Movie Genre Prediction bajo metodología CRISP

[ac] Camilo Andrés González Vargas, [ac] Juan David Valencia Sandoval, [ac] Paula Alejandra Velasco Cuberos, [ac] Pavel Mauricio Dussan Gutiérrez, [ac] Tatiana Roa Ahumada, [ac] Yuri Angélica Reina Guzmán, [bc] Sergio Alberto Mora Pardo

a. Estudiante de Maestría en Analítica para la Inteligencia de Negocios
b. Profesor, Departamento de Ingeniería Industrial
c. Pontificia Universidad Javeriana, Bogotá, Colombia

BUSINESS UNDERSTANDING

Background:

En la industria del entretenimiento, la capacidad de clasificar películas en géneros correctos es crucial para la organización de los catálogos en plataformas de streaming y servicios de distribución digital. Plataformas como Netflix, Amazon Prime, y Disney+ ofrecen miles de títulos, lo que hace indispensable el uso de herramientas automáticas para clasificar el contenido y facilitar la navegación de los usuarios. Los géneros de películas no solo permiten a los espectadores identificar el tipo de contenido que prefieren ver, sino que también ayudan a las plataformas a ofrecer recomendaciones más precisas y personalizadas basadas en los patrones de consumo de los usuarios.

El volumen de datos audiovisuales que generan estas plataformas, junto con el hecho de que las películas pueden pertenecer a múltiples géneros simultáneamente, ha llevado al desarrollo de sistemas automatizados de categorización. Este escenario multietiqueta plantea desafíos técnicos, ya que las películas pueden abarcar varios géneros, y se requiere predecir simultáneamente todas las etiquetas correctas en función de la trama (González, 2024). La clasificación automática de géneros de películas basada en la trama proporciona una ventaja competitiva significativa para estas plataformas, ya que permite mejorar la precisión de los sistemas de recomendación.

Al utilizar modelos de aprendizaje automático es posible analizar las descripciones de las tramas y predecir a qué géneros podría pertenecer una película. Estos modelos son particularmente efectivos en este tipo de tareas de clasificación, ya que pueden aprender patrones importantes a partir de los datos textuales.

En el presente proyecto, la clasificación de géneros a partir de la trama de la película tiene como objetivo no sólo automatizar la organización de catálogos, sino también mejorar la experiencia del usuario. Una clasificación precisa permite a los usuarios descubrir contenido relevante de manera más rápida y eficaz, lo que aumenta su satisfacción con el servicio. Además, una recomendación de películas más precisa y personalizada puede incrementar el tiempo que los usuarios pasan en la plataforma, lo que resulta en una mayor fidelización y retención de clientes. El análisis de tendencias de consumo y la clasificación automática de contenido a gran escala también pueden contribuir al análisis del mercado cinematográfico, mejorando la toma de decisiones comerciales para las plataformas (González, 2024).

Determine Business objectives

Business goal:

BG1: Facilitar la selección de contenido para optimizar el descubrimiento de películas basadas en las búsquedas y preferencias del usuario, mejorando la precisión de las recomendaciones y personalizando la oferta de contenido para aumentar la satisfacción y el tiempo de interacción en la plataforma.

Business success criteria:

BG1 – KPI1: Precisión en el proceso de clasificación del contenido.

Determine Data mining goals

Data mining goal:

DMG1: Desarrollar un modelo de clasificación que prediga la probabilidad de que una película pertenezca a un género a partir de la trama.

Data mining success criteria:

DMG 1 – KPI1: Obtener un AUC mínimo de 0.89 en entrenamiento.

DATA UNDERSTANDING

Se cuenta con dos conjuntos de datos, un conjunto de entrenamiento y uno de prueba. El conjunto de datos de entrenamiento contiene 7,895 y el de prueba 3,383 observaciones de películas y está compuesto por cinco columnas que ofrecen información clave sobre cada película. Estas son: el año de lanzamiento, el título en formato de texto, la descripción textual de la trama de cada película, la lista de géneros asociados a cada película y la calificación de la película, que está representada como un valor numérico de tipo flotante. Es posible visualizar estas columnas en la tabla 1, la cual muestra el ejemplo de 5 de las 7,895 películas que conforman el conjunto de datos.

year	title	plot	genres	rating
2003	Most	most is the story of a single father who takes...	['Short', 'Drama']	8.0
2008	How to Be a Serial Killer	a serial killer decides to teach the secrets o...	['Comedy', 'Crime', 'Horror']	5.6
1941	A Woman's Face	in sweden , a female blackmailer with a disfi...	['Drama', 'Film-Noir', 'Thriller']	7.2
1954	Executive Suite	in a friday afternoon in new york , the presi...	['Drama']	7.4
1990	Narrow Margin	in los angeles , the editor of a publishing h...	['Action', 'Crime', 'Thriller']	6.6

Tabla 1. Columnas del conjunto de datos de entrenamiento

En total se identifican 24 géneros en el set de datos, en dónde se demuestra que el drama y la comedia son líderes con una amplia diferencia frente a los demás géneros.

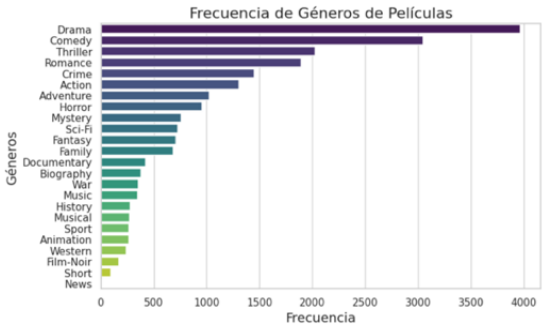


Figura 1. Películas por género

Se puede observar un aumento en la producción de películas en las últimas décadas, lo que refleja una tendencia general creciente en la industria del cine. Es posible que haya años específicos con picos significativos, lo que podría coincidir con tendencias culturales o el auge de ciertas franquicias cinematográficas.

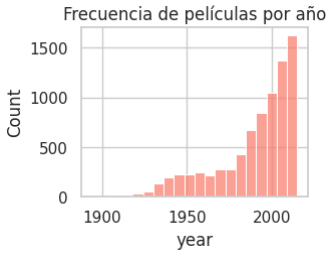


Figura 2. Frecuencia de películas por año

La distribución de calificación parece ser relativamente normal, con un ligero sesgo hacia calificaciones altas, pocos títulos tienen calificaciones muy bajas (por debajo de 5) o muy altas (por encima de 9), lo que sugiere que la mayoría de las películas reciben críticas moderadas a favorables.

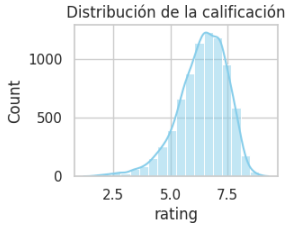


Figura 3. Visualización de distribución de calificación

Algo interesante y a considerar es el desbalance de clases al encontrar películas que tienen asociados hasta 9 géneros, mientras otras sólo se asocian con 1. Esto puede dificultar la predicción.

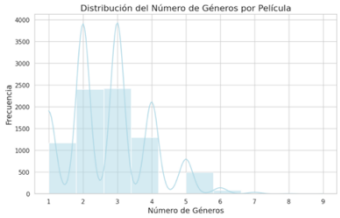


Figura 4. Géneros por película

DATA PREPARATION

TRANSFORMACIÓN DE LA VARIABLE PREDICTORA TRAMA ('PLOT')

Para alcanzar los objetivos propuestos, se llevaron a cabo diferentes procedimientos de preparación y preproce-

samiento de los datos a partir de las siguientes 3 estrategias:

1ra estrategia (Métodos de vectorización básicos): En primer lugar, se busca realizar una representación vectorial de las palabras de las diferentes tramas de las películas a partir de los métodos *CountVectorized* y *ifidfVectorized***.**

Sin embargo, previo a la representación y tokenización de las palabras en los dos métodos, se propondrá 3 maneras diferentes de preprocesar las tramas, las cuales se describen a continuación:

Preprocesamiento 1 'Original': Se busca normalizar todas las palabras en minúsculas (lowercasing), eliminar las palabras con poco contenido semántico (stopwords) y los signos de puntuación. El uso de los métodos **CountVectorized** y **TfidfVectorized** aplican estos preprocesamientos antes de aplicar la tokenización de manera que no se requiere de otras herramientas para esto.

Para la tokenización en los 2 métodos, se variarán 2 de sus parámetros con el fin de realizar un barrido de sus combinaciones como método de experimentación en los resultados que se obtendrán con los modelos de clasificación. Los parámetros son: min_df (de 1 a 10) y ngram_range ((1,1) y (1,4)).

Preprocesamiento 2 'Lemmatizing: Al igual que el procesamiento 1, se realizará un proceso de limpiezadonde se utilizará la librería *nltk* para realizar primero el lowercasing y luego la eliminación de stopwords y de signos de puntuación. En adición a este proceso de limpieza, se aplicará un proceso de lematización. Sin embargo, se mantendrá en los métodos de tokenización el parámetro *stop_words = 'english'* que también aplica la eliminación de stopwords.

Al igual que en el procesamiento 1, una vez realizado el proceso de limpieza y de lematización, se procederá con la tokenización variando de igual manera los 2 parámetros descritos anteriormente.

Preprocesamiento 3 'Stemming': Este procesamiento se realizará prácticamente igual que el procesamiento 2, con la diferencia de realizar el proceso de stemming en vez de la lematización.

Una vez llevado a cabo uno de los 3 preprocesamientos, se realizará el mismo procedimiento para todos ellos, la división entre datos de entrenamiento y de prueba. El conjunto de datos de entrenamiento se divide en un 33% para pruebas y un 67% para el entrenamiento mediante la función *train_test_split*, que asegura que los datos se separen de manera aleatoria y reproducible con una semilla (random_state=42).

2da estrategia (Vectorización word2vec): Como segunda estrategia se aplica el método de vectorización **word2vec** para transformar las tramas de las películas en vectores numéricos. Este método se basa en la representación de palabras en un espacio vectorial, capturando relaciones semánticas entre ellas, limitación que presenta la primera estrategia. A continuación, se describe su aplicación:

Preprocesamiento: Normalización del texto que incluye: conversión de todas las palabras en minúsculas (lowercasing), eliminar las palabras con poco contenido semántico (stopwords) y los signos de puntuación. Se aplica la técnica de stemming para reducir las palabras a su raíz o forma base. Esto permite agrupar palabras que comparten la misma raíz.

Creación de embeddings con el modelo W2v_model: Se entrena un modelo Word2Vec con las tramas de las películas preprocesadas. Este modelo aprende a representar cada palabra en el texto como un vector en un espacio multidimensional, donde captura las relaciones semánticas entre palabras, posicionando las más similares cerca unas de otras en este espacio. Se establece un tamaño de vector predefinido (en este caso 200), ajustado para equilibrar entre la capacidad del modelo y la complejidad del problema.

Construcción del vector promedio de los embeddings: Para cada trama de película, se genera un vector promedio que representa la combinación de las palabras que contiene. Se toma cada palabra del texto, se busca su vector correspondiente en el modelo W2v_model, y los vectores se suman y se promedian para obtener una única representación vectorial de la trama completa.

Posteriormente, se genera una lista donde cada entrada corresponde al vector promedio de una trama de película. Esta lista de vectores se utiliza como entrada para los modelos de clasificación.

División entre datos de entrenamiento y prueba: El conjunto de datos de vectores generados con Word2Vec se divide en un 33% para pruebas y un 67% para el entrenamiento mediante la función *train_test_split*, que asegura que los

datos se separen de manera aleatoria y reproducible con una semilla (random_state=42).

3ra estrategia (USE – Modelo de vectorización pre-entrenado): Para la tercera estrategia se aplica el método Universal Sentence Encoder (USE), un modelo de aprendizaje profundo diseñado para capturar el significado de oraciones completas en lugar de palabras individuales, como lo hace Word2Vec, el cual se desarrolló de la siguiente manera:

A diferencia de las otras 2 estrategias, el método USE no requirió un preprocesamiento previo dado que el recibe las tramas con el formato original y el decide que preprocesamiento requiere para llevar a cabo la tokenización.

Para cada trama de película, se genera un vector de tamaño 512 dimensiones en una representación densa que intenta codificar toda la información relevante sobre la oración en términos de su significado.

El modelo USE se utiliza para obtener un vector por cada trama, los cuales se almacenan en una columna adicional del conjunto de datos. Estos vectores se utilizan posteriormente en el proceso de clasificación.

Al igual que en Word2Vec, los datos vectorizados se dividen en conjuntos de entrenamiento y prueba con la misma proporción permitiendo entrenar los modelos de clasificación y evaluar su rendimiento.

TRANSFORMACIÓN DE LA VARIABLE DE INTERÉS 'GENRES'

Adicionalmente a la preparación de las tramas como variable predictora, se realiza también la transformación de la variable de interés a predecir 'genres' que corresponde a los géneros asociados a las películas en el conjunto de datos. Para ello, empleando la función *MultiLabelBinarizer*, se convierte la lista de géneros en una matriz binaria donde cada columna corresponde a un único género y cada película tendrá un 1 en caso de ser clasificada en la correspondiente columna, 0 en caso de no estarlo.

Dataset description:

Las vistas minables trabajadas para el desarrollo de todos los modelos a aplicar son las siguientes:

Estrategia 1 – TfidfVectorizer y CountVectorized: En esta estrategia, varios conjuntos de entrenamiento son implementados. Teniendo en cuenta que se van a generar para cada uno de los 2 métodos de tokenización los 3 preprocesamiento de las tramas de las películas y con 2 rangos de ngrams, se obtendrán 12 grandes estructuras de vistas minables en esta estrategia. Si bien se quiere también variar el parámetro min_df, el cual también cambia la dimensionalidad del número de atributos predictores, se estudiará aquel valor que mejor resultado brinde y se volverá un parámetro fijo como ya lo es el rango ngrams.

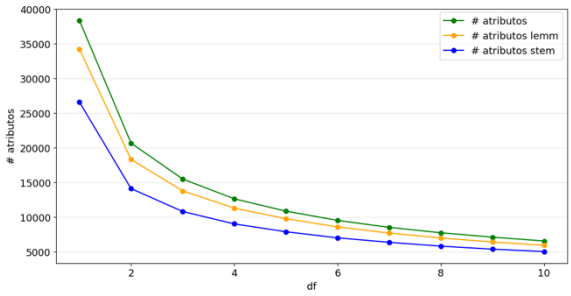
Como se visualiza en la figura 5, estas 12 grandes vistas minables van a compartir la mayoría de los valores de sus correspondientes arreglos. Estos valores en común son el arreglo de la clase de entrenamiento con **5289** tramas de películas y **24** posibles géneros a los que pertenece su clasificación. De igual manera, para la clase de validación se tiene un arreglo de **2606** tramas de películas y **24** posibles géneros.

```
Atributos entrenamiento: 5289 , a)
Atributos validación: 2606 , a)
Clase entrenamiento: (5289, 24)
Clase validación: (2606, 24)
```

Figura 5. Vista minable general de la estrategia 1

Sin embargo, en donde se van a diferenciar las diferentes vistas minables será en la dimensionalidad de los arreglos de los atributos de entrenamiento y validación. Si bien van a compartir el mismo número de tramas (**5289**), el número de atributos (**a**) variará en función de las palabras que sean consideradas para conformar el diccionario del método de tokenización.

En las figuras 6 y 7, se visualiza justamente el impacto sobre la dimensionalidad mencionado previamente por el parámetro min_df, tanto en el método TfidfVectorizer y el CountVectorized, según el rango de ngrams fijado ((1,1) o (1,4)). En la figura 6, vemos que el número de atributos resulta exactamente igual aplicando un método u otro con rango ngrams = (1,1).

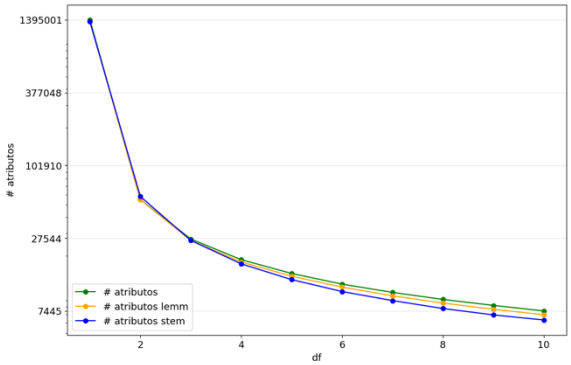


min_df	1	2	3	4	5
P1 (Original)	38,370	20,677	15,501	12,647	10,870
P2 (Lemmatize)	34,239	18,332	13,788	11,308	9,774
P3 (Stemming)	26,615	14,123	10,819	9,033	7,899

min_df	6	7	8	9	10
P1 (Original)	9,537	8,531	7,753	7,122	6,565
P2 (Lemmatize)	8,600	7,707	7,001	6,421	5,950
P3 (Stemming)	7,022	6,373	5,826	5,389	5,045

Figura 6. # Atributos de train y test con TfidfVectorizer y CountVectorized con ngram range (1,1) según df y tipo de preprocesamiento

De igual manera, en la figura 7 también se identifica el mismo comportamiento dónde, para el rango de ngrams = (1,4), ambos métodos de tokenización cuentan con el mismo tamaño de vocabulario.



min_df	1	2	3	4	5
P1 (Original)	1,395,001	54,997	27,113	18,709	14,596
P2 (Lemmatize)	1,352,276	54,932	26,422	17,939	13,903
P3 (Stemming)	1,347,102	58,261	26,448	17,346	13,072

min_df	1	2	3	4	5
P1 (Original)	12,054	10,391	9,156	8,226	7,445
P2(Lemmatize)	11,420	9,768	8,576	7,669	6,954
P3 (Stemming)	10,553	8,970	7,776	6,934	6,329

Figura 7. # Atributos de train y test con TfidfVectorizer y CountVectorized con ngram range (1,4) según df y tipo de preprocesamiento

Estrategia 2 - Vectorización word2vec*:* El conjunto de entrenamiento cuenta con **5289** tramas de películas y cada una está representada por un vector de **200 atributos**. El conjunto de validación tiene **2606** tramas y, al igual que el conjunto de entrenamiento, cada muestra está representada por un vector de **200** atributos.

```
Atributos entrenamiento: (5289, 200)
Atributos validación: (2606, 200)
Clase entrenamiento: (5289, 24)
Clase validación: (2606, 24)
```

Figura 8. Vista minable estrategia 2

La variable objetivo para el conjunto de entrenamiento tiene **5289** tramas, y cada una pertenece a una o más de **24 clases**, lo que indica que hay 24 posibles géneros o combinaciones de géneros. La dimensión de esta vista minable se representa en la figura 8.

Estrategia 3 - (USE – Modelo de vectorización pre-entrenado): El conjunto de entrenamiento cuenta con **5289** tramas de películas y cada una está representada por un vector de **512 atributos**, esto dado que el modelo pre-entrenado cuanta con un diccionario ya establecido de **512** tokens. El conjunto de validación tiene **2606** tramas y, al igual que el conjunto de entrenamiento, cuenta con **512** atributos. Los arreglos de las clases de entramiento cuentan con **5289** y **2606** tramas respectivamente, y para cada una de estas tramas pertenece una o más de las **24** clases o géneros. La dimensión de esta vista minable se representa en la figura 9.

```
Atributos entrenamiento: (5289, 512)
Atributos validación: (2606, 512)
Clase entrenamiento: (5289, 24)
Clase validación: (2606, 24)
```

Figura 9. Vista minable estrategia 3

MODELING

A continuación, se describe brevemente cada uno de los modelos implementados para la clasificación de las tramas de las películas con sus correspondientes parámetros definidos.

OneVsRestClassifier(modelo): Previo a explicar los 3 modelos implementados para resolver el problema multiclase, es necesario explicar esta función dado que

será primordial su uso como estrategia para habilitar la clasificación multiclase.

OneVsRestClassifier es una estrategia común en machine learning para manejar problemas de clasificación multiclase y multietiqueta. Consiste en entrenar un clasificador binario para cada clase individual de manera independiente, aprendiendo a distinguir una clase específica del resto. [1]

La función *OneVsRestClassifier* crea un clasificador binario para cada una de las clases (géneros) y lo entrena de forma independiente con la misma información de entrada. Para la predicción, se evalúan todos los clasificadores y se obtiene una probabilidad para cada clase, asignando la pertenencia (1) o no pertenencia (0) de la observación con el género.

Modelo 1 - RandomForest: Es un es un modelo de ensambles basado en la combinación de varios árboles de decisión con el propósito de mejorar la precisión y reducir el sobreajuste que un único árbol por sí solo suele tener.

Es un algoritmo que utiliza el método de bagging (Bootstrap Aggregating), lo que significa que varios modelos débiles se combinan para crear un modelo fuerte al promediar o combinar los resultados de muchos modelos independientes, reduciendo el riesgo de grandes errores que un solo modelo podría cometer.

A continuación, se listan los parámetros establecidos para la ejecución de este modelo con todas las vistas minales procedentes de las 3 estrategias de preprocesamiento.

- n_jobs=-1
- n_estimators=200
- max_depth=15
- random_ state =42

Modelo 2 - Regresión logística: es un método de clasificación utilizado principalmente en problemas donde la variable objetivo es categórica clasificando observaciones en dos o más clases.

Se basa en estimar una probabilidad al realizar la combinación lineal y aplicando el resultado en una función logística, sigmoide en caso de ser un problema binario, para hacer predicciones. Se considera un umbral que, al ser superado, clasifica a la observación en la clase de interés.

Teniendo en cuenta que la predicción de géneros de películas representa un problema de predicción multiclases, la función *OneVsRestClassifier* explicada previamente, habilita emplear el modelo de regresión logística para el desarrollo del proyecto.

A continuación, se listan los parámetros establecidos para la ejecución de este modelo con todas las vistas minales procedentes de las 3 estrategias de preprocesamiento.

- max_iter=1000
- random_state=42
- n_jobs=-1
- solver='lbfgs'

Modelo 3- XGBoost: El modelo XGBoost (Extreme Gradient Boosting) es una implementación optimizada y escalable del método de boosting, una técnica de ensamblaje que combina múltiples modelos débiles (normalmente árboles de decisión simples) para formar un modelo fuerte, mejorando la precisión de las predicciones. Esta estrategia de boosting está basada en árboles de decisión que ajustan secuencialmente modelos para corregir errores y mejorar el rendimiento.

A continuación, se listan los parámetros establecidos para la ejecución de este modelo con las vistas minales procedentes de la 1ra estrategia de preprocesamiento.

- n_jobs=-1,
- n_estimators=100
- max_depth=10
- random_state=42
- eval_metric='mlogloss'

Para las vistas minales procedentes de la 2da y 3ra estrategia de preprocesamiento, se modifica el parámetro n_estimators=200.

EVALUATION:

Se aplicaron 3 modelos, para los cuales se calculó el AUC como métrica de evaluación.

A continuación, se realiza la descripción de los modelos a partir de los resultados obtenidos con base en la métrica anteriormente indicada:

Modelo 1 - RandomForest: La mejor configuración de preprocesamientos y métodos de tokenización alcanzó un **AUC** de 0,8468 con el modelo de regresión logística y la

3ra estrategia de preparación de los datos con el tokenizador USE.

En relación con la 1ra estrategia de preparación de los datos se evidencia que el preprocesamiento 3, en dónde se implementa stemming, tuvo de manera general mejores métricas que los otros dos preprocesamientos. Esta tendencia es posible identificarla principalmente en las figuras 11, 12 y 13. Esta tendencia también es posible remarcarla en la tabla 2, dónde 3 de los mejores 4 resultados tras establecer los mejores min_df correspondiente al preprocesamiento 3.

Adicionalmente, es posible remarcar que el método de tokenización TfidfVectorized logró un ligero mejor desempeño, alcanzando valores del AUC cercanos al 0.83, con respecto al método CountVectorized, el cual alcanzó valores más cercanos al 0.82. Esto se observa al analizar las gráficas 10 y 11 en contraste con las gráficas 12 y 13.

Sin embargo, la diferencia en el uso de los métodos en la estrategia 1 es apenas significativa y no parece tener gran influencia elegir un método sobre otro.

En relación con el rango de ngrams, los modelos que tuvieron como parámetro el rango (1,4) generó valores más cercanos y altos entre los 3 preprocesamientos, evidenciando este comportamiento en los gráficos 11 y 13, en comparación con los modelos que usaron el rango (1,1), en los gráficos 10 y 12.

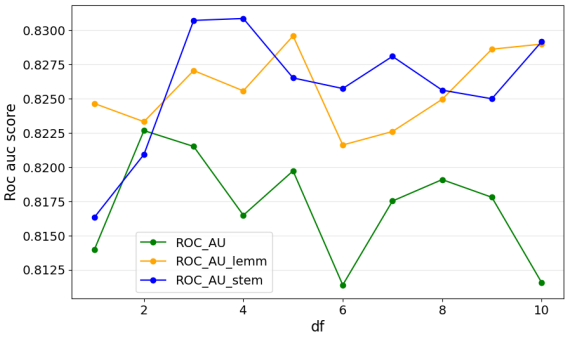


Figura 10. AUC tras tokenización con TfidfVectorizer con ngram_range (1,1) según tipo de preprocesamiento y min_df

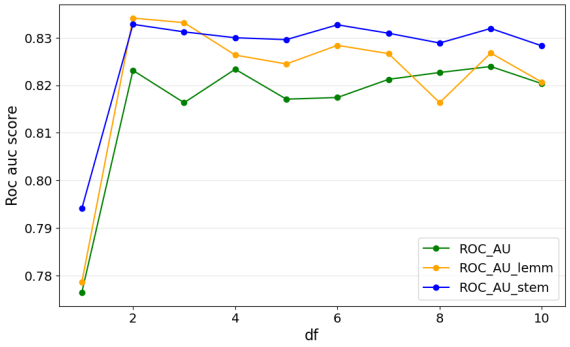


Figura 11. AUC tras tokenización con TfidfVectorizer con ngram_range (1,4) según tipo de preprocesamiento y min_df

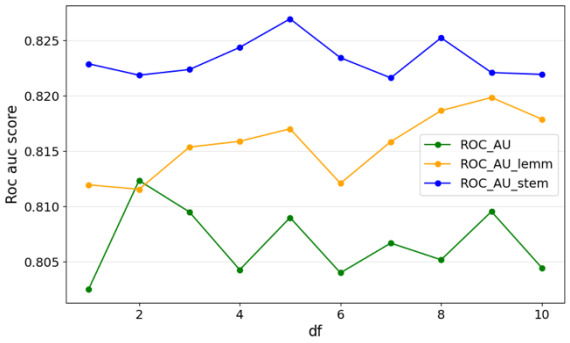


Figura 12. AUC tras tokenización con CountVectorized con ngram_range (1,1) según tipo de preprocesamiento y min_df

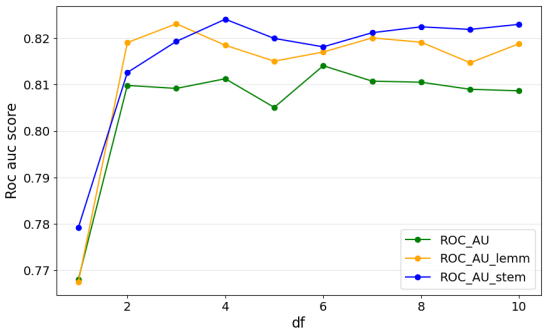


Figura 13. AUC tras tokenización con CountVectorized con ngram_range (1,4) según tipo de preprocesamiento y min_df

método tokenización	Preprocesamiento + parám. min_df	Mejor auc
TfidfV(ngram_range=(1,1))	P3 (stem) - min_id = 4	0,8308
TfidfV(ngram_range=(1,4))	P2 (lemm) - min_id = 2	0,8341
countV(ngram_range=(1,1))	P3 (stem) - min_id = 5	0,8269
countV(ngram_range=(1,4))	P3 (stem) - min_id = 4	0,8241

Tabla 2. Mejores resultados con RandomForest según método de tokenización y preprocesamientos 1ra estrategia

En cuanto a la 2da estrategia, la tokenización Word2Vec obtuvo las métricas más bajas al modelar con

RandomForest. Esta estrategia apenas supera el umbral del 0.7 estando lejos del mejor registro de las otras 2 estrategias. En contrapartida, la 3ra estrategia alcanza una métrica del 0.8468 superando las mejores métricas de la 1ra estrategia.

Estrategia de vectorización	auc score
Estrategia 2 - Word2Vec	0,7126
Estrategia 3 – USE	0,8468

Tabla 3 Valores AUC del modelo RandomForest según estrategias

Modelo 2 - Regresión logística:

Al aplicar regresión logística para predecir el género de películas, utilizando las técnicas de vectorización de la 1ra estrategia (TF-IDF y CountVectorizer) y configuraciones, se obtuvo el mejor rendimiento al aplicar **TfidfVectorized (ngram_range = (1,4))** con preprocesamiento **P3 (stem)** y un parámetro **min_id = 2**, alcanzando una métrica AUC de **0.8823.**

Teniendo en cuenta el desarrollo de la primera estrategia se identifica que al aplicar la técnica de stemming se obtienen mejores resultados en el AUC, a partir de ello se aplica en el preprocesamiento la técnica de stemming para la estrategia 2.

método tokenización	Preprocesamiento + parám. min_df	Mejor auc
TfidfV(ngram_range=(1,1))	P3 (stem) - min_id = 6	0,8801
TfidfV (ngram_range=(1,4))	P3 (stem) - min_id = 2	0,8823
countV(ngram_range=(1,1))	P1(orig) - min_id = 1	0,8484
countV(ngram_range=(1,4))	P3 (stem) - min_id = 1	0,8554

Tabla 4. Mejores resultados con Regresión Logística según método de tokenización y preprocesamientos 1ra estrategia

Las gráficas que se presentan a continuación ilustran el comportamiento del AUC con respecto al parámetro min_df, aplicando los diferentes métodos de tokenización y preprocesamiento:

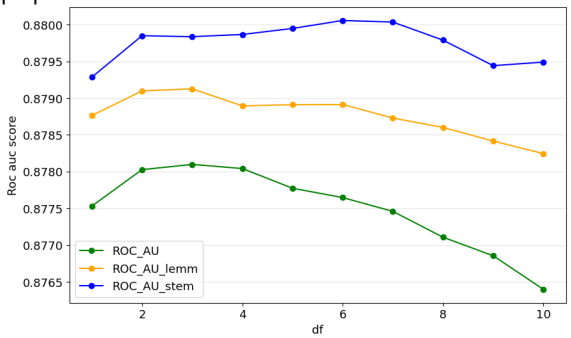


Figura 14. AUC tras tokenización con TfidfVectorizer con ngram_range (1,1) según tipo de preprocesamiento y min_df

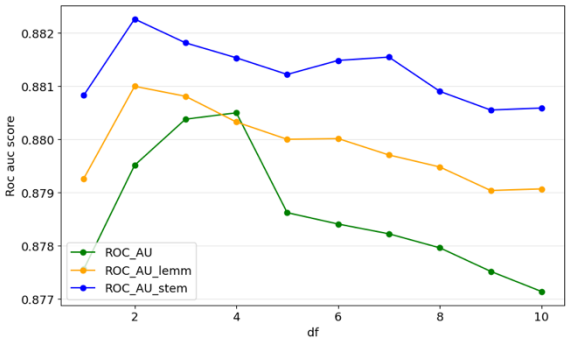


Figura 15. AUC tras tokenización con TfidfVectorizer con ngram_range (1,4) según tipo de preprocesamiento y min_df

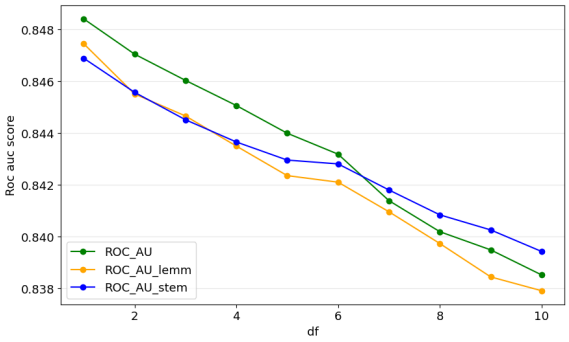


Figura 16. AUC tras tokenización con CountVectorized con ngram_range (1,1) según tipo de preprocesamiento y min_df

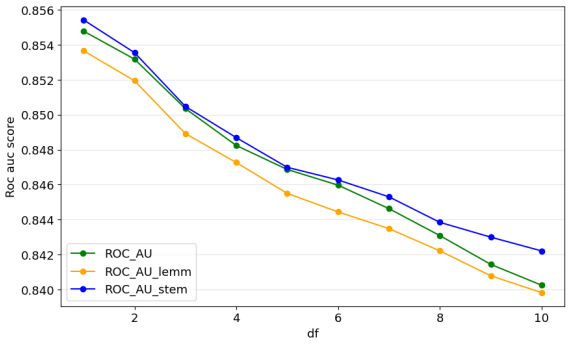


Figura 17. AUC tras tokenización con CountVectorized con ngram_range (1,4) según tipo de preprocesamiento y min_df

Para la segunda y tercera estrategia se logró un AUC de **0,7554**, lo que indica un desempeño moderado en la clasificación de los géneros. El modelo basado en USE logró un desempeño superior, alcanzando un AUC de **0,8996**. Esto muestra que USE capturó mejor las relaciones semánticas en las tramas de las películas, permitiendo una clasificación más precisa. Con este método y el modelo de regresión logística se alcanza la métrica establecida como objetivo de minería. A continuación, se muestran los

valores correspondientes a las estrategias de vectorización empleadas:

Estrategia de vectorización	auc score
Estrategia 2 - Word2Vec	0,7554
Estrategia 3 – USE	0,8996

Tabla 5. Valores AUC del modelo Regresión logística según estrategias

Modelo 3 – XGBoost: La mejor combinación de preprocesamiento y métodos de tokenización logró un AUC de 0.8861 utilizando el modelo XGBoost y la tercera estrategia de preparación de datos con el tokenizador USE.

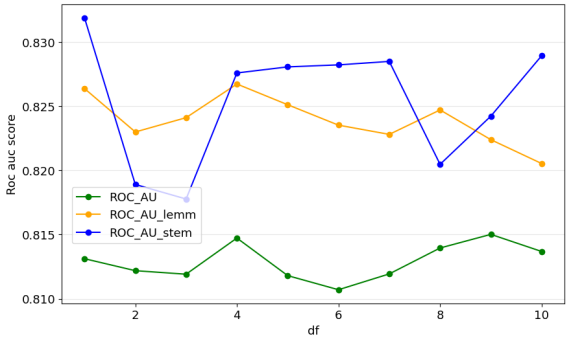


Figura 18. AUC tras tokenización con TfidfVectorizer con ngram_range (1,1) según tipo de preprocesamiento y min_df

Respecto de la primera estrategia de preparación, se observa que, al igual que en el primer modelo, el tercer preprocesamiento, que incluye la técnica de stemming, produjo mejores métricas que las otras dos alternativas; tendencia que se puede apreciar en las figuras 18 y 19.

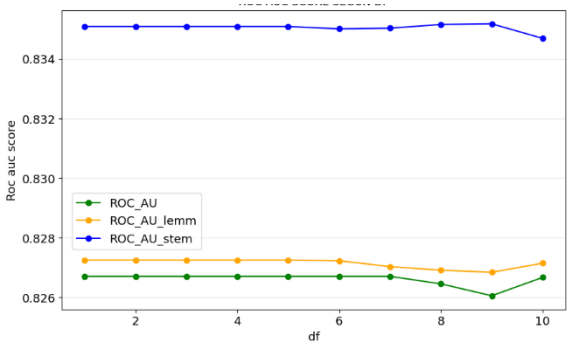


Figura 19. AUC tras tokenización con CountVectorized con ngram_range (1,1) según tipo de preprocesamiento y min_df

Al comparar los métodos de tokenización se encontró que la diferencia entre TfidfVectorizer y CountVectorizer no es representativa, ya que ambos se sitúan alrededor de 0.83 (ver Tabla 6)

método tokenización	Preprocesamiento + parám. min_df	Mejor auc
TfidfV(ngram_range=(1,1))	P3 (stem) - min_id = 1	0,8352
countV(ngram_range=(1,1))	P3 (stem) - min_id = 9	0,8318

Tabla 6. Mejores resultados con XGBoost según método de tokenización y preprocesamientos 1ra estrategia

En referencia a las otras 2 estrategias se observó que la utilización del método de clasificación Word2Vec arrojó resultados significativamente inferiores, ya que se encuentra 0.10 puntos por debajo de la estrategia 1 y 0.15 puntos por debajo de la estrategia 3 en términos de AUC.

método de clasificación	auc score
Estrategia 2 - Word2Vec	0,7371
Estrategia 3 – USE	0,8861

Tabla 7. Valores AUC del modelo XGBoost según 2da y 3ra estrategia

RECOMENDACIONES DE NEGOCIO

Enmarcados en los objetivos de minería de datos y de negocio se recomiendan las siguientes estrategias según los modelamientos:

- Implementar un modelo de clasificación que utilice regresión logística, junto con un método de tokenización basado en el modelo USE (Universal Sentence Encoder). Este enfoque permitirá identificar con mayor precisión los géneros de cada película basándose en su trama y las búsquedas realizadas por los usuarios, mejorando la organización del contenido y reduciendo el tiempo que los usuarios dedican a buscar películas e incrementando su tiempo de consumo de contenido en la plataforma.
- Utilizar la clasificación para identificar los géneros predichos con mayor y menor representación en el catálogo de la plataforma, con el fin de combinar estos datos con información de consumo de los usuarios para la adquisición de nuevo contenido.
- Clasificar automáticamente nuevos contenidos por género en grandes catálogos de películas o producciones recientes, utilizando el modelo USE, lo que permite agilizar el proceso de etiquetado y mejorar la precisión en la asignación de géneros. Optimizando la búsqueda de películas en las plataformas y facilitando la gestión eficiente del contenido.

BIBLIOGRAFIA

[1]Scikitlearndevelopers. (n.d.). sklearn.multiclass.OneVsRestClassifier. Scikit-learn 1.3.0 documentation.

https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html

González Mallo, A. (2024). Extracción de características basadas en sinopsis para la clasificación de películas en géneros cinematográficos. Extraído de: https://ruc.udc.es/dspace/handle/2183/39294