FACULTAD DE INGENIERÍA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

INTRODUCCIÓN A LA CIENCIA DE DATOS

Tarea - 2

AÑO 2024

GRUPO 14:

Fernando Sellanes Fajardo
Lucas Sellanes Fajardo

FECHA: 4 de julio de 2024

Tabla de contenidos

1. Introducción

La Tarea 2 de la asignatura se trata de una continuación de la Tarea 1, en la que se reutiliza la base de datos relacional abierta sobre las obras de Williams Shakespeare. El propósito de esta tarea es reforzar los conceptos clave de aprendizaje automático, centrándose en el proceso y la interpretación de resultados a nivel conceptual, sin ahondar en la optimización exhaustiva de las métricas de rendimiento.

1.1. Objetivos:

- Tomando un dataset reducido de sólo 3 personajes, generar un conjunto de entrenamiento y testeo estratificado que represente los párrafos correspondientes a los personajes estudiados.
- Transformar el texto del conjunto de entrenamiento a la representación numérica de conteo de palabras o *bag of words* y TF-IDF. Visualizar los datos realizando un análisis de las componentes principales (PCA).
- Entrenar un modelo Multinomial Naive Bayes y reportar métricas de desempeño como *accuracy*, *precision* y *recall*.
- Implementar una técnica de validación cruzada para optimizar los hiper-parámetros.
- Evaluar y comparar diferentes modelos de clasificación de texto.
- Evaluar el impacto de cambiar personajes en el conjunto de datos y visualizar el (des)balance de datos utilizando el PCA.
- Buscar información sobre una técnica alternativa de extraer *features* de texto.

Estos objetivos están diseñados para proporcionar una comprensión integral del proceso de análisis y modelado de datos de texto, desde la limpieza y preparación de los datos hasta la evaluación y comparación de modelos predictivos.

2. Dataset y representación numérica del texto

2.1. Dataset de entrenamiento-testeo

A partir de lo desarrollado en la Tarea 1, se toma la función `clean_text()` desarollada para limpiar los signos de puntuación del cuerpo de texto de los párrafos. Una vez limpios los párrafos, se procede a tomar los cuales corresponden a los siguientes personajes de la obra de Shakespeare:

- **Antony:** También conocido como Marco Antonio es uno de los personajes principales en la obra *"Antony and Cleopatra"* de William Shakespeare. Es un general romano y uno de los triunviros que gobiernan Roma tras el asesinato de Julio César. Antonio es un hombre apasionado y valiente, conocido tanto por su destreza en el campo de batalla como por su carisma político. Su relación amorosa con Cleopatra, la reina de Egipto, es central en la trama y eventualmente lleva a su caída.
- **Cleopatra:** Es la reina de Egipto y una figura poderosa en la obra *"Antony and Cleopatra"*. Es una mujer de gran belleza e inteligencia, capaz de manipular y seducir a quienes la rodean. Su amor por Antony es profundo, pero también complicado por su naturaleza política y estratégica. Cleopatra es una líder astuta que busca mantener su poder y la independencia de Egipto frente a Roma.
- **Queen Margaret:** Es un personaje prominente en varias de las obras de Shakespeare, específicamente en las trilogías de "Henry VI" y "Richard III". Es la esposa del Rey Enrique VI de Inglaterra y madre del Príncipe Eduardo. Margarita es una figura poderosa y a menudo despiadada, conocida por su valentía y ferocidad en la lucha por el poder durante las Guerras de las Rosas.

Con estos tres personajes se separan los datos en dos conjuntos *train-test* (70-30 %) para entrenar al modelo. Es importante en esta instancia realizar una correcto separado de los datos que no induzca problemas al entrenar al modelo como sesgos, por ello se realiza un muestro estratificado a modo de asegurar que todos los subgrupos estén adecuadamente representados.

En la Fig. 1 se presenta una visualización donde se observa que los datos de entrenamiento y testeo son representativos de la participación de los personajes en las obras de Shakespeare.

2.2. Representación numérica del texto

El siguiente paso de procesamiento del dataset es convertirlo en una representación numérica del conjunto de texto que permita trabajar al modelo predictivo. En este trabajo se utilizan dos técnicas para el proceso de *features* de texto.

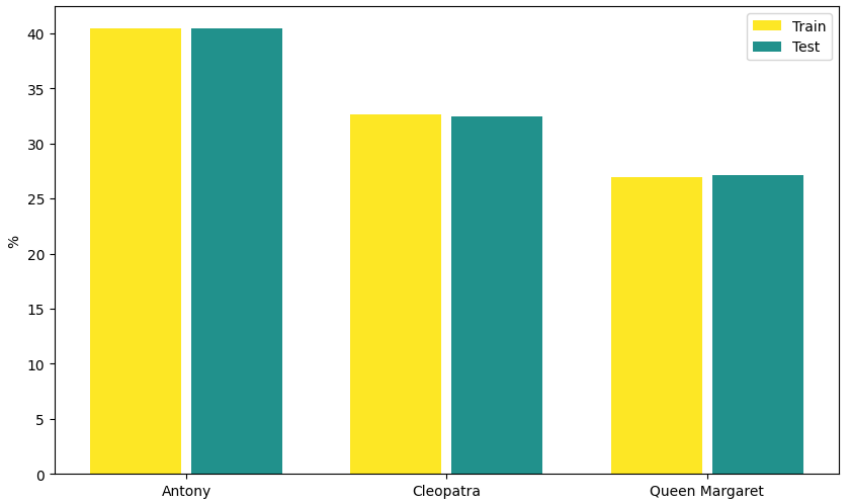


Figura 1: Balance de párrafos de cada personaje para los dataset de entrenamiento y testeo.

2.2.1. Conteo de palabras o *bag of words*

La técnica de conteo de palabras o *baf of words* se basa en un método de representación numérica de texto para ser utilizado en el procesamiento del lenguaje natural. Se basa en tres pasos:

1. Dividir el texto en **tokens** definidas como palabras individuales o conjuntos de palabras.
2. Crear un conjunto de todos los **tokens** definidos en los documentos.
3. Vectorizar cada documento como vectores de frecuencias de palabras.

A modo de ejemplo, esto se puede utilizar para clasificar comentarios como positivos o negativos. Por ejemplo, sí las reseñas de un local son las siguientes:

1. Documento 1: “Me encanta este producto, es fantástico.”
2. Documento 2: “Este es el peor producto que he comprado.”
3. Documento 3: “Producto increíble, lo recomiendo a todos.”

El vocabulario del modelo será: [Me, encanta, este, producto, es, fantástico, peor, he, comprado, increíble, recomiendo, todos]. Y entonces la matriz devuelta del texto vectorizado es la siguiente:

Palabra	Doc 1	Doc 2	Doc 3
Me	1	0	0
encanta	1	0	0
este	1	1	1
producto	1	1	1
es	1	1	0
fantástico	1	0	0
peor	0	1	0
he	0	1	0
comprado	0	1	0
increíble	0	0	1
recomiendo	0	0	1
todos	0	0	1

Al utilizar el método de conteo de palabras, la matriz resultante se trata de una *sparse matrix* de tamaño documentos × tokens. Estos tipos de matrices se encuentran repletas de valores ceros dado que en los documentos de texto las palabras en general son diferentes entre sí, siendo una ventaja a la hora de operar con ellas utilizando herramientas que realicen operaciones con matrices ignorando las entradas con cero. De esta forma, permite un uso eficiente de la memoria y una ejecución rápida de las operaciones matriciales.

A la hora de representar el texto con este método es interesante utilizar los *n-gramas* para darle contexto a las oraciones. Los *n-gramas* son una forma de representar los documentos relacionando los elementos (palabras) consecutivos. De esta forma se pueden utilizar combinaciones de 1 a n palabras y no perder el contexto.

En el ejemplo anterior el nuevo vocabulario al utilizar unigramas (1,1) o bigramas (2,2) queda de la siguiente forma:

- Vocabulario utilizando unigramas (1,1): [Me, encanta, este, producto, es, fantástico, peor, he, comprado, increíble, recomiendo, todos]
- Vocabulario utilizando bigramas (2,2): [Me encanta, encanta este, este producto, producto es, es fantástico, este es, es el, el peor, peor producto, producto que, que he, he comprado, producto increíble, increíble lo, lo recomiendo, recomiendo a, a todos]

2.2.2. *Term Frecuency - Inverse Document Frecuency*

La técnica *Term Frecuency - Inverse Document Frecuency* (TF-IDF) se trata de una transformación numérica utilizada para el procesamiento de lenguaje natural para evaluar la importancia de una palabra o conjunto de palabras (en el caso de utilizar *n-gramas*) en un documento dentro del conjunto de los documentos. En esta técnica se mide la frecuencia del término (TF) como el número de veces que el termino aparece en el documento sobre el número total de documentos y la frecuencia inversa de documentos (IDF) como el logaritmo del total de documentos sobre el número de documentos donde aparece este término.

El TF-IDF es el producto de estos cálculos y pondera las palabras para reducir la importancia de las palabras comunes del idioma en el dataset.

2.3. Análisis de los Componentes Principales (PCA) al conjunto de entrenamiento

A continuación se busca visualizar el comportamiento de los primeros componentes principales de los conjuntos de entrenamiento frente a los procesamientos de texto explicados anteriormente y al filtrado del texto utilizando *stop words*. Para esto se definen 8 combinaciones de los anteriores modificando la complejidad del procesamiento:

1. Sin IDF, sin *stop words*, unigrama (1,1).
2. Sin IDF, con *stop words*, unigrama (1,1).
3. Sin IDF, sin *stop words*, bigrama (1,2).
4. Sin IDF, con *stop words*, bigrama (1,2).
5. Con IDF, sin *stop words*, unigrama (1,1).
6. Con IDF, con *stop words*, unigrama (1,1).
7. Con IDF, sin *stop words*, bigrama (1,2).
8. Con IDF, con *stop words*, bigrama (1,2).

Aplicando este procesamiento se obtienen diferentes *sparse matrix* de tamaño 438×2613 y 438×8510 cuando se utiliza el unigrama y el bigrama respectivamente. Esto indica que los documentos son 438 en el conjunto de entrenamiento y que para el caso del unigrama se identifican 2613 palabras únicas pero que cuando se toman también las combinaciones de dos palabras, estas aumentan a 8510.

Cada uno de estos dataset de entrenamiento es transformado para obtener los primeros dos componentes principales y poder realizar una visualización en un espacio bidimensional. En la Fig. 2 se presentan las primeras 2 componentes principales para cada una de las combinaciones. De esta se desprende que a medida que se complejiza la transformación realizada los primeros PCA tienden a capturar cierta variación de los datos, observándose que en la última combinación (con IDF, con *stop words*, bigrama (1,2)) los primeros componentes parecen asimilar la variación de una mejor forma.

Sin embargo, el uso de sólo dos componentes principales no permite clasificar a los personajes, posiblemente porque estos personajes hablan el mismo idioma y con una formalidad propia de la época sin tecnicismos característicos de cada uno.

Considerando que las dos primeras componentes no logran capturar suficiente información se analiza como varía la varianza explicada a medida que se incorporan más componentes para la parametrización 8 (Fig. 3). Este análisis nos devuelve que la varianza explicada aumenta un poco menos de 1 % por cada componente que se

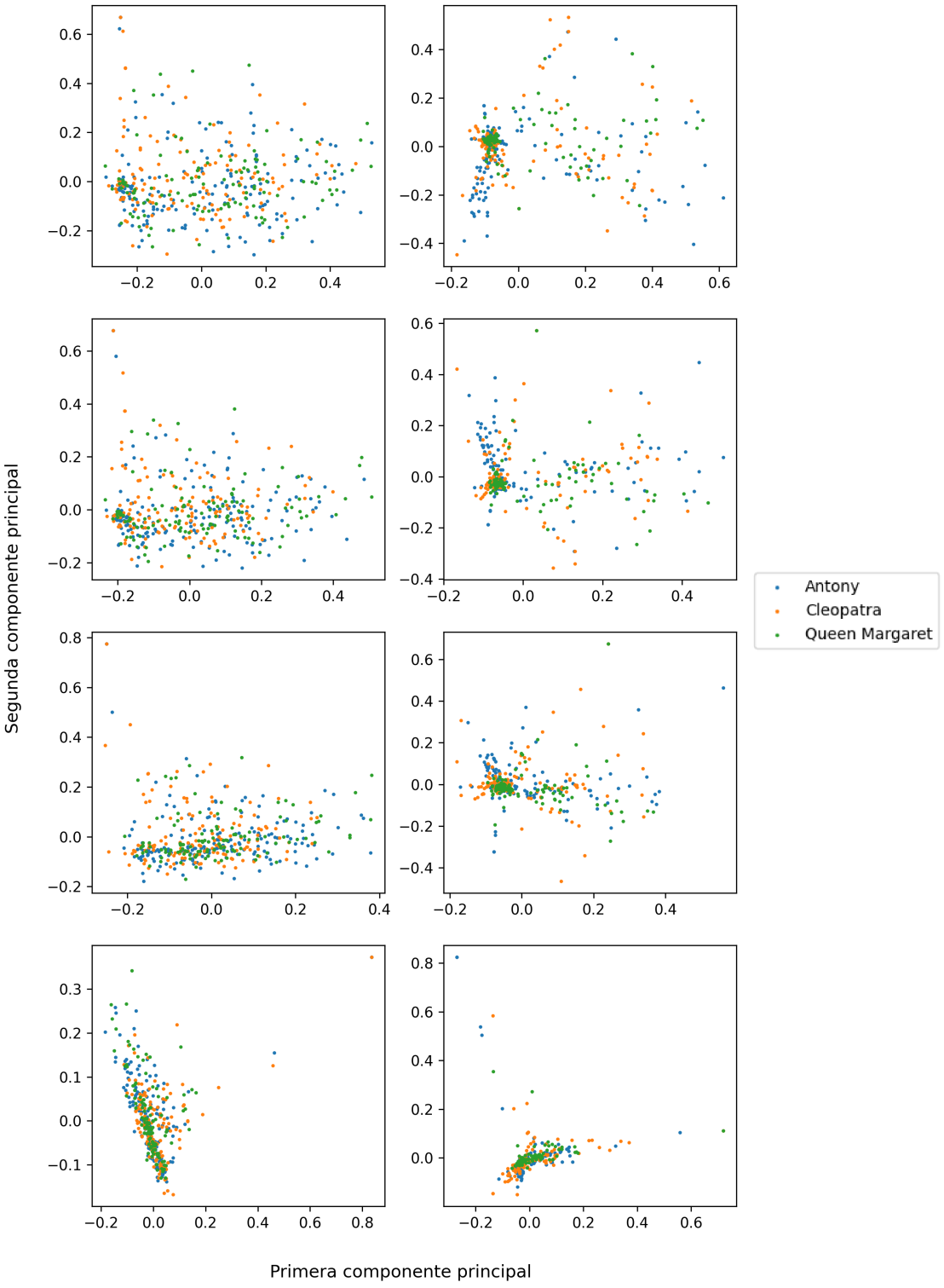


Figura 2: Primeras dos componentes principales para las 8 combinaciones de procesamiento de los dataset de entrenamiento.

agrega, sugiriendo que la información relevante que permita la separación de los personajes está distribuida a lo largo de varias dimensiones.

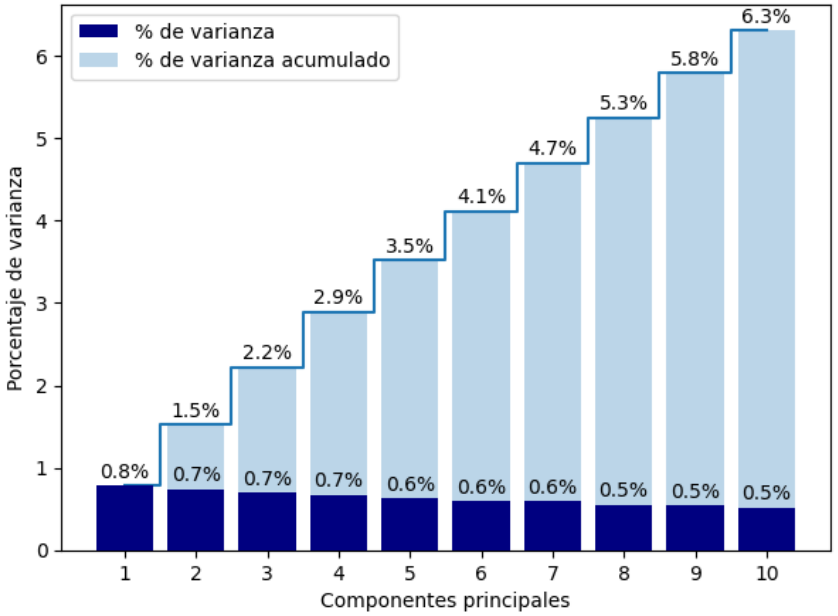


Figura 3: Varianza explicada con el acumulado de los componentes principales para la parametrización 8.

3. Entrenamiento y evaluación de modelos

3.1. Modelo Multinomial Naive Bayes

En esta tarea se propone entrenar el modelo Multinomial Naive Bayes para predecir sobre el conjunto de testeo. Este modelo se trata de un algoritmo de aprendizaje supervisado utilizado para la clasificación de texto y datos categóricos, basándose en el teorema de Bayes para calcular la probabilidad de que un dato pertenezca a una clase, dadas las características.

Para esto, se toma el conjunto de entrenamiento al cual se le aplica el filtrado con las *stop words* del ingles y la transformación TF-IDF separando el texto en bigramas (1,2). A este conjunto se lo entrena utilizando el modelo y se realizan las predicciones sobre los datos de validación. Para evaluar el desempeño del modelo se calculan las siguientes métricas y la matriz de confusión:

- ***accuracy:*** es la proporción de predicciones correctas contra el total de predicciones realizadas.

$$accuracy = \frac{\text{Número de predicciones correctas}}{\text{Número total de predicciones}} \tag{1}$$

- ***precision:*** es la relación de los verdaderos positivos contra los que se predicen positivos.

$$precision = \frac{V_P}{V_P + F_P} \tag{2}$$

- ***recall:*** es la proporción de verdaderos positivos que fueron correctamente clasificados como positivos.

$$recall = \frac{V_P}{V_P + F_N} \tag{3}$$

- **Matriz de confusión:** es utilizada para evaluar el rendimiento del modelo visualizando los aciertos y errores de las predicciones. Cada fila de la matriz representa las instancias verdaderas y cada columna las instancias predichas.

Es importante tener en cuenta varias de las métricas a la hora de evaluar el desempeño de un modelo de predicción y no tomar la *accuracy* como única. Por ejemplo, cuando se tiene un problema de desbalance de clases, si los datos de entrenamiento están desbalanceados, se puede lograr una *accuracy* alta al predecir siempre la clase mayoritaria. A su vez, cuando se desea estudiar a las minorías, la *accuracy* no lo representa de buena forma. Por ello, otras métricas como el *recall* y la *precision* toman importancia al ser indicadores específicos de algunos rendimientos.

Una vez entrenado el modelo con las parametrizaciones tomadas, en la Tabla 1 se presentan las métricas obtenidas diferenciando en personajes y en la Fig. 4 la matriz de confusión para los personajes.

En la matriz de confusión se observa que el modelo quedó correctamente entrenado para identificar cuando un párrafo está dicho por Antony, pero esto generó que a

	Antony	Cleopatra	Queen Margaret
Accuracy	0,54		
Precision	0,49	0,56	0,9
Recall	0,87	0,3	0,35

Tabla 1: Métricas para evaluar el modelo.

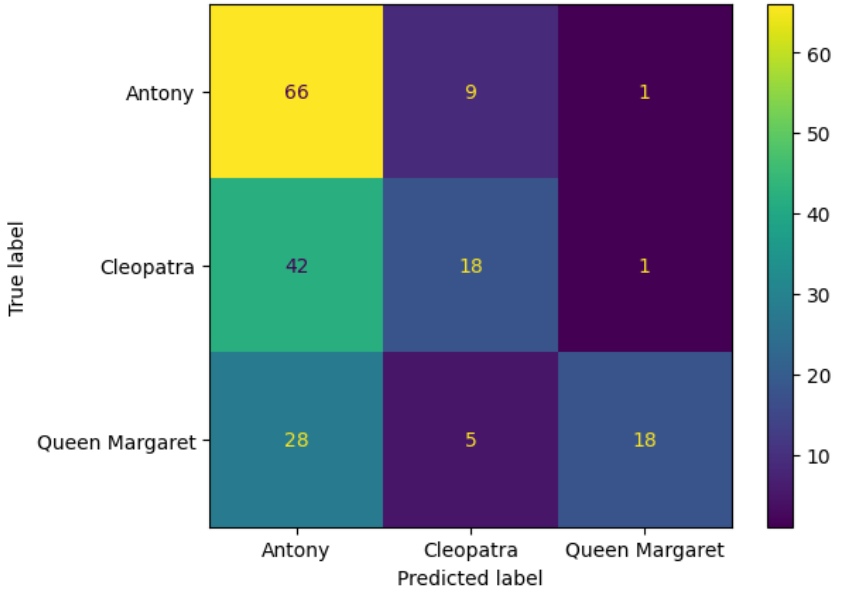


Figura 4: Matriz de confusión para el modelo Multinominal Naive Bayes del conjunto de testeo.

gran parte de los párrafos pertenecientes a Cleopatra y Queen Margaret también se los adjudique a Antony. Esto se debe a que el modelo se ve desbalanceado por existir una mayor cantidad de párrafos de Antony en el cuerpo de entrenamiento. A su vez, por ser los tres personajes de la época y con similares características de poder, es posible que sus párrafos también tengan similares características y el modelo se los adjudique para el que mayormente se encuentra entrenado.

Asimismo, se observa un comportamiento similar con las métricas de *precision* y *recall*. Para el personaje de Queen Margaret se registra una *precision* muy alta, esto es debido a que el modelo las veces que predijo que podía ser Queen Margaret acertó un 90 %, sin embargo el *recall* de 35 % nos indica que de los párrafos que realmente pertenecían a este personaje, predijo un bajo porcentaje. De manera opuesta sucede con Antony, dado que el modelo tiene una *precision* baja por predecir muchas veces Antony y que este párrafo no sea de él, y un *recall* alto por predecir correctamente los verdaderos positivos.

3.2. Técnica de validación cruzada

Las técnicas de validación cruzada son utilizadas para evaluar la capacidad lograr mejores resultados de independizándose del conjunto utilizado para entrenamiento y

validación. Estas técnicas se basan en dividir el conjunto de datos de entrenamiento en diferentes particiones (*folds*) como se observa en la Fig. 5 y entrenar múltiples veces el modelo utilizando estos diferentes *folds* como conjuntos de validación y los restantes de entrenamiento. Para cada una de estas validaciones se calculan las métricas y se estima de forma general el desempeño de los modelos.

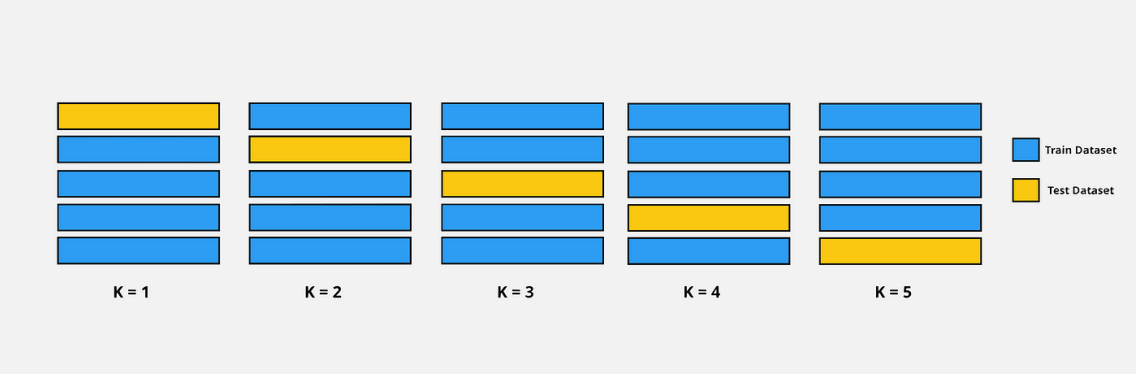


Figura 5: Ejemplo de separación del conjunto de datos en *k-folds*.

En esta tarea se utiliza una técnica de validación cruzada estratificada para identificar hiper-parámetros, buscando resultados más generales del modelo. En esta técnica se garantiza que la separación de los *folds* tenga la misma distribución de las clases, en este caso personajes.

Se setean las mismas 8 combinaciones de parámetros utilizados anteriormente para evaluar la variación de los primeros dos componentes principales. Además se define realizar la validación cruzada con 4 *folds* de datos para cada seteo de parámetros. En la Fig. 6 se presenta un gráfico de violín con la *accuracy* para las 8 combinaciones.

La visualización permite observar diferentes aspectos del modelo, como que el filtrado según las *stop words* del inglés mejora el rendimiento general, evitando que el modelo le de importancia a las palabras sin contenido semántico. Por otro lado, no se observan grandes diferencias en las métricas al utilizar las parametrizaciones de TF-IDF o los *n-gramas*.

Esta técnica de validación cruzada destaca los casos en los que el conjunto de validación elegido te influye en el entrenamiento del modelo, un aspecto que no es buscado al entrenar los modelos. En este caso, cuando se toman la quinta combinación de parámetros (con IDF, sin stop words, unigrama (1,1)), la *accuracy* presenta una desviación standard muy grande que indica que la elección del conjunto de validación es relevante.

Finalmente, se utiliza el *accuracy* como métrica para determinar cual es el mejor modelo para rentrenarlo sobre todo el conjunto de entrenamiento disponible. De esta forma se adoptan los siguientes hiper-parámetros con una *accuracy* promedio del 58,7 %:

`Sin IDF, con stop words, unigrama (1,1)`

Una vez rentrenado el modelo se utiliza el conjunto de testeo para evaluarlo de igual

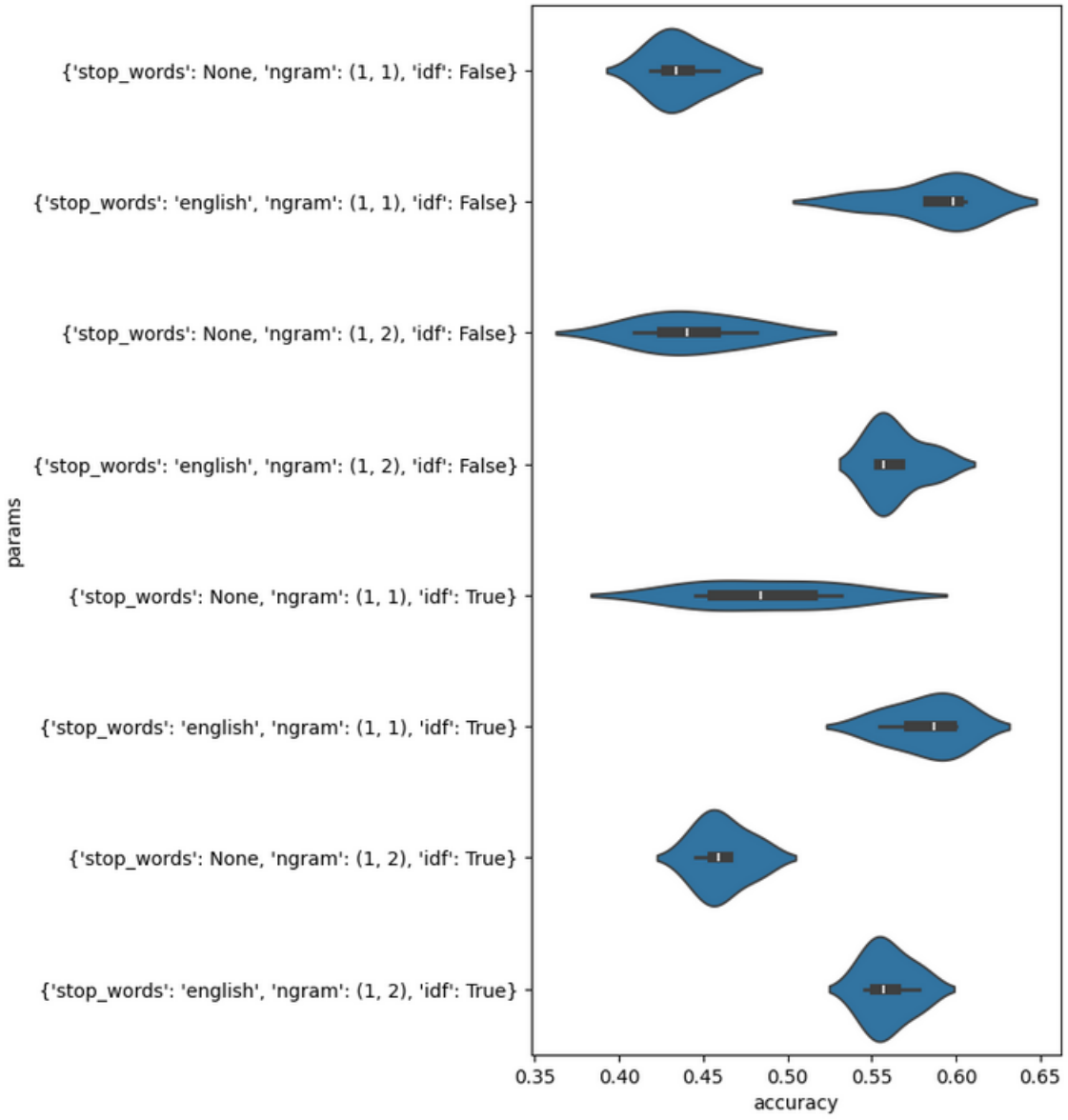


Figura 6: *Accuracy* obtenida para la validación cruzada realizada con los 8 sets de parámetros.

forma que con el modelo anterior. En la Tabla 2 y en la Fig. 7 se presentan las métricas obtenidas. Como era esperable al visualizar el gráfico de violín no se observa una gran mejora frente al modelo anterior entrenado, observándose los mismos inconvenientes.

	Antony	Cleopatra	Queen Margaret
Accuracy	0,56		
Precision	0,51	0,57	0,84
Recall	0,86	0,33	0,41

Tabla 2: Métricas de evaluación del modelo entrenado utilizando las técnicas de validación cruzada.

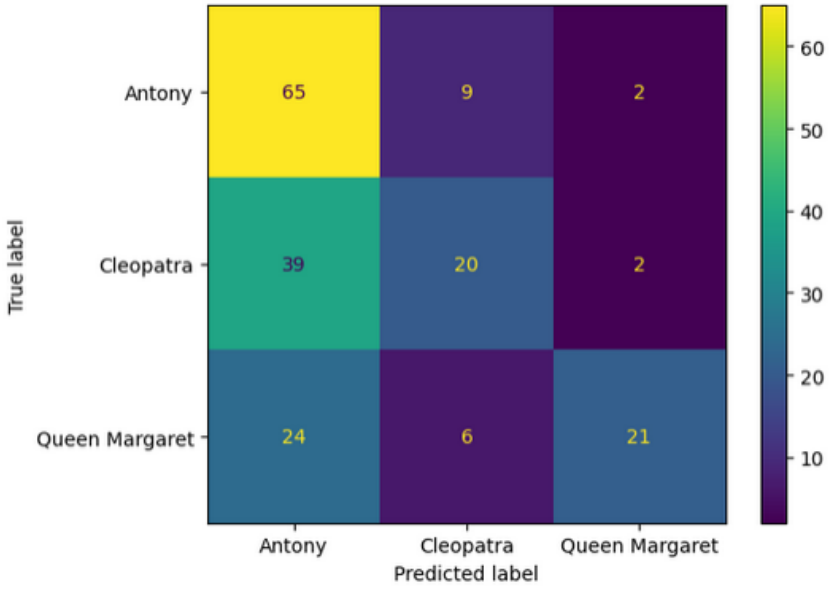


Figura 7: Matriz de confusión para el modelo Multinominal Naive Bayes entrenado utilizando las técnicas de validación cruzada.

3.3. Otro modelo de Procesamiento de Lenguaje Natural (NPL)

Para la evaluación de otro modelo de procesamiento de lenguaje natural, se tomó en cuenta las *Supported Vector Machines* (SVM). Estas son un tipo de algoritmo de aprendizaje supervisado que es utilizado principalmente para tareas de clasificación y su objetivo principal es encontrar un hiperplano en un espacio multidimensional que divida las diferentes clases de datos de manera óptima como se observa en el ejemplo de la Fig. 8.

Ventajas:

- Eficiente en espacios de alta dimensión.

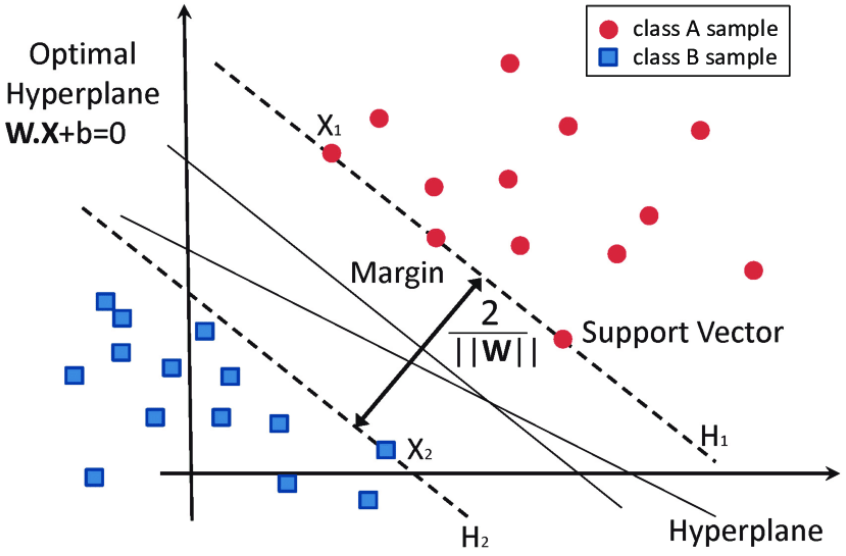


Figura 8: Ejemplo de separación en un espacio multidimensional por un plano.

- Eficaz incluso cuando el número de dimensiones es mayor que el número de muestras.
- Uso eficiente de la memoria.

Desventajas:

- No funciona bien con conjuntos de datos grandes debido a su alta complejidad computacional.
- Requiere una cuidadosa selección y ajuste de los parámetros y del kernel.
- Sensible al ruido en los datos, ya que los vectores de soporte determinan el hiperplano.

Para implementar el modelo con el dataset, se realiza el mismo procedimiento que para el modelo Multinominal Nayves Bayes, entrenando este utilizando técnicas de validación cruzada de separación en 4 *folds*. Utilizando esta técnica, el conjunto de *train* se obtuvo una *accuracy* promedio de 59,8 %.

A continuación, en la Tabla 2 se presenta las métricas obtenidas comparando la predicción resultante del modelo del conjunto *test*, así como también la matriz de confusión asociada en la Fig. 9.

	Antony	Cleopatra	Queen Margaret
Accuracy	0,585		
Precision	0,57	0,54	0,73
Recall	0,74	0,49	0,47

Tabla 3: Métricas de evaluación del conjunto *test* del modelo SVM entrenado utilizando las técnicas de validación cruzada.

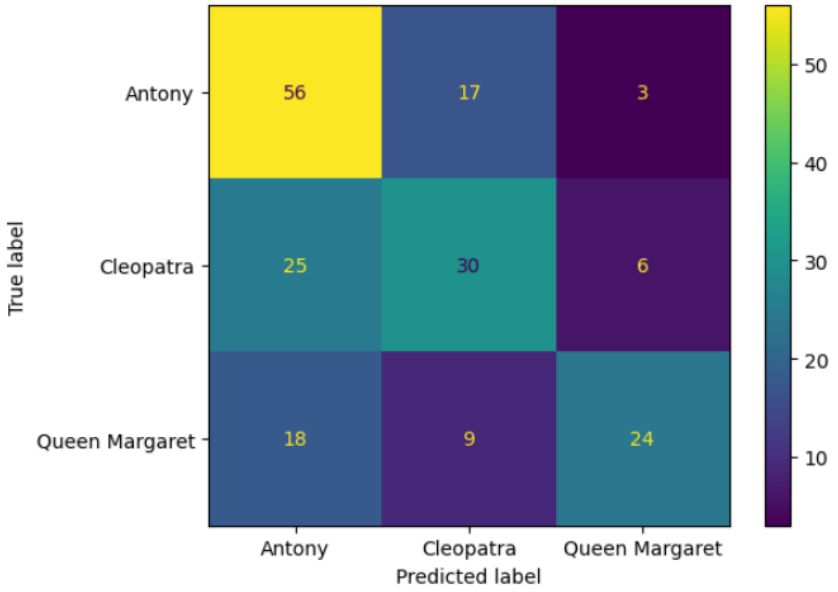


Figura 9: Matriz de confusión para el modelo SVM del conjunto de testeo.

Comparando las métricas expuestas en las tablas 2 y 3 se puede concluir que, SVM tiende a tener mejor rendimiento en términos de *recall* para las clases Cleopatra y Queen Margaret. MNB muestra una *precision* superior en las clases Cleopatra y Queen Margaret, y una mejor *recall* para la clase Antony.

En el caso de que se buscara elegir cual modelo es mejor, esta decisión depende de la métrica que se considere más importante para la tarea específica de clasificación. Si se prioriza el *recall*, SVM puede ser la mejor opción. Si se prioriza la *precision*, MNB podría ser más adecuado para ciertas clases.

3.4. (Des)balance de datos

A continuación, se observa que sucede cuando en el dataset hay una clase la cual se encuentra exageradamente en desbalance frente a las demás. Para ello se quita el personaje de Queen Margaret y se colocan las direcciones del director para utilizar una clase que ni siquiera se considera un personaje, por lo que sus párrafos no son diálogos. Como se observa en la Fig. 10 ahora el conjunto de datos se encuentra totalmente desbalanceado obteniendo un conjunto de entrenamiento y testeo que tiene más de un 90 % de esta clase.

Para visualizar que efectos tiene esto en los datos se grafican las primeras 2 componentes principales para la parametrización más básica (sin IDF, sin *stop words* del inglés y utilizando unigramas (1,1)) y con la mayor posibilidad de filtrado y procesamiento (con IDF, con *stop words* del inglés y utilizando bigramas (1,2)). Estos se observan en la Fig. 11 donde la parametrización más compleja se ve ordenada en tres direcciones. Si bien estas tres direcciones no separan las clases, es esperable que estas primeras componentes tengan una fuerte influencia en explicar las 3 clases.

Para observar esto, se grafica la varianza explicada para los 10 primeros componentes

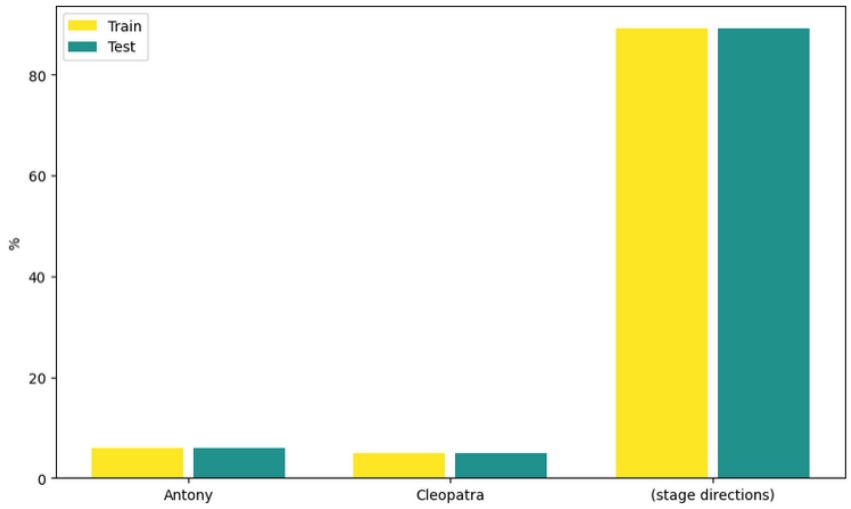


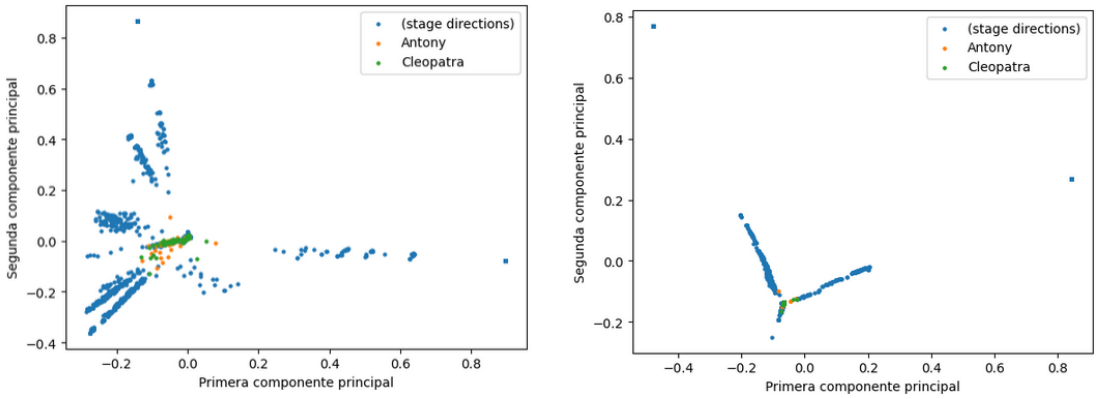
Figura 10: Balance de clases en entrenamiento y testeo cambiando el subconjunto de personajes.

principales en el caso de la parametrización más compleja en la Fig. 12. A diferencia de la primera elección de personajes, cuando se incorporan las direcciones del director en el dataset las dos primeras componentes principales toman gran importancia respecto a las demás, acumulando cerca de un 20 % de la varianza.

3.5. Técnicas alternativas de extracción de *features* de texto

Un ejemplo de técnicas de extracción de *features* de texto es el uso de *word embeddings* para representar el vocabulario del cuerpo de entrenamiento como vectores de un conjunto de baja dimensión. En esta técnica las palabras se agrupan por palabras con significados similares según relaciones semánticas y de sintáxis.

Otro ejemplo es el uso de la técnica *Named Entity Recognition*, donde se identifican diferentes entidades del texto como nombres de personas, lugares, fechas, entre otros.



(a) Parametrización más básica.

(b) Parametrización más compleja.

Figura 11: Primeras dos componentes principales.

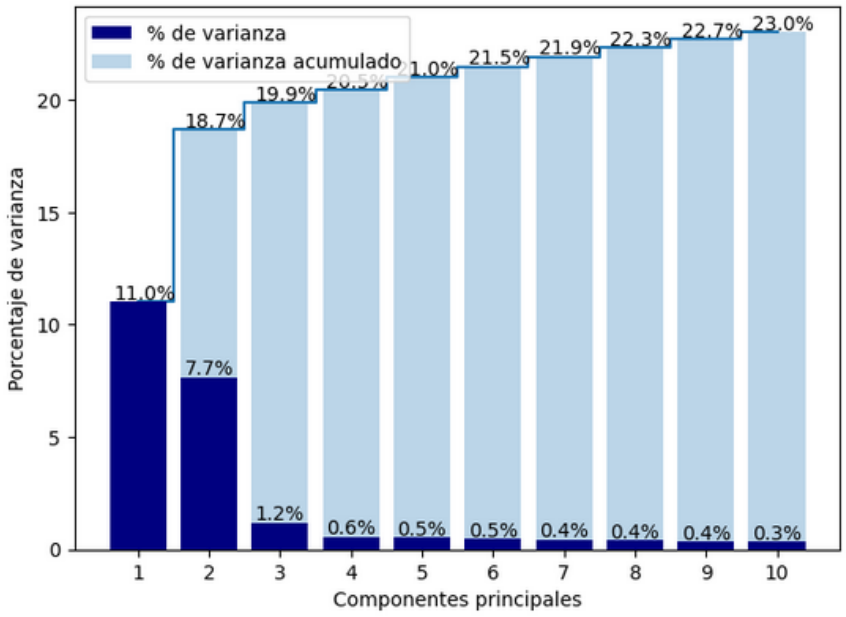


Figura 12: Varianza explicada con el acumulado de los componentes principales para la parametrización más compleja.

Para ambas técnicas se espera que al utilizarlas los resultados mejoren dado que se prioriza el contexto de las palabras en los párrafos y no se entrena el modelo a partir de la frecuencia de aparición de las palabras solamente.

Asimismo, la identificación de entidades puede ayudar a mejorar particularmente en esta tarea con la predicción de personajes como Queen Margaret frente a Antony y Cleopatra. Esto esperable ya que estos personajes se encuentran en obras diferentes en las cuales las entidades pueden ser diferentes. Sin embargo, también podría afectar a la predicción de Antony y Cleopatra por compartir la misma historia y entonces confundir ambos personajes.