpus (medido según el coeficiente de gini), perdiendo la capacidad de predecir las categorías menos representadas.

Figura 15. Matrices de confusión de los tres corpus.



Para prevenir este tipo de degradación sería prudente aplicar técnicas de balanceo de datos, que agregando o quitando datos a cierta clase logran nivelar las proporciones del corpus. [6] Las técnicas de sobremuestreo, como Adaptive Synthetic Algorithm (ADASYN) y Synthetic Minority Oversampling (SMOTE) agregan datos de las clases menos representadas. Las de submuestreo, como Edited Nearest Neighbor (ENN), Random Under Sampling (RUS) y Tomek Links (TL), quitan datos de las clases más representadas. Mientras que las técnicas híbridas, como la Easy Ensemble (EE) o Balance Cascade (BC), combinan técnicas de sobremuestreo y submuestreo para lograr un mejor equilibrio [8].

3.4. Técnica alternativa

Una técnica alternativa para extraer features del texto, que a diferencia del Bag of Words con la medida TF-IDF, sí considera el contexto semántico y tiene la capacidad de manejar sinónimos es el **método Word2Vec**. En el que partiendo de un gran corpus de texto se genera un espacio vectorial de muchas dimensiones donde las palabras con significado similar estarán cerca.

Para esto se utiliza una red neuronal sencilla, con una capa de entrada en la que cada palabra del corpus tiene su propia entrada, una capa de proyección con tantas funciones de activación como las dimensiones que se desee dar al espacio vectorial y una capa de salida con tantas salidas como palabras del corpus. Esta red será optimizada para poder predecir una palabra según algún contexto (como la palabra anterior, la siguiente o ambas). Terminado el entrenamiento se usará el valor de los pesos que van de la entrada de cada palabra a las funciones de activación para ubicar esta palabra en el espacio vectorial.

Cuando el entrenamiento se realiza de forma correcta el espacio vectorial generado conserva las propiedades semánticas, incluso permitiendo utilizar propiedades aritméticas con conceptos (sumando el vector *madre* al vector *hombre* se obtendrá un punto del espacio cercano al vector *padre*). Luego de contar con este espacio vectorial es posible extender su utilidad para representar documentos tomando el promedio de las palabras que lo componen.

A modo anecdótico, uno de los autores de este documento considera importante mencionar la similitud notada (probablemente debido a su falta de una correcta formación Filosófica) entre el espacio vectorial generado por la técnica del word2vec y el Mundo de las Ideas postulado por Platón, donde habitan los entes inmateriales, absolutos, inmutables y universales, de donde (según el filósofo) derivaría todo lo que existe en nuestro Mundo Sensible.

3.5. Modelo FastText

Otro modelo para la clasificación de texto es FastText [9]. Representa las palabras como vectores los cuales permiten, al igual que con el modelo word2vec, considerar información semántica. Otra ventaja que tiene es su capacidad para trabajar con palabras raras. Por otro lado, debido a que tiene una arquitectura de redes neuronales, requiere de mayor tiempo y recursos que el clasificador Multinomial Naive Bayes, además FastText no está diseñado para trabajar con frases largas ya que su procesamiento se enfoca en palabras.

Al entrenar el modelo se obtiene un accuracy de 0.618, la matriz de confusión se muestra en la Figura 16 y los datos de precision, recall y F1-score en la Tabla 7. Evaluando contra la métrica F1-score del clasificador Multinomial Naive Bayes se obtiene una mejor performance, y una performance parecida al que utiliza la mejor configuración.

Figura 16. Matriz de confusión para el modelo FastText.

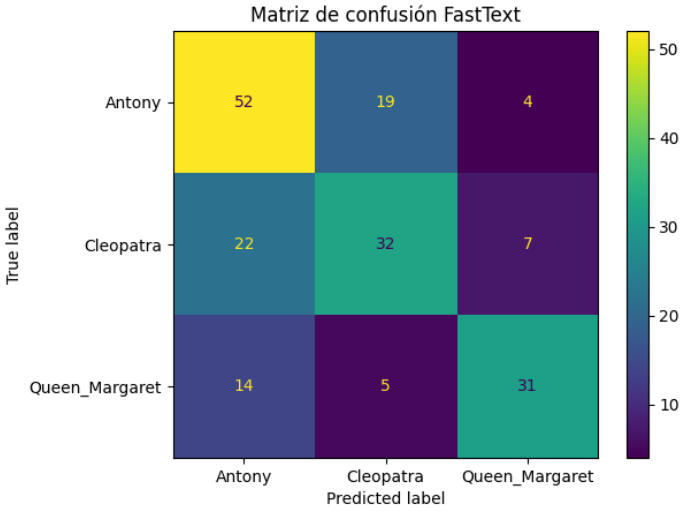


Tabla 7. Desempeño del modelo FastText.

Personaje	Precision	Recall	F1-score	Support
Antony	0.59	0.69	0.64	75
Cleopatra	0.57	0.52	0.55	61
Queen Margaret	0.74	0.62	0.67	50

Referencias bibliográficas

[1] Sitio web Relational Database Repository, accedido por última vez 21/05/2023.
https://relational.fit.cvut.cz/dataset/Shakespeare

[2] Sitio web Quora, accedido por última vez 21/05/2023.
https://www.quora.com/Which-character-has-the-most-lines-in-a-Shakespeare-play

[3] Sitio web GitHub, accedido por última vez 21/05/2023.
https://gist.github.com/sebleier/554280

[4] Test de Bechdel, accedido por última vez 21/05/2023.
https://feministfrequency.com/video/the-bechdel-test-for-women-in-movies/

[5] Sitio web de GitHub, accedido por última vez 21/05/2023.
https://github.com/lupish/introcd_grupo_17

[6] Técnicas para equilibrar conjuntos de datos desbalanceados
https://www.medwave.cl/resumenescongreso/ci2022/desarrollotecnologicoyprocesosenergeticos/8583.html

[7] Video Decision and Classification Trees, Clearly Explained!!!, accedido por última vez 06/07/2023.
https://youtu.be/_L39rN6gz7Y

[8] Exploratory Study on Class Imbalance and Solutions for Network Traffic. Classification
https://uvadoc.uva.es/bitstream/handle/10324/54159/Exploratory-study-class-imbalance.pdf

[9] Word representations - fastText.
https://fasttext.cc/docs/en/unsupervised-tutorial.html

[10] Sitio Stack Overflow para el código de gini, respuesta de Warren Weckesser. Accedido por última vez 06/07/2023.
https://stackoverflow.com/questions/39512260/calculating-gini-coefficient-in-python-numpy#:~:text=The%20Gini%20Coefficient%20of%20the,not%20%22all%20equally%20probable%22.

Anexo 1. Preguntas interesantes

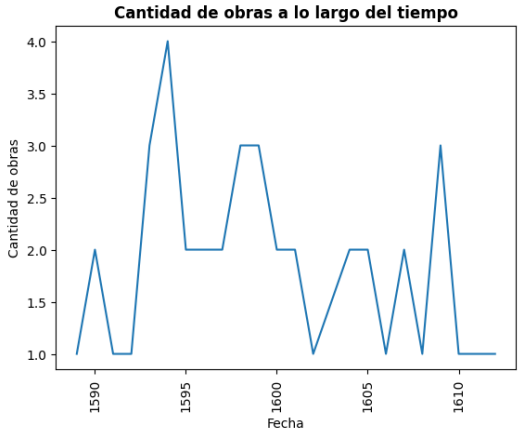
A continuación se listan preguntas que podrían ser interesantes para responder:

- Buscar frases repetidas: ¿es posible detectar frases repetitivas (conjuntos ordenados de palabras) que se utilicen de forma regular?, ¿algunos personajes tienen mayor predisposición a hablar con frases prearmadas que otros?, ¿algunos géneros son más propensos a usar frases repetidas?
- Buscar arquetipos de personajes: ¿es posible entrenar un clasificador de arquetipos de personajes utilizando este corpus?, ¿qué tipos de arquetipos de personajes se encuentran más presentes en sus obras?, ¿la capacidad de detectar arquetipos es local al género literario o a modo de ejemplo es posible entrenar un clasificador de arquetipos con comedias y que sea bueno detectando arquetipos en tragedias?
- Predecir características: ¿es posible entrenar un clasificador de personajes que prediga el sexo o la edad de estos a partir de sus líneas?
- Test de Bechdel: ¿dada la estructura de los datos es posible encontrar si una obra cumple con el Test de Bechdel [4] (al menos dos personajes femeninos, con nombres, que hablan entre sí de un tema que no sea un hombre)?
- Palíndromos: ¿qué palíndromos (cadena de palabras o letras simétricas, que pueden ser leídos igual de derecha a izquierda o izquierda a derecha) se encuentran presentes en las obras?
- Predictor de muertes: ¿es posible entrenar un clasificador que utilizando las primeras líneas de un personaje prediga si este se muere o no?, ¿cuántas líneas son necesarias para que el predictor sea eficiente?, ¿cuán dependiente del género literario es este clasificador?

Anexo 2. Obras a lo largo de los años

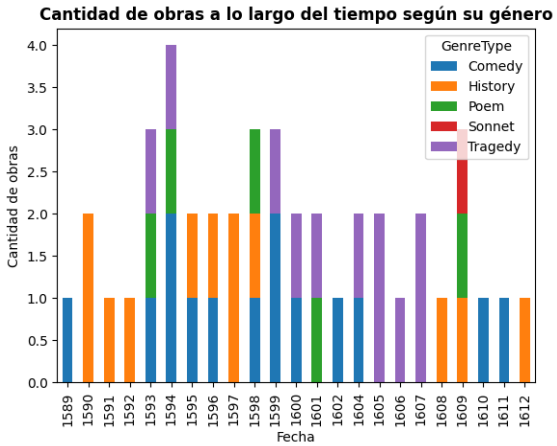
Como primer acercamiento a la información graficamos la cantidad de obras a lo largo de los años, pudiendo observarse que desde 1589 hasta 1612 el artista produjo obras de forma ininterrumpida, pero variando su productividad.

Figura 17. Gráfica mostrando la cantidad de obras a lo largo del tiempo.



Buscando qué explica la variación se vuelve a graficar la cantidad de obras a lo largo del tiempo, pero esta vez como un gráfico de barras. Pudiendo notar períodos de mayor interés en ciertos géneros, como el que va de 1590 a 1592 en que sólo compuso ficción histórica, o el de 1604 a 1607 en que seis de las siete obras producidas fueron tragedias.

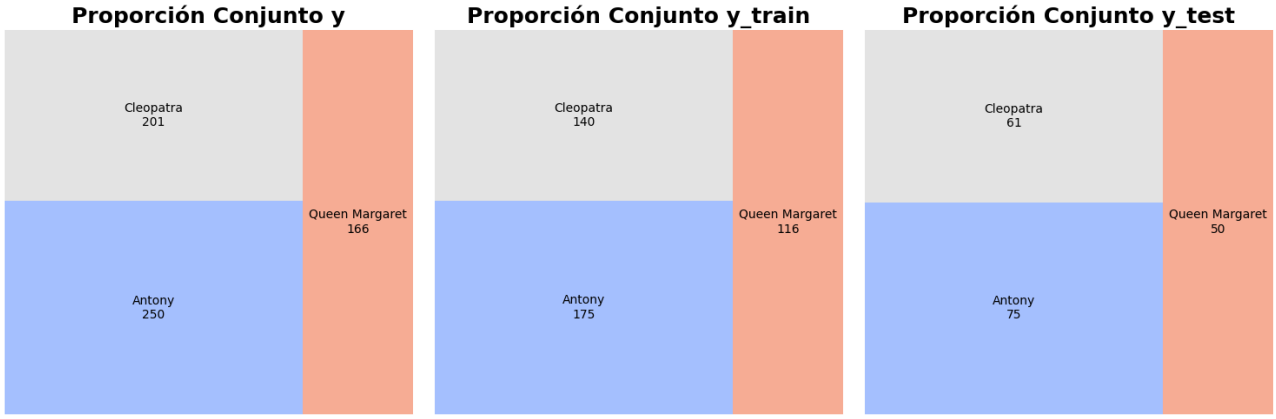
Figura 18. Gráfica sobre la cantidad de obras a lo largo del tiempo según su género literario.



Anexo 3. Proporciones en los corpus

En la Figura 19 se grafican las proporciones de cantidad de párrafos dicho por cada personaje para el total de párrafos, los pertenecientes al conjunto de entrenamiento y al de test. Los gráficos muestran que hay un balance entre la cantidad de párrafos de cada personaje en los distintos conjuntos.

Figura 19. Proporción de la cantidad de párrafos por personaje en: total de párrafos (izquierda), el conjunto de entrenamiento (medio) y conjunto de test (derecha).



Anexo 4. Configuraciones de los parámetros

La Tabla a continuación muestra las diferentes configuraciones posibles para los parámetros: utilizar stopwords, 2-gramas y uso del algoritmo IDF.

Tabla 7. Configuraciones para los parámetros.

Configuración	Uso stopwords	Uso bi-gramas	Uso IDF
1	No	No	No
2	No	No	Sí
3	No	Sí	No
4	No	Sí	Sí
5	Sí	No	No
6	Sí	No	Sí
7	Sí	Sí	No
8	Sí	Sí	Sí