“Análisis de datos y modelos de predicción: Shakespeare”

Curso: “Introducción a la ciencia de datos”

Año: 2023

Grupo: 5

Autores:
- Agustín Porley Santana
- Julián Rodríguez

Introducción a la base de datos

La base de datos a analizar es una base relacional con la obra completa de William Shakespeare, dicha base de datos cuenta con cuatro tablas:

- paragraphs
- characters
- chapters
- works

La tabla paragraphs es utilizada para guardar los párrafos de las obras. En su estructura podemos apreciar que existe una columna donde se guarda el párrafo como texto plano (PlainText). Cada entrada en esa tabla tiene asignado un número de identificación (id), un número correspondiente al párrafo (ParagraphNum), un número indicando el personaje a quien corresponde el párrafo (character_id) y un identificador de capítulo (chapter_id). El párrafo almacenado se encuentra casi totalmente identificado a excepción de la obra a la cual pertenece. La tabla characters es usada para guardar la información de los personajes. Cada entrada en la tabla cuenta con un número identificador (id), un nombre de personaje (CharName), una abreviatura del nombre (Abbrev) y una descripción (Description). Como se puede deducir esta tabla está relacionada directamente con la tabla paragraphs, donde se guarda el número identificador de cada personaje, permitiendo buscar el nombre del personaje en la tabla characters a través de su identificador. La tabla chapters tiene la función de guardar la información referente a los capítulos. En ella se puede encontrar la información del acto (Act) y la escena (Scene) a la que pertenece cada capítulo, así como también su número identificador. Del mismo modo que se relacionan las tablas paragraphs y characters, podemos obtener el nombre del capítulo al que pertenece un párrafo, extrayendo el número identificador del capítulo de la tabla paragraphs y buscar su nombre en la tabla chapters. En esta tabla también hace referencia a la obra en la que se encuentra cada capítulo, la misma queda determinada por un identificador (work_id). La tabla works guarda la información referente a las obras, donde se puede encontrar el número identificador (id), el título (Title), el título largo (LongTitle), el año en que fue escrita (Date), y el género a la que pertenece la obra. Con esta información, podemos extraer el identificador de la obra a la que pertenece cierto capítulo y encontrar toda la información de esa obra en la tabla works. Como es de esperarse, esta última tiene una relación directa con la tabla chapters..

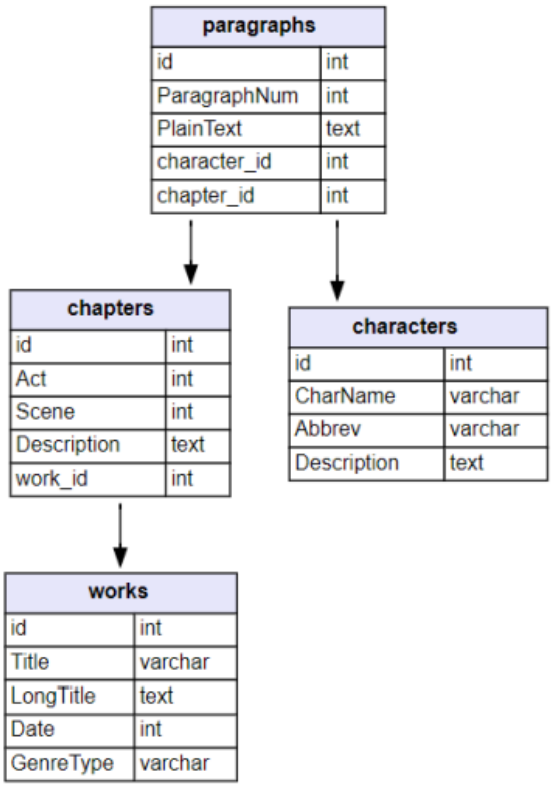

Figura 1: Relación entre las tablas

En la figura 1, se presenta un resumen que indica la relación entre las tablas de la base de datos relacional de manera más esquemática.

Análisis primario de los datos

Realizando un primer acercamiento a la calidad de los datos se puede analizar la falta de datos. Dentro de la tabla “Characters” existen datos faltantes, específicamente dentro de la categoría “Description” hay 646 espacios vacíos de 1266 y además la categoría “Abbrev” cuenta con 5 datos faltantes del total de 1266. El resto de la base de datos parecería estar completa.

También es de interés analizar la cantidad de párrafos por personaje, en este caso se puede apreciar que la mayor cantidad de texto en las obras son direcciones de escena y no diálogos de personajes. Aun así, el personaje con más diálogos es el poeta o “la voz poética de Shakespeare”. Para visualizar la producción de obras de Shakespeare a lo largo de los años se puede graficar un histograma, dividido por períodos, donde se vea la cantidad de obras generadas en cada período.

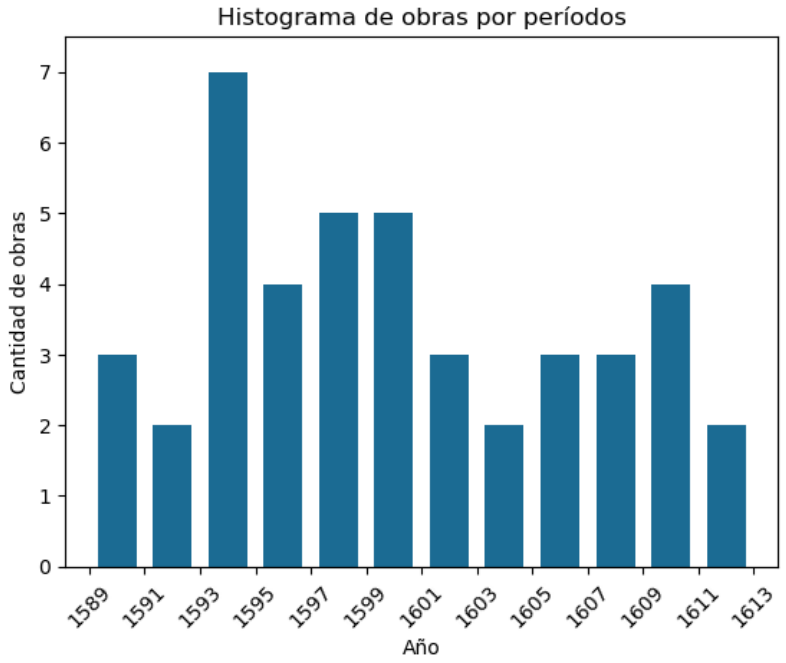

Figura 2: Histograma de obras por períodos.

En la Figura 2 se puede observar cómo su período más productivo fue entre los años 1593 y 1601. Si realizamos el gráfico análogo, pero clasificando la cantidad de obras por género se obtiene la Figura 3.

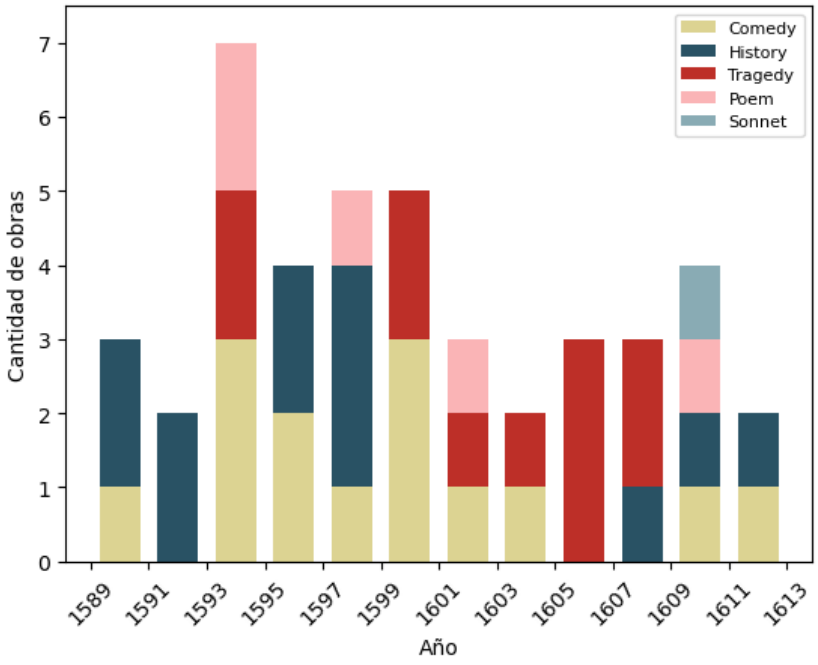

Figura 3: Histograma de obras por períodos apilado por género.

En la Figura 3 se puede observar que inició su carrera escribiendo obras de comedia e historia. Como se puede apreciar, la escritura de comedias fue una constante a lo largo de su producción.

Algo a recalcar es que hubo un período entre el 1599 y el 1607 donde no realizó obras de historia. Por último, al finalizar su carrera se introdujo en el género de los sonetos.

Con el fin de extraer más información de los datos, se decide realizar una limpieza de la base de datos. Para esto, se agregan los siguientes signos de puntuación: ";", "]", ".", ":", "!", "¡", "?", "¿", "-". Adicionalmente se transforman las palabras que comienzan o finalizan con comillas simples agregando a la función " ‘" y "’ ". En este caso las expresiones de posesión de palabras en plural serán tratadas como una palabra terminada con comilla simple y por lo tanto no será contada como una palabra diferente. A modo de ejemplo, si en el texto estuviera la frase "My parents' car.", luego de la transformación quedaría "my parents car". En este caso "parents' " y "parents" serán tratados como la misma palabra. En el caso de las expresiones de posesión en singular o las contracciones gramaticales habituales,serán tratadas como palabras diferentes de las que no tengan contracciones. Si en el texto aparece la expresión "What's", luego de la transformación queda como "what's" y es tratado como una palabra diferente de "what". Luego de realizada la limpieza, se puede extraer fácilmente las palabras más frecuentes. Por medio de un gráfico de barras (Figura 4) se logra identificar con facilidad cuales son las palabras más frecuentes considerando toda la obra.

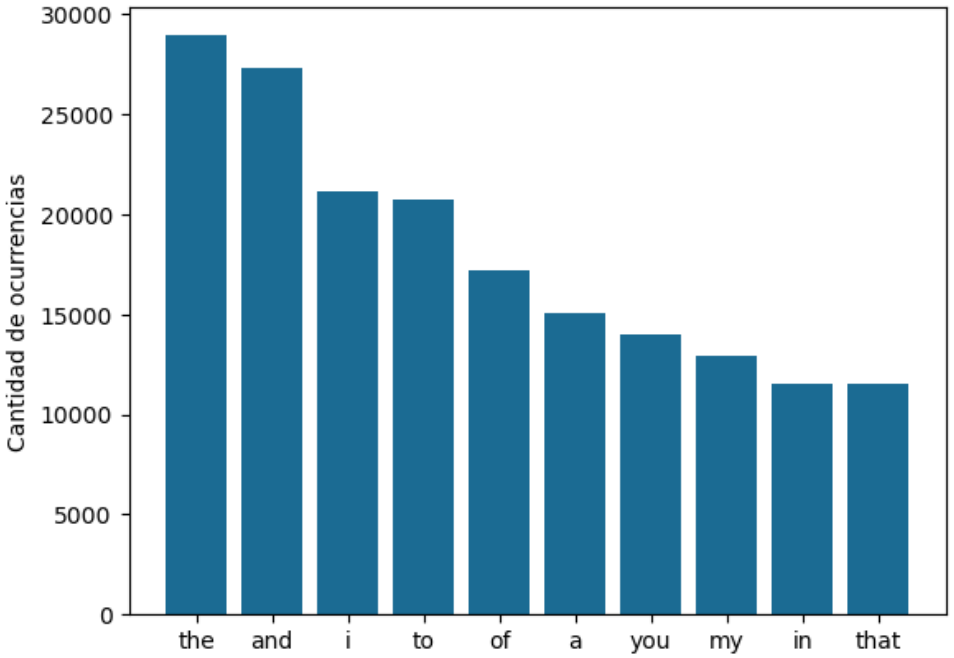

Figura 4: Ocurrencia de palabras en toda la obra

Ya evaluamos qué personaje posee mayor cantidad de párrafos, pero algo interesante es observar cuál personaje posee mayor número de palabra (Figura 5).

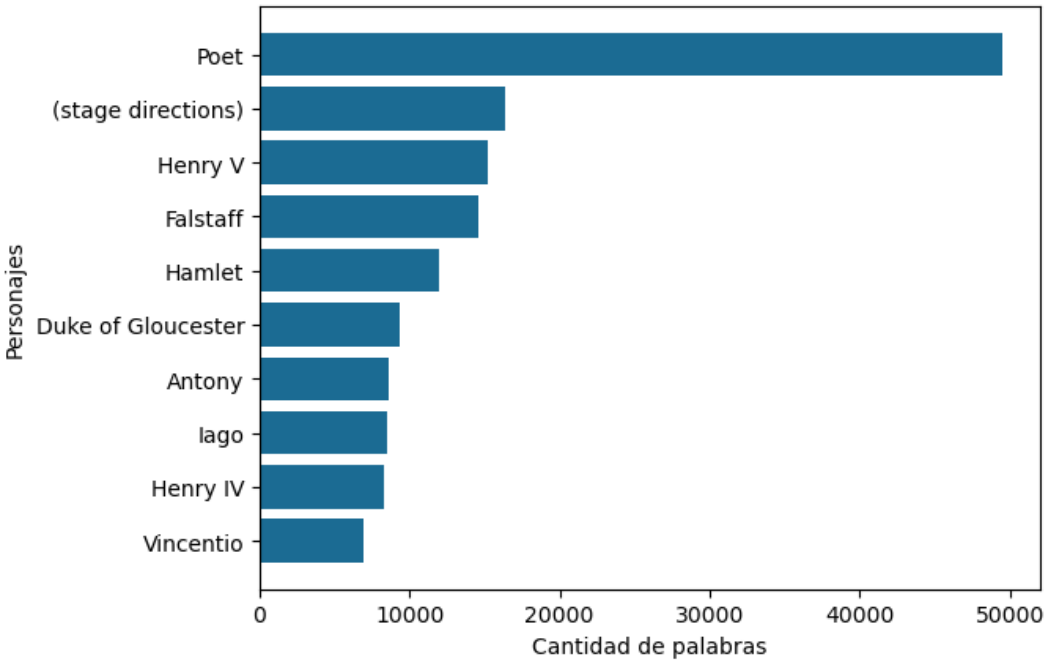

Figura 5: Cantidad de palabras por personaje

Lo que primero llama la atención es que hay un personaje con muchas más palabras que el resto. Como se agrupó por nombre tal vez personajes con el mismo nombre se contaron de manera repetida pero que en realidad tienen un número de identificación diferente. Lo otro que resalta es que aparece una el nombre “(stage directions)” nuevamente el cual no es estrictamente un personaje, sino que son las explicaciones para el movimiento de los personajes en escena. En este caso habría que omitir en la visualización a “(stage directions)” y corroborar que el nombre “Poet” corresponde a un solo identificador. En caso de que sea más de uno se debería volver a ordenar filtrando por identificador y luego asignando nombre. Aun así, parece lógico que la voz poética sea el “personaje” con más palabras debido a que los poemas se construyen de la misma, a diferencia de las obras donde las direcciones escénicas tienen un papel secundario.

Estudio de caso: Análisis de 3 personajes.

Filtrado y creación de los set de trabajo

Para poder realizar un análisis más detallado se decide filtrar los datos de manera de obtener solamente los párrafos de tres personajes (Cleopatra, Antony y Queen Margaret). Se pretende, a través de diferentes técnicas, poder deducir a qué personaje corresponde determinado párrafo.

Luego del filtrado se procede a separar el set de datos en ‘train’ y ‘test’, de esta forma poder encontrar los parámetros de los modelos con el set de entrenamiento y poder validar los mismos contra un set de datos que no fue utilizado para el entrenamiento (set de testeo) .

En la Figura 6 se presenta un gráfico de barras donde se puede ver la cantidad de párrafos por personaje, así como la división entre los datos de ‘train’ y ‘test’.

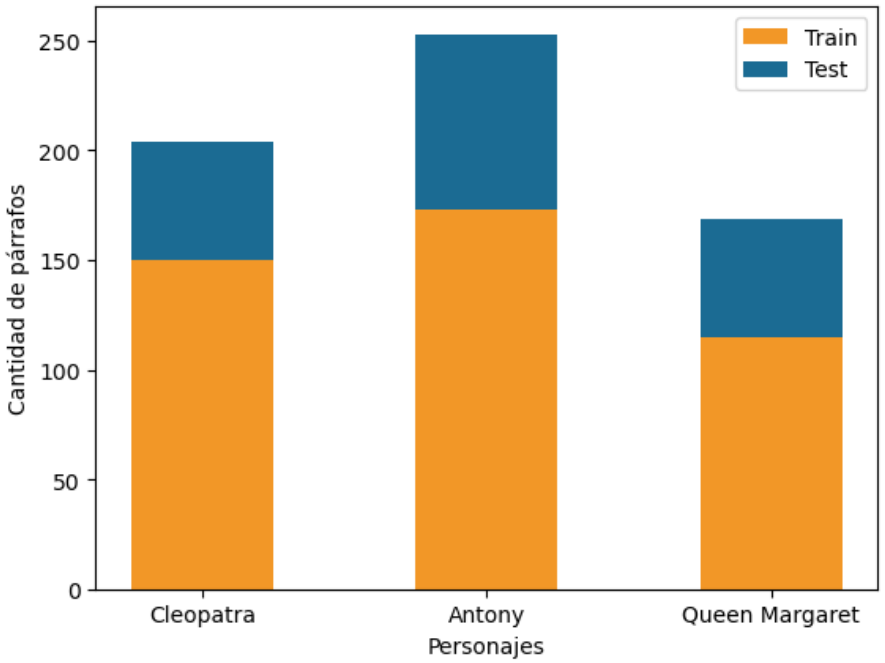

Figura 6: Cantidad de párrafos por personaje y proporción de 'train' y 'test'.

Si bien el gráfico muestra que la cantidad de párrafos no es igual para todos los personajes, el desbalance no es crítico. La proporción usada para 'train' y 'test' fue del 30%.

Extracción de features: 'Bag of Words' [1]

En primera instancia, para entrenar los diferentes modelos es necesario transformar el texto en alguna clase de secuencia numérica. En este caso se usará la técnica conocida como 'Bag of Words', donde a cada palabra se le asigna una posición dentro de un vector (normalmente la posición se asigna por orden alfabético) y luego se contabiliza las ocurrencias de esa palabra dentro de un documento.

A modo de ejemplo, supongamos que tenemos dos documentos (en esta caso dos oraciones) :

- El perro se llama Fido.
- El perro es blanco.

Podemos representar estos dos documentos de la siguiente forma:

Documento	blanco	el	es	fido	llama	perro	se
doc_1	0	1	0	1	1	1	1
doc_2	1	1	1	0	0	1	0

Tabla 1: Representación de textos utilizando 'bag of words'.

A este tipo de matriz resultante se conoce como 'sparse matrix', donde cada fila representa un documento, cada columna es una palabra distinta, y cada entrada en la matriz es la cantidad de veces que aparece dicha palabra en ese documento. De ese modo el texto

queda representado únicamente por números. Se denomina ‘sparse matrix’ o matriz dispersa debido a que la gran mayoría de los valores en la matriz son ceros, debido a que no todos los documentos tienen la totalidad de las palabras.

La desventaja de esta metodología es que se ocupa una columna por cada palabra, para cada texto, por ende, se indican las palabras que pertenecen al mismo como las que no. Si tuviera que alojarse en memoria varios documentos, con una gran diversidad de palabras, esto requeriría muchos recursos.

Otra desventaja de esta técnica es que no contempla el órden de las palabras dentro de un documento, esto evita que se contemple la sintaxis del idioma. Si tenemos un texto que dice ‘el perro es blanco’ va a tener un significado diferente a que sí dice ‘el blanco es perro’. Sin embargo ambos textos tienen la misma representación numérica.

Para tener en cuenta el orden, se puede utilizar una representación conocida como ‘n-gram’, donde las palabras son tomadas en grupo de n palabras contiguas en el documento. A modo de ejemplo, un bi-grama corresponde a realizar la misma representación que la Tabla 1 pero tomando grupos de 2 palabras contiguas.

Documento	el perro	perro se	se llama	llama fido	perro es	es blanco
doc_1	1	1	1	1	0	0
doc_2	1	0	0	0	1	1

Tabla 2: Representación de texto en bi-grama.

Se pueden combinar las representaciones e implementar el mono-grama y adicionar el bi-grama. En este sentido se tiene la frecuencia de las palabras dentro de un texto y en cierta medida también se contempla el órden, aunque este muy acotado.

‘Term Frequency’ y ‘Inverse Document Frequency’ [1]

Luego de aplicar la representación de la ocurrencia de las palabras dentro de un texto, se puede ir un paso más y normalizar la misma. A este resultado se lo conoce como ‘term frequency’ y se implementa dividiendo la ocurrencia de cada palabra por la cantidad de palabras de un documento. La información capturada por este parámetro es cuánto sobresale dicha palabra en el documento, cuan más alta la frecuencia es más probable que dicha palabra sea una buena descripción del contenido del documento.

En uno de los ejemplos anteriores, tenemos:

- El perro se llama Fido.

El documento cuenta con 5 palabras, ya cada palabra aparece una sola vez, por lo que a cada palabra le corresponde el valor 0.2.

Documento	blanco	el	es	fido	llama	perro	se
doc_1	0	0.2	0	0.2	0.2	0.2	0.2

doc_2	0.25	0.25	0.25	0	0	0.25	0

Tabla 3: Representación en 'term frequency'

Otro parámetro importante para medir la calidad de los datos es la 'inverse Document Frequency' o la frecuencia documental invertida, la misma mide cuantas veces cuantas veces la palabra aparece en el total de los documentos. Se calcula como el logaritmo de la relación entre el número de documentos y la cantidad de documentos donde aparece la palabra i (dfi).

$$IDF_i = log\left(\frac{N}{df_i}\right)$$

Es una representación que de cuanta información posee dicha palabra, una palabra que se repite en todos los documentos tiene una IDF de 0, indicando que esta palabra no aporta información específica a ningún documento.

blanco	el	es	fido	llama	perro	se
log(2)	0	log(2)	log(2)	log(2)	0	log(2)

Tabla 4: Valor de 'Inverse Document Frequency'

PCA (Principal Component Analysis) [2][3]

El PCA es una técnica utilizada para análisis de datos de gran dimensión. El objetivo de esta técnica es poder reducir las dimensiones logrando hacer más manejable el set de datos para poder visualizar ciertas características perdiendo la menor cantidad de información posible. Para que esto se cumpla la distorsión de los datos debe ser pequeña.

A través del PCA, se busca la recta que mejor se aproxime a los datos, llamada la primera componente principal. Luego en función de los requerimientos se pueden ir calculando las siguientes componentes, las cuales son ortogonales a las anteriores. A grandes rasgos se transforma el set de datos proyectando los mismos en las componentes principales, reduciendo su dimensión. El resultado es un set de datos, de menor dimensión, pero que podría conservar gran parte de la información.

Para cuantificar qué tanta información contiene cada componente, se puede calcular el porcentaje de varianza explicado por cada componente. Dicho valor nos da una medida de que tan dispersos son los datos en la componente calculada, nos da una idea de la cantidad de variabilidad en los datos que puede ser atribuida a esa componente. En otras palabras, cuánto del total de la varianza es "explicada" por la componente. En la Figura 8, se puede apreciar una gráfica que representa la varianza explicada por cada componente y la acumulacion de las mismas, al aumentar el número de componentes se puede apreciar que la varianza explicada de cada componente tiende a 0 y que el 1% de la varianza se encuentra en la primera componente, mientras que la varianza acumulada tiene un valor de 6% cuando se utilizan 10 componentes principales.

Se realizó un PCA para el set de datos de entrenamiento, una representación gráfica del mismo se puede apreciar en la Figura 7. Como se puede apreciar, visualmente es difícil

encontrar grupos de datos o tendencias marcadas en dicho análisis que permitieran separar grupos o clusters.

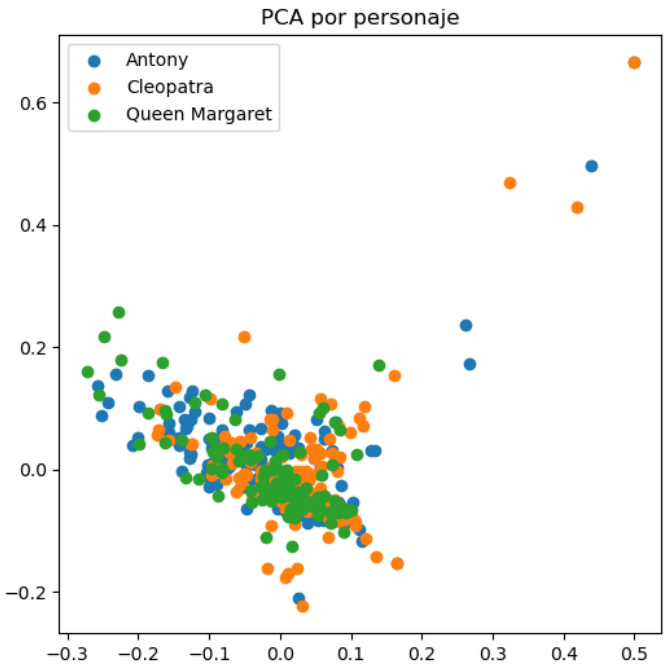

Figura 7: Representación gráfica de las dos primeras componentes principales

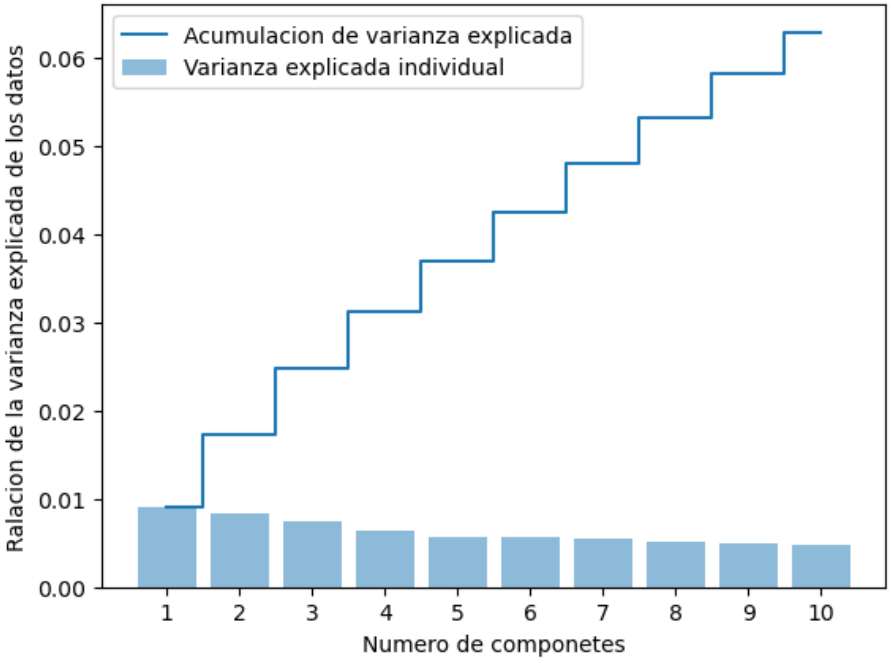

Figura 8: Varianza explicada por componente y acumulacion de la varianza explicada

Modelo Multinomial Naive Bayes [4][5]

Para poder generar un modelo que prediga a qué personaje pertenece un párrafo determinado, se selecciona el Modelo Multinomial Naive Bayes. El modelo multinomial Naive Bayes es un modelo que se basa en la probabilidad condicional y el teorema de Bayes. Para resumir el modelo, se postula el siguiente ejemplo, tomando los personajes Cleopatra (C), Antony (A) y Queen Margaret (QM).

Supongamos como es el caso de este ejercicio que queremos predecir a partir de las palabras que dicen los personajes si un párrafo pertenece a C, A o QM. Asumimos que el set de palabras por personaje es el siguiente:

C: {[Hola Hola Hola]; [Egipto]; [Egipto Noche]}

A: {[Hola Cleo]; [Cleo Egipto]; [Noche]; [Noche Amor Amor]}

QM:{[Hola Buenas Buenas Buenas]; [Londres Londres]; [Londres Hola Hola Hola]}

Primero calculemos la probabilidad de que a un personaje le corresponda un párrafo:

$$P(C) = \frac{\#párrafos\ \ dichos\ por\ C}{\#párrafos\ \ total} = \frac{3}{10} = 0.30$$

$$P(A) = \frac{4}{10} = 0.40$$

$$P(QM) = \frac{3}{10} = 0.30$$

Por otro lado veamos la probabilidad de cada palabra sea dicha por ese personaje, calculada como la frecuencia relativa de una palabra en el total de palabras de un personaje:

Personajes	Hola	Egipto	Noche	Cleo	Amor	Londres	Buenas
C	0.5	0.33	0.17	0	0	0	0
A	0.125	0.125	0.25	0.25	0.25	0	0
QM	0.40	0	0	0	0	0.30	0.30

Tabla 5: Frecuencia relativa de cada palabra por personaje.

Como se puede apreciar hay varias palabras que poseen como valor de probabilidad 0, lo cual llevaría el modelo a cometer errores y obteniendo una predicción incorrecta. Para evitar dicho problema, se agrega el valor alfa a la cantidad de veces que se agrega el conjunto de palabras en todos los personajes.

C: {Hola Hola Hola Egipto Egipto Noche} + α * {Hola Egipto Noche Cleo Amor Londres Buenas}

A: {Hola Cleo Cleo Egipto Noche Noche Amor Amor} + α * {Hola Egipto Noche Cleo Amor Londres Buenas}

QM:{Hola Buenas Buenas Buenas Londres Londres Londres Hola Hola Hola} + α * {Hola Egipto Noche Cleo Amor Londres Buenas}

Esto es equivalente a decir que al conteo de cada frecuencia relativa se le adiciona el valor alfa.

Tomando en este ejemplo $\alpha = 1$, las probabilidades son:

Personajes	Hola	Egipto	Noche	Cleo	Amor	Londres	Buenas
C	0.31	0.23	0.15	0.08	0.08	0.08	0.08
A	0.13	0.2	0.13	0.2	0.2	0.07	0.07
QM	0.29	0..06	0.06	0.06	0.06	0.57	0.57

Tabla 6: Frecuencia relativa adicionando alfa de cada palabra por personaje.

Una vez calculadas dichas probabilidades a partir del set de entrenamiento el modelo predice si el párrafo pertenece a un personaje de la siguiente manera:

Asumamos que quiero saber a qué personaje pertenece el párrafo (Par) “Hola Hola Hola Noche Amor”. El modelo primero calcula la probabilidad de que dicho párrafo pertenezca a los diferentes personajes:

$$P(Par|C) = P(C) * P(Hola|C)^3 * P(Noche|C) * P(Amor|C)$$

$$P(Par|C) = 0.30 * 0.31^3 * 0.15 * 0.08 = 1.072e\text{-}4$$

$$P(Par|A) = 0.40 * 0.13^3 * 0.13 * 0.2 = 0.2285e\text{-}4$$

$$P(Par|QM) = 0.30 * 0.29^3 * 0.06 * 0.06 = 0.2634e\text{-}4$$

Como el valor de la probabilidad es mayor para Cleopatra, el modelo predeciría que este párrafo pertenece a este personaje.

Para evitar errores por punto flotante al ser productorias muy extensas, se utiliza la sumatoria del logaritmo de cada probabilidad.

Métricas del modelo [2]

La accuracy del modelo planteado tiene un valor de 0.47, lo que corresponde a decir que el modelo acertó un 47% de las veces. El problema de esta métrica es que solo nos muestra cuando el modelo acertó, no nos permite distinguir si para un conjunto determinado el modelo se comporta de mejor manera o no, si hay un desbalance en el conjunto de datos esta métrica puede llevarnos a la idea que el modelo es bueno cuando en realidad solo predice que los datos pertenecen al conjunto con mayor volumen de datos.

Para evaluar la calidad de los modelos se propone calcular la precisión y la recuperación de dicho modelo.

La precisión es el porcentaje de aciertos que obtuvo el modelo para una determinada categoría. Es decir, del total de las predicciones de cierta categoría, cuántas pudo acertar. A modo de ejemplo, si se tuviera una categoría 'azul', y el modelo predice que hay tres elementos que pertenecen a esa categoría, pero solo dos de ellos realmente pertenecen, entonces la precisión para esa categoría toma el valor dos tercios.

La recuperación en cambio, contabiliza qué porcentaje de elementos de una categoría puede acertar. A modo de ejemplo si en un set de datos se tiene 5 elementos de una categoría, pero el modelo solamente acierta en 3 de ellos, entonces la recuperación toma el valor de tres quintos.

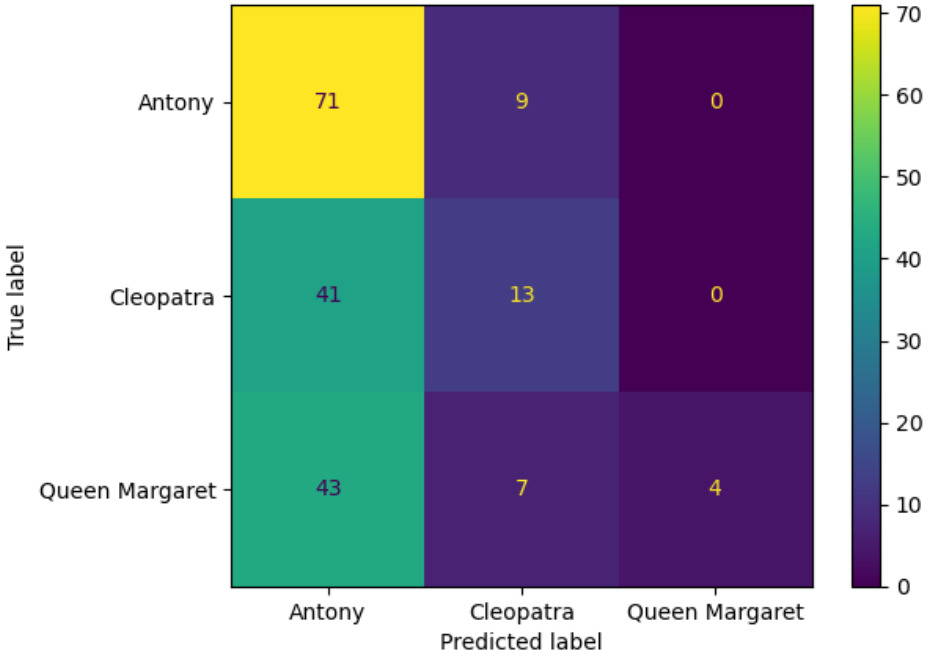

Figura 9: Matriz de confusión del modelo planteado

Para el caso de la matriz de confusión, la precisión se calcula como el valor en la posición *i,i* dividido por la suma de la columna *i*. La recuperación, en cambio, se calcula como el valor en la posición *i,i* dividido por la suma de la fila *i*.

Personaje	Precisión	Recuperación
Antony	0.46	0.89
Cleopatra	0.45	0.24
Queen Margaret	1.00	0.07

Tabla 7: Valores de Precisión y Recuperación

Validación cruzada [2]

Esta técnica consiste en separar el set de entrenamiento en varias partes, luego usar todas las partes menos una para entrenar, y la parte que queda sola para testear. Luego se repite el procedimiento cambiando la parte seleccionada para testear. Al finalizar se evalúa el promedio de los parámetros.

Figura 10: Esquema explicativo de la validación cruzada

Esta técnica permite reducir la varianza de los resultados. La desventaja es que hay que ejecutar el algoritmo de entrenamiento la cantidad de veces correspondiente a las divisiones que tengamos.

Hiper-parámetros de los modelos [2]

Cada etapa en el modelado tiene una secuencia de ajustes que denominamos hiper-parámetros. Estos ajustes permiten obtener resultados diferentes con el mismo set de datos, y dependiendo de los datos, algunas configuraciones pueden ser más útiles que otras.

Para el caso de estudio, utilizando la técnica de 'Bag of words', podemos decidir si es conveniente o realizar un filtrado por 'stop words', o incluso si utilizar determinado n-grama consigue mejores resultados.

Para el caso del algoritmo de Naive-Bayes, podemos seleccionar el parámetro 'alpha' el cual es el número que se le suma al conteo de palabras para evitar tener un conteo nulo y que dé errores en el resultado final. Dicho parámetro no tiene por qué ser un entero, por lo que encontrar un valor óptimo puede resultar complicado. Una técnica usada para buscar un alpha óptimo es lo que se conoce como 'grid search'. Se prueban varios valores de alpha en determinado rango, equiespaciados y se va ajustando a aquellos que tienen mejores resultados. De ese modo se logra obtener el hiper-parámetro más adecuado.

A continuación se presenta la tabla con los hiper-parámetros elegidos en cada entrenamiento:

Ensayo	stop words	n-grama	idf	alpha
1	None	(1,2)	True	1.0
2	None	(1,1)	False	1.0
3	english	(1,1)	False	1.0
4	english	(1,1)	True	1.0

5	english	(1,2)	False	1.0
6	english	(1,1)	False	0.2
7	english	(1,1)	False	0.3
8	english	(1,1)	False	0.4
9	english	(1,1)	False	0.5

Tabla 8: Hiper-parámetros en cada ensayo

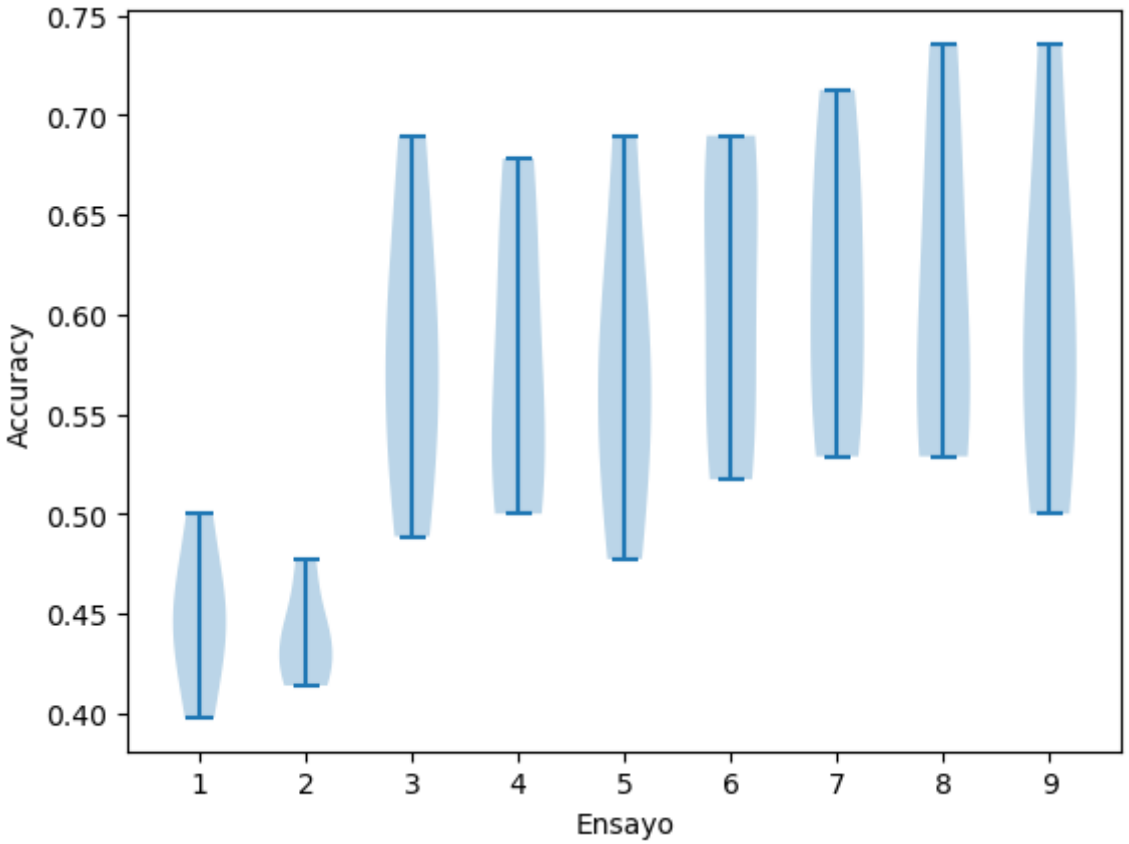

Figura 11: Diagrama de violín para los valores de accuracy de cada ensayo

Ensayo	Accuracy	Varianza
1	0.448	0.001126
2	0.438	0.000493
3	0.582	0.004611
4	0.575	0.004487
5	0.573	0.005019
6	0.610	0.004857
7	0.614	0.004461

8	0.610	0.006185
9	0.601	0.006519

Tabla 9: Valores promedio accuracy y varianza para los distintos modelos planteados

Observando los resultados, el ensayo 7 es el que mejor promedio de accuracy presenta y menor varianza, por lo que sería el mejor caso.

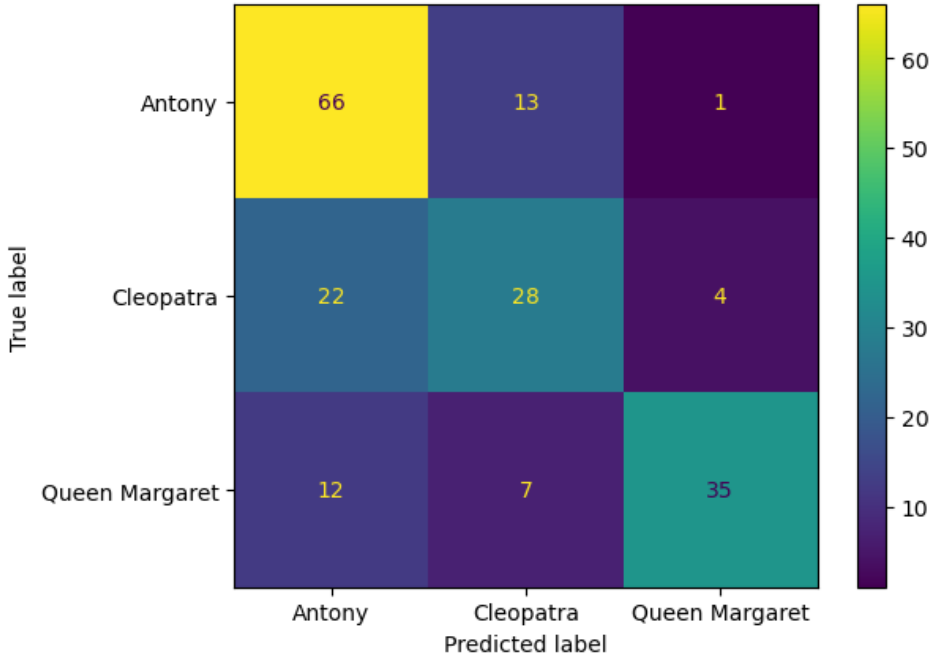

Figura 12: Matriz de confusión para el mejor caso de hiper-parámetros

Personaje	Precisión	Recuperación
Antony	0.66	0.82
Cleopatra	0.58	0.52
Queen Margaret	0.88	0.65

Tabla 10: Valores de Precisión y Recuperación

Para este caso, el valor de accuracy es de 0.69. Algo interesante es que el mejor resultado se obtuvo utilizando la metodología 'Bag of Words' utilizando un n-grama (1,1) la cual no contempla el órden de las palabras. Como ya se vió antes, el significado de los párrafos puede estar marcado por dicho órden, y esa información no se contempla a la hora de obtener una clasificación.

Modelo basado en Support vector machine (SVM) [5]

Se seleccionó el método SDGClassifier para ver si el mismo es más eficiente a la hora de predecir a qué personaje corresponde un párrafo determinado, en este método se seleccionó el hiper-parámetro 'loss' en el valor 'hinge', el cuál basa el algoritmo en SVM. El mismo crea hiperplanos que separan el conjunto de palabras por personaje, utilizando un

método de optimización para hallar los parámetros de dichos hiperplano a continuación, predice con la posición en el hiperespacio a que personaje pertenece cierto párrafo.

En este caso, el valor de accuracy es del 61%.

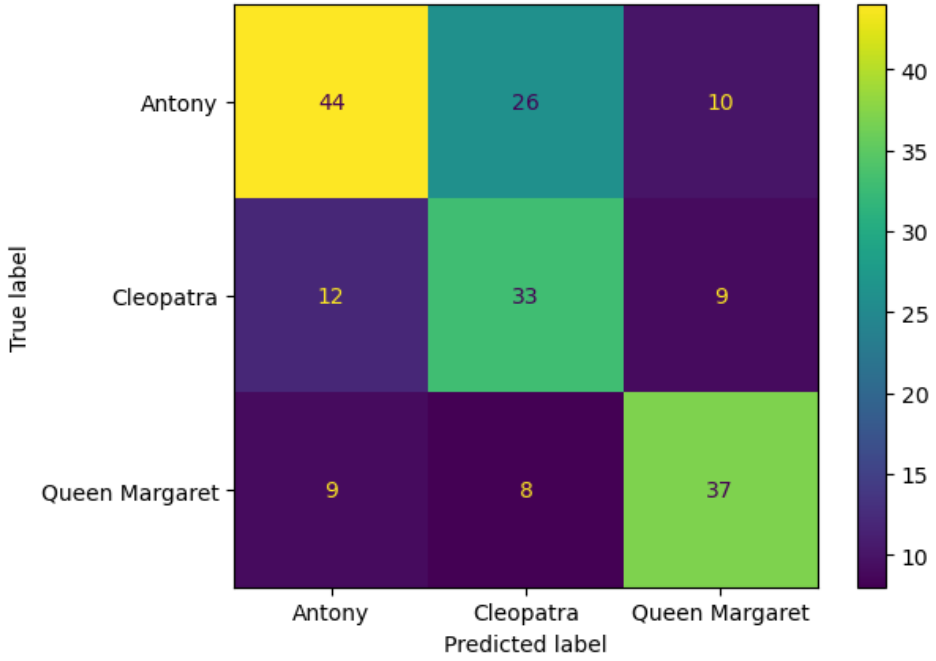

Figura 13: Matriz de confusión del modelo SDGClassifier

Personaje	Precisión	Recuperación
Antony	0.68	0.55
Cleopatra	0.49	0.61
Queen Margaret	0.66	0.69

Tabla 11: Valores de Precisión y Recuperación

Como se puede apreciar por los valores de Accuracy este modelo tuvo un desempeño menos eficiente que el modelo de Naive Bayes, aun así, en este modelo no se realizó ningún tipo de estudio con validación cruzada o hiper parámetros de forma de optimizar su desempeño.

Estudio de caso: Desbalance de párrafos

Para poder evaluar qué pasa si un personaje posee mayor cantidad de párrafos, o sea, la cantidad de datos por personaje es desbalanceada, se elige cambiar el personaje de Antony por uno que posee casi el doble de párrafos en la obra Falstaff. En la figura 14, se puede apreciar como el desbalance en número de párrafos entre el set de datos, apreciándose que el nuevo personaje posee la mitad de los párrafos del conjunto de datos.

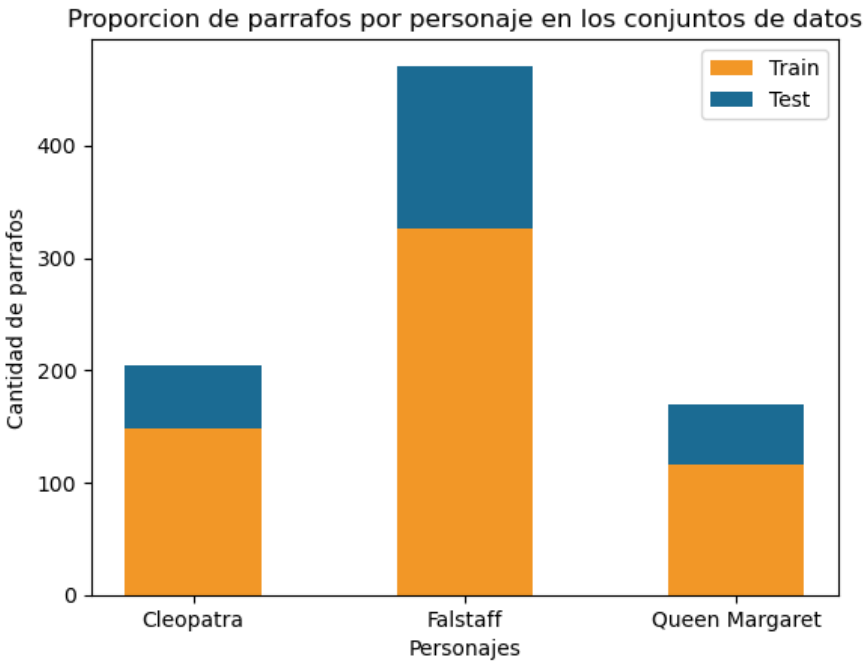

Figura 14: Cantidad de párrafos por personaje en el nuevo set de datos

Con el set de entrenamiento entrenamos el modelo Naive Bayes con parámetros óptimos Stopword='english', n-grama = (1,1), idf = False y alpha=0.3. El resultado es que el modelo predice con exactitud un 69% de los datos en el set de testeo. En la figura y tablas siguientes se pueden apreciar las métricas del modelo.

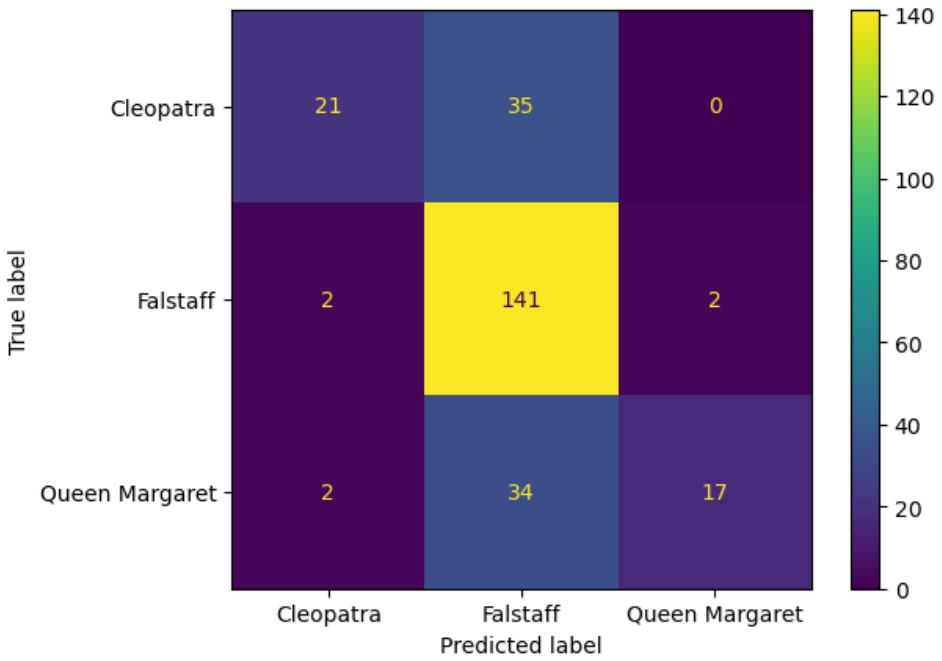

Figura 15: Matriz de confusión del modelo Naive Bayes con desbalance de datos

Personaje	Precisión	Recuperación
Cleopatra	0.84	0.52

Falstaff	0.67	0.97
Queen Margaret	0.89	0.32

Tabla 12: Valores de Precisión y Recuperación del modelo Naive Bayes con desbalance de datos

Si bien la accuracy del modelo es mejor, el mismo predice que mayoritariamente los párrafos pertenecen al personaje con mayor número de párrafos debido al desbalance de datos, no hace una buena distinción.

Submuestreo y sobremuestreo [6]

Para obtener buenos resultados, es conveniente tener los set de datos balanceados. Muchas veces no se consigue esto, por lo que se recurre a las técnicas de submuestreo y sobremuestreo.

El submuestreo se refiere a quitar datos de un grupo que contenga más que el resto. Si bien logra el cometido de balancear el set de datos, se podría perder información que puede ser importante a la hora de entrenar el modelo.

El sobremuestreo se trata de duplicar datos de la clase más disminuída hasta balancear el set. El problema que puede traer esta técnica es el sobreajuste del modelo a determinada característica de los datos duplicados.

Se podrían aplicar ambas técnicas al set de datos para obtener un resultado adecuado.

Word Embedding [7]

Otra manera de transformar palabras en vectores numéricos es la técnica Word Embedding o “incrustación de palabras” la misma se basa en asignar vectores numéricos a cada palabra. Esta asignación se realiza de tal manera que las palabras que tengan origen en un contexto similar puedan relacionarse entre ellas. Esta técnica se diferencia de otras en el sentido que representa frases y palabras en vectores con valores numéricos no binarios, es una representación densa de palabras en un espacio vectorial reducido. Por ejemplo, es esperable que la palabra “café” y “jugo” estén relativamente cerca, pero si miramos la palabra “edificio” estará más alejada de estas.

Existen tres técnicas principales de word embedding: Traditional word embedding, Static word embedding y Contextualized word embedding. Traditional word embedding se basa en la frecuencia y considera todo el documento para descubrir la significación de palabras raras en el documento, cuenta la ocurrencia de cada palabra y la coocurrencia de palabras Alumnos ejemplos son los ya usados, como *tf* e *idf*. Static word embedding es una predicción basada en la probabilidad de cada palabra y mapea cada palabra como un vector. Este método se denomina estático en el sentido que no altera el contexto una vez aprendido y la integración de las tablas no cambia entre diferentes oraciones. Por último, Contextualized word embedding se basa en el contexto de una palabra en particular, en este caso, palabras similares podrán tener representación de contexto contradictorio. Esta representación cambia dinámicamente basada en el contexto en que cada palabra aparece en el texto.

Es de esperar que esta técnica al considerar las relaciones sintácticas del lenguaje, mejore la predicción y el análisis de datos, permitiendo una mejor diferenciación entre los párrafos de cada personaje.

Fast text

Luego de entrenar un modelo de fast text con el set de datos, se consigue el siguiente resultado:

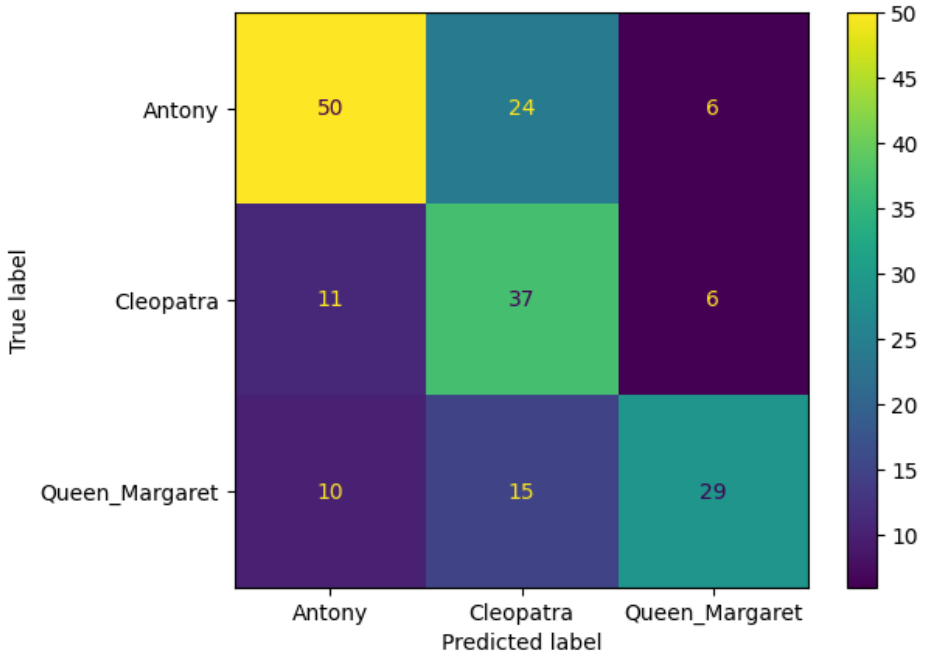

Figura 16: Matriz de confusión para el modelo fast text

Personaje	Precisión	Recuperación
Cleopatra	0.70	0.62
Falstaff	0.49	0.69
Queen Margaret	0.71	0.54

Tabla 12: Valores de Precisión y Recuperación del modelo Fast text

El valor de accuracy para este caso resultó 62%, lo que no mejora la predicción del modelo de Naive-Bayes con los mejores hiper-parámetros encontrados.

Podría ser de utilidad este método para trabajar con textos donde pueden aparecer palabras nuevas o raras, ya que al separar las palabras en partes, puede encontrar similitudes en palabras nunca vistas con palabras usadas en el entrenamiento.

Referencias

[1] Foundations of Statistical Natural Language Processing, Manning & Schütze, Mit Press, ISBN 0-262-13360-1, 1999

[2] (2023) Notas del Curso: “Introducción a la ciencia de datos”. Universidad de la república. Facultad de Ingeniería.

[3] René Vidal, Yi Ma, S.S. Sastry. Generalized Principal Component Analysis.

[4] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning with Applications in R. Second Edition. 2021.

[5] C.D. Manning, P. Raghavan and H. Schuetze (2008). Introduction to Information Retrieval. Cambridge University Press.

[6] R. Mohammed, J. Rawashdeh and M. Abdullah, "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results," 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2020, pp. 243-248, doi: 10.1109/ICICS49469.2020.239556.

[7] Selva Birunda, S., Kanniga Devi, R. (2021). A Review on Word Embedding Techniques for Text Classification. In: Raj, J.S., Iliyasu, A.M., Bestak, R., Baig, Z.A. (eds) Innovative Data Communication Technologies and Application. Lecture Notes on Data Engineering and Communications Technologies, vol 59. Springer, Singapore. https://doi.org/10.1007/978-981-15-9651-3_23