



PRESENTACIÓN DE LOS RESULTADOS

ANÁLISIS DE TWEETS PARA RECOMENDACIÓN DE PELÍCULAS

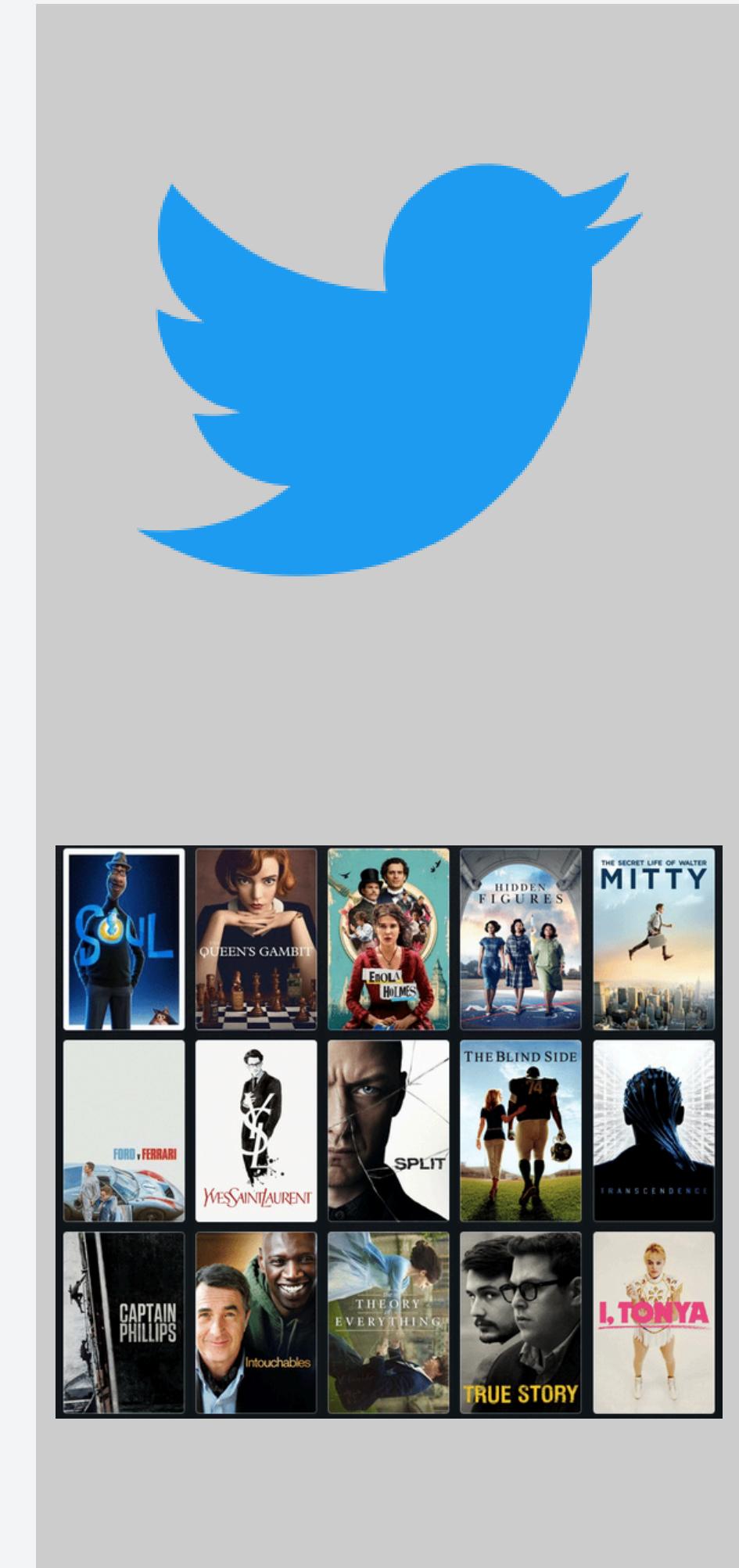
RESUMEN



Desarrollar un sistema de recomendación de películas personalizado que extraiga y analice tweets para identificar preferencias cinematográficas.



El sistema deberá recoger los tweets de usuarios específicos, entender si le gusta, extraer sobre la película que se habla y extraer información relevante para generar recomendaciones de películas ajustadas a los gustos personales de cada usuario.



Extracción de datos



Manejo de datos



Visualizaciones



Procesamiento de lenguaje natural

spacy



Aprendizaje automático



DATASETS

Tweets extraídos

	film_name	review_text	polaridad
0	Anatomía de una caída	Dalf @Dalf_noir · 22 mar. Ya pude ver anatomy ...	positivo-Recomendable
1	¡Shazam! La furia de los dioses	Mr. Savitar @MrSavitar1790 · 17 mar. 🎬🎥 HACE UN...	neutro/negativo-noRecomendable
2	El astronauta	CumbanchaVolante @Cumbanch · 23 mar. 120 minut...	neutro/negativo-noRecomendable
3	La Liga de la Justicia de Zack Snyder	El Pare 🎬 Cine y series @ElPare89 · 18 mar....	positivo-Recomendable
4	John Wick 4	Lugar Cinema 🎬 @LugarCinema · 25 mar. 1 AÑO DEL...	positivo-Recomendable

Reviews
filmaffinity

	film_name	gender	film_avg_rate	review_rate	review_title	review_text
0	Ocho apellidos vascos	Comedia	6,0	3.0	OCHO APELLIDOS VASCOS...Y NINGÚN NOMBRE PROPIO	La mayor virtud de esta película es su existen...
1	Ocho apellidos vascos	Comedia	6,0	2.0	El perro verde	No soy un experto cinéfilo, pero pocas veces m...
2	Ocho apellidos vascos	Comedia	6,0	2.0	Si no eres de comer mierda... no te comes esta...	Si no eres un incondicional del humor estilo T...
3	Ocho apellidos vascos	Comedia	6,0	2.0	Aida: The movie	No sé qué está pasando, si la gente se deja ll...
4	Ocho apellidos vascos	Comedia	6,0	2.0	UN HOMBRE SOLO (Julio Iglesias 1987)	"Pero cuando amanece,y me quedo solo,siento en...

Dataset películas

	Series_Title	Series_Title_Spanish	Genre
0	The Shawshank Redemption	Cadena perpetua	Drama
1	The Godfather	El Padrino	Crime, Drama
2	The Dark Knight	El caballero oscuro	Action, Crime, Drama
3	The Godfather: Part II	El Padrino: Parte II	Crime, Drama
4	12 Angry Men	12 hombres enojados	Crime, Drama

01

02

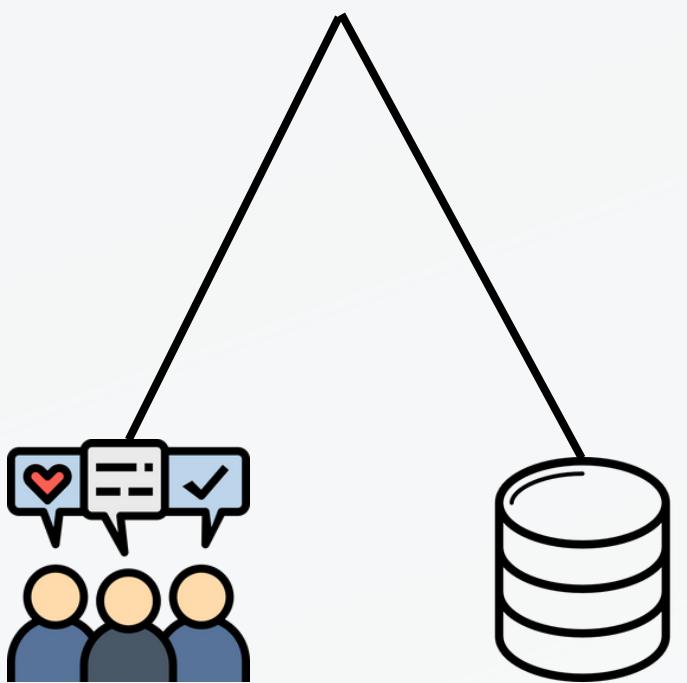
03

04

05

06

EXTRACCIÓN
DE TWEETS
REALES



CARGA DE
DATOS DE
SOPORTE

ANÁLISIS DE
SENTIMENTOS



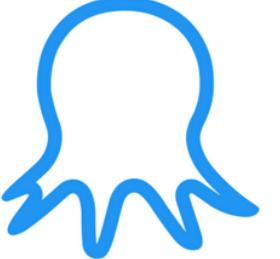
EXTRACCIÓN
PELÍCULA



ALGORITMO
DE
RECOMENDACI
ÓN



PUESTA EN
PRODUCCIÓN
EN LOCAL



EXTRACCIÓN DE TWEETS

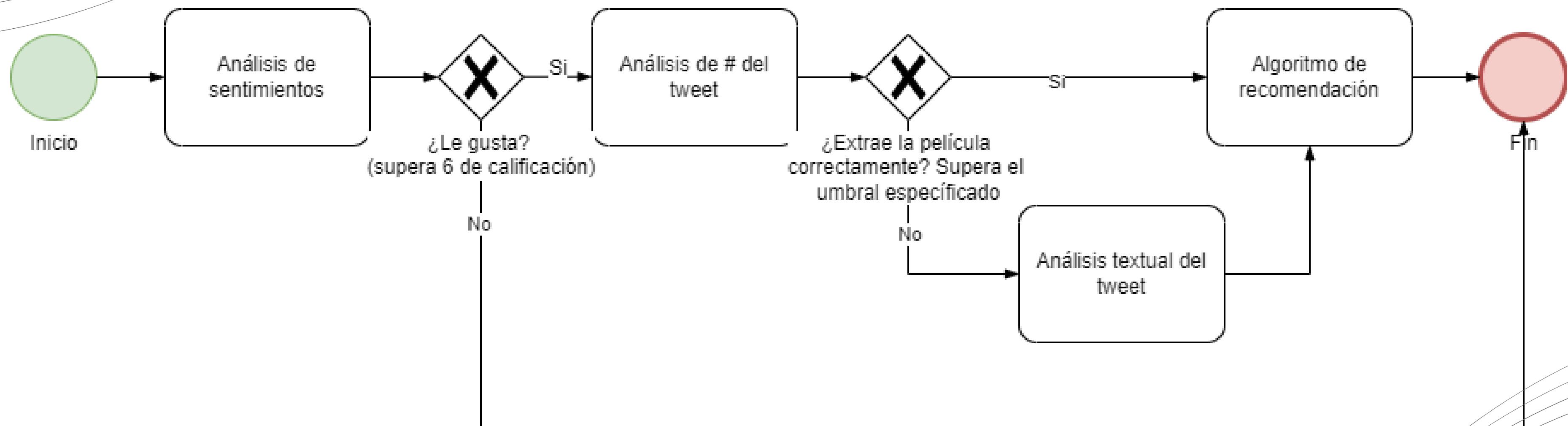
WEB SCRAPING



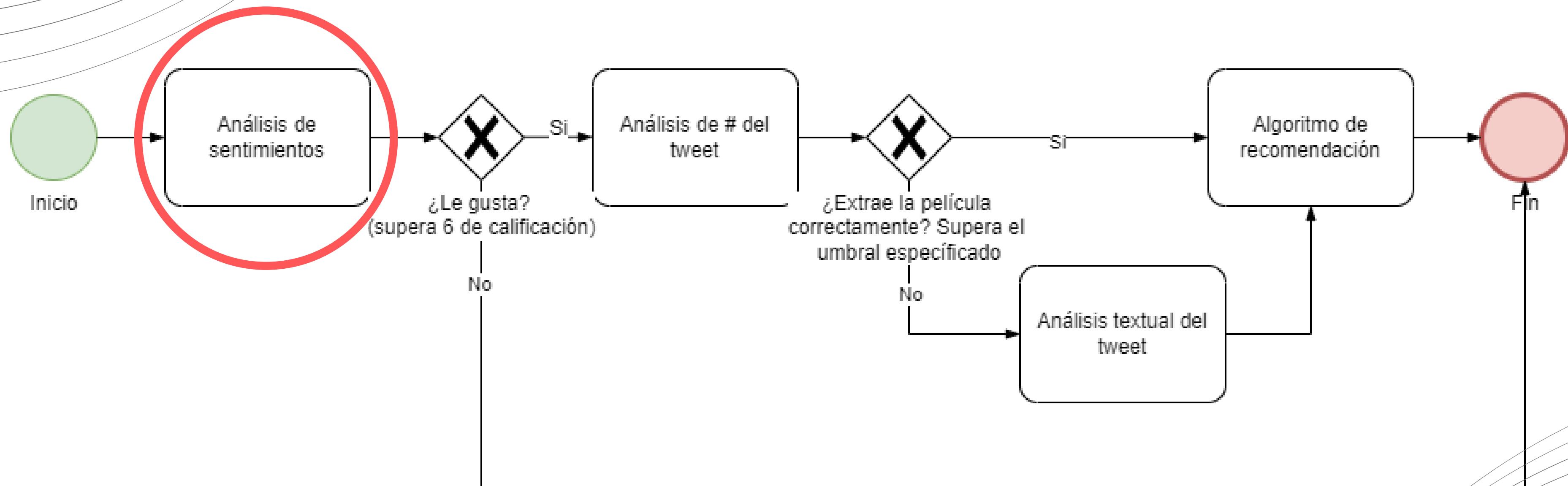
- Tweets españoles.
- Tweets en los que se pueda saber leyéndolos si al usuario le ha gustado la película o no.
- Tweets en los que de alguna manera se pudiera identificar la película de la que habla.



PROCESO BPMN DE CLASIFICACIÓN



PROCESO BPMN DE CLASIFICACIÓN



ANÁLISIS DE SENTIMIENTOS

- **Discretizamos la variable objetivo(rate) en dos clases** --> >6 recomendable y <= 6 no recomendable
- **Normalizacion/Preprocesamiento** --> Con el modelo en español de spacy quitamos los signos de puntuación, quitar palabras menores de tres caracteres y mayores que 11 caracteres, quitamos stop_words, quitamos saltos de línea y quitamos todo lo que no sea nombre adjetivo verbo y adverbio.

'apellido vasco ningun nombre virtud pelicula jugar topico extremo patrio andaluz vasco poner grito cielo indicar madurez pese pese corrijo jugar mofa topico vasco vasco poner grito cielo indicar grado cuestion llaga abierto dispuesto sangrar correccion andaluz motivo guasa apellido vasco valiente oportuno razn espanol acudir masa masa cine producto tele acertado fecha estreno verano brutal ejemplar campán marketing aplaudo celebrar decir repito oscuro placer pelicula mala insistir olvidar principal proposito cine entretenido decir mantengo problema apellido vasco malo friki peli mala"(como sharknado xanadu condenar perdonar él punto palabra define mediocre simple sencillo impersonal hecha interesar novedoso sacar elemento entorno totalmente ajeno hostil argumento basico comedia cine cine explicar dema elemento encajar absoluto pelicula caida picado comienzo prometedor vergonzoso pasar comica topico manido comedia enredo cesar oír replicar yo historia contado tener razon seguir contar gana forma origin al volver principal problema cine patrio foraneo mayoria director confundir argumento argumento desarrollo grandisima pelicula espanola llevar ano sufrir desinflar suerte mitad recorrido cineasta molestar historia rematarla cosa encajen escuela cine surg en paí olvidar ocupar ejemplo torpemente presentado falta gana director refugiar enredo salir intuir temer principio fuerza salir producto descalabro rovirar inspirado monologo desparpajo karra elejalde dar vida unico personaje creible historia carmen machi carmen machi claro lago llamar él actriz serio insultar resto else capaz arrancar risa carcajada resultar ratos doler pensar publico esperar cine espanol llenar sala doler pensar cine ambito manos mediocr doler pensar salvar cine espanol agua cita amigo regino mateo pelicula cine'

ANÁLISIS DE SENTIMIENTOS

- Representacion vectorial de texto --> Bolsa de palabras
- Dos tipos:
 - Conteo

#Creamos un modelo de bolsa de palabras con las 2000 palabras más frecuentes de los textos de entrenamiento que aparezcan por lo menos en 500 documentos distintos.

```
bow_kaggle = CountVectorizer(max_features=2000, min_df=500, ngram_range=(1,2))
```

```
#EJEMPLOO
from sklearn.feature_extraction.text import CountVectorizer
comentarios = ["malisima", "no no malisima", "me gustó", "no me gustó", "no creo que sea una buena pelicula", "es buenisima"]

count_vectorizer = CountVectorizer(min_df=2, max_df=0.5, ngram_range=(1,2))

features = count_vectorizer.fit_transform(comentarios)

df = pd.DataFrame(
    features.todense(),
    columns=count_vectorizer.get_feature_names_out()
)
print(df)
```

	gustó	malisima	me	me gustó	no
0	0	1	0	0	0
1	0	1	0	0	2
2	1	0	1	1	0
3	1	0	1	1	1
4	0	0	0	0	1
5	0	0	0	0	0

ANÁLISIS DE SENTIMIENTOS

- Term frequency

```
# ngramas o bigramas (columnas) que minimo aparezcan en 2 textos y maximo en 50% de los textos
tfidf = TfidfVectorizer(min_df=2, max_df=0.5, ngram_range=(1,2))
```

- **Representacion vectorial de texto** --> Buscar palabras positivas y negativas -->
Especie de bolsa de palabras utilizando como columnas palabras positivas y
negativas. BOW de conteo. A esa bolsa de palabras se le hace la lematizacion. Esta se
le aplica a stemming
- **Transformer and hugging face** --> No conseguimos utilizarlo. Fallos en la
instalación.

ANÁLISIS DE SENTIMIENTOS

Evaluacion de las representaciones

```
evaluaciones_bow= evaluadorModelos2([knn_5, knn_10, knn_15, rf_20 ], x_kaggle_bow, y_kaggle, 5, 'accuracy')
evaluaciones_tfidf= evaluadorModelos2([knn_5, knn_10, knn_15, rf_20 ], x_kaggle_tfidf, y_kaggle, 5, 'accuracy')
evaluaciones_posNeg= evaluadorModelos2([knn_5, knn_10, knn_15, rf_10 ], vectorizer_posNeg_reviews, y_kaggle, 5, 'accuracy')
```

	Predictor	Rendimiento	Tiempo Ejecución (s)	CV	Scoring
0	KNeighborsClassifier()	0.565034	2.274915	5	accuracy
1	KNeighborsClassifier(n_neighbors=10)	0.579217	2.303843	5	accuracy
2	KNeighborsClassifier(n_neighbors=15)	0.581194	2.263948	5	accuracy
3	RandomForestClassifier(max_depth=20, n_estimators=10)	0.651399	51.042700	5	accuracy

ANÁLISIS DE SENTIMIENTOS

Evaluacion de las representaciones

	Predictor	Rendimiento	Tiempo Ejecución (s)	CV	Scoring
0	KNeighborsClassifier()	0.569920	2.330773	5	accuracy
1	KNeighborsClassifier(n_neighbors=10)	0.578752	2.335756	5	accuracy
2	KNeighborsClassifier(n_neighbors=15)	0.603166	2.324753	5	accuracy
3	RandomForestClassifier(max_depth=20, n_estimators=10, random_state=42)	0.653143	55.175453	5	accuracy

	Predictor	Rendimiento	Tiempo Ejecución (s)	CV	Scoring
0	KNeighborsClassifier()	0.542368	1.202784	5	accuracy
1	KNeighborsClassifier(n_neighbors=10)	0.559801	1.244674	5	accuracy
2	KNeighborsClassifier(n_neighbors=15)	0.554224	1.277588	5	accuracy
3	RandomForestClassifier(max_depth=10, n_estimators=10, random_state=42)	0.568640	11.549123	5	accuracy

ANÁLISIS DE SENTIMIENTOS

Predicción con los tweets reales

```
# Resultados de los modelos predictivos entrenados con las reviews informales

#BOW

print(cross_val_score(estimator=rf_20_tweets_sentimiento_bow, X=X_tweets_bow, y=y_tweets, cv=5, scoring='accuracy').mean())
print(cross_val_score(estimator=rf_20_tweets_sentimiento_bow, X=X_tweets_bow, y=y_tweets, cv=5, scoring='balanced_accuracy').mean())

print(cross_val_score(estimator=knn_15_sentimiento_bow, X=X_tweets_bow, y=y_tweets, cv=5, scoring='accuracy').mean())
print(cross_val_score(estimator=knn_15_sentimiento_bow, X=X_tweets_bow, y=y_tweets, cv=5, scoring='balanced_accuracy').mean())

#_tfidf

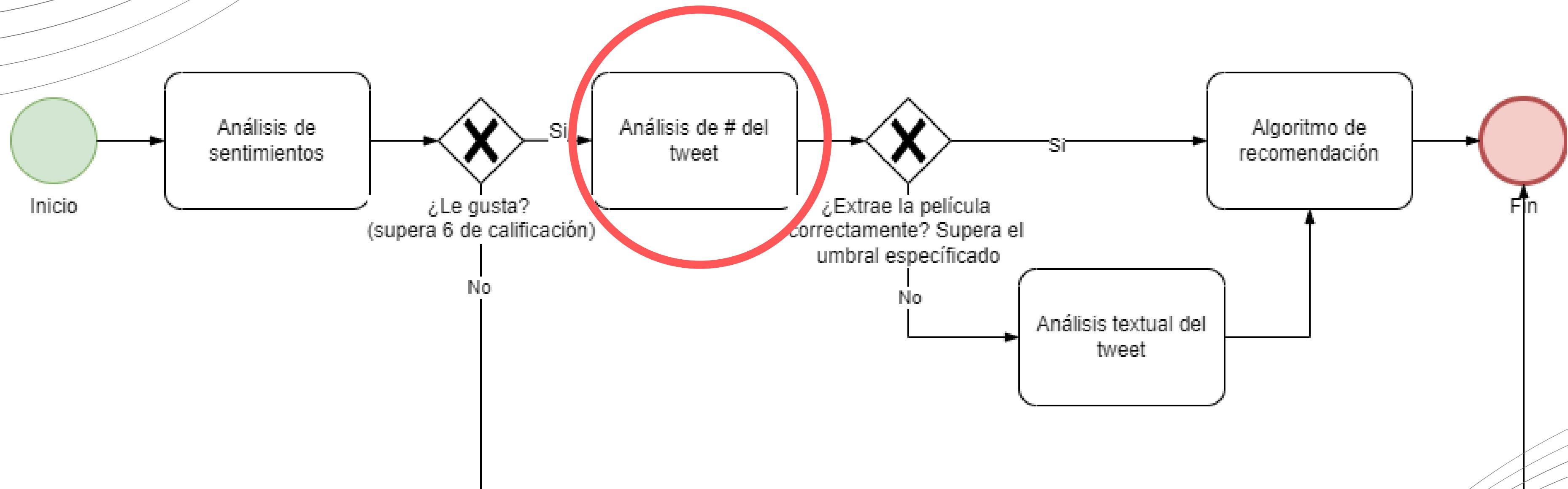
print(cross_val_score(estimator=rf_20_tweets_sentimiento_tfidf, X=X_tweets_tfidf, y=y_tweets, cv=5, scoring='accuracy').mean())
print(cross_val_score(estimator=rf_20_tweets_sentimiento_tfidf, X=X_tweets_tfidf, y=y_tweets, cv=5, scoring='balanced_accuracy').mean())

print(cross_val_score(estimator=knn_15_sentimiento_tfidf, X=X_tweets_tfidf, y=y_tweets, cv=5, scoring='accuracy').mean())
print(cross_val_score(estimator=knn_15_sentimiento_tfidf, X=X_tweets_tfidf, y=y_tweets, cv=5, scoring='balanced_accuracy').mean())

```

0.8
0.7166666666666666
0.67
0.5
0.76
0.75
0.67
0.5

PROCESO BPMN DE CLASIFICACIÓN



ANÁLISIS DE

- Funcion para extracción del #
- Devuelve una lista del texto del hashtag sin el simbolo #
- Aplicar la funcion a las columnas del df

	film_name	review_text	polaridad	name	username	date	hashtags
0	Anatomía de una caída	Ya pude ver anatomy of a fall y wow la películ...	positivo-Recomendable	Dalf	@Dalf_noir	22 mar.	
1	¡Shazam! La furia de los dioses	🎬🎬 HACE UN AÑO!!⏳⏳ Hoy hace un año se estrenaba...	neutro/negativo-noRecomendable	Mr. Savitar	@MrSavitar1790	17 mar.	[DCEU]
2	El astronauta	120 minutos de "el astronauta" con Adam Sandler...	neutro/negativo-noRecomendable	CumbanchaVolante	@Cumbanch	23 mar.	
3	La Liga de la Justicia de Zack Snyder	La #JusticeLeague cumple hoy 3 años. Su estreno...	positivo-Recomendable	El Pare 🎬 Cine y series	@ElPare89	18 mar.	[JusticeLeague, ZackSnyder]
4	John Wick 4	1 AÑO DEL ESTRENO DE JOHN WICK 4 🎬!! Una de las...	positivo-Recomendable	Lugar Cinema 🎬	@LugarCinema	25 mar.	
5	The iron claw	Vi The Iron Claw y si quieren que de opiniones...	positivo-Recomendable	Mr. TNA	@Owens_Avila02	23 mar.	

ANÁLISIS DE

Función para relacionar hashtag con películas:

- Recibe la lista de hashtags
- Normalización de los hashtags y del dataset de películas
- Vectorización de texto
- KNeighborsClassifier(n_neighbors=1)
- Selecciona el título con la menor distancia normalizada al hashtag

```
0      {}
1      {}
2      {}
3      {'JusticeLeague': ['Zack Snyder's Justice Leag...'}
4      {}
5      {}
6      {}
7      {}
8      {}
9      {}
10     {}
11     {}
12     {'Maléfica': ['Maléfica', 0.0]}
13     {'ApocalypseNow': ['Apocalypse Now', 0.0]}
14     {'SuperMarioBrosMovie': ['Super Mario Bros.: T...'}
15     {'JusticeLeagueDark': ['Justice League Dark: A...'}
16     {'Maleficio': ['Maleficio', 0.0]}
17     {'Eliminado': ['Eliminado', 0.0]}
18     {'notimetodie': ['No Time to Die', 0.0]}
19     {'ElGranShowman': ['El Gran Showman', 0.0]}
20     {'Fenomenosecuestro': ['Fenomeno Secuestro', 0...}
Name: hashtags, dtype: object
```

ANÁLISIS DE

```
def normalizar(palabra):
    return ''.join(e for e in palabra.lower() if e.isalnum())

#LEO LA BD PELICULAS DIRECTAMENTE --> meto todas las del inglés y las de español para que coja tambien los hashtag en inglés
peliculas= datos_knn['Title-Dependiente'].tolist() + datos_knn['Title-DependienteEnglish'].tolist()

peliculas_normalizadas = [normalizar(pelicula) for pelicula in peliculas]
hashtags_normalizados = [normalizar(h) for h in hashtags]

#vectorizer_char = CountVectorizer(analyzer='char')
#vectorizer_char_wb = CountVectorizer(analyzer='char_wb')
vectorizer_char = CountVectorizer(analyzer='char', ngram_range=(2, 3))
X_char_train = vectorizer_char.fit_transform(peliculas_normalizadas)

knn_char = KNeighborsClassifier(n_neighbors=1)
knn_char.fit(X_char_train, np.arange(len(peliculas_normalizadas)))

X_char_hashtags = vectorizer_char.transform(hashtags_normalizados)

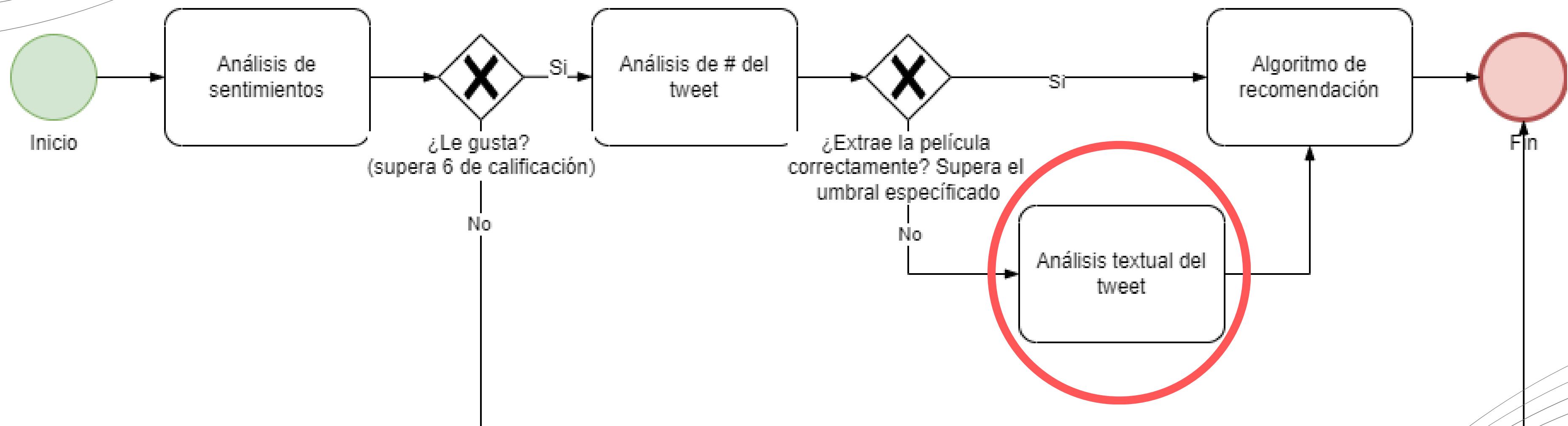
if X_char_hashtags.shape[0] == 0:
    return {} # Retorna un diccionario vacío si no hay hashtags para procesar
```

ANÁLISIS DE

```
df_reviews_tweets['hashtags'].apply(relacionar_hashtags_peliculas_bag_definitivo, limite_distancia=1)
```

```
# Devuelve solo el par con la menor distancia normalizada y que sea de una distancia menor de la indicada como límite  
#generalmente 1  
if menor_distancia>limite_distancia:  
    res= {}  
else:  
    res= {hashtag_seleccionado: [pelicula_seleccionada, menor_distancia]}  
return res
```

PROCESO BPMN DE CLASIFICACIÓN



ANÁLISIS TEXTUAL DEL TWEET

Función normalizar películas:

- Eliminación de Tildes y Conversión a Minúsculas
- Spacy para tokenización y lematización
- Normalización de las películas

```
normalized_doc = " ".join([word.lemma_ for word in doc if
                           not word.is_punct and # Filtra signos de puntuación
                           3 < len(word.text) < 11 and
                           '\n' not in word.text
                           ])
normalized_corpus.append(normalized_doc)
```

ANÁLISIS TEXTUAL DEL TWEET

Flujo de trabajo:

- Aplicar la normalización de películas
- Vectorización de las películas normalizadas
- Normalización de las películas y transformación a BoW
- Aplicar misma vectorización a los tweets
- Entrenamiento del modelo knn

```
knn_pelicula = KNeighborsClassifier(n_neighbors=1)
knn_pelicula.fit(X_BD_peliculas_bow, y_BD_peliculas)

pelicula_tweets_knn=knn_pelicula.predict(X_peliculas_tweets_bow)
pelicula_tweets_knn

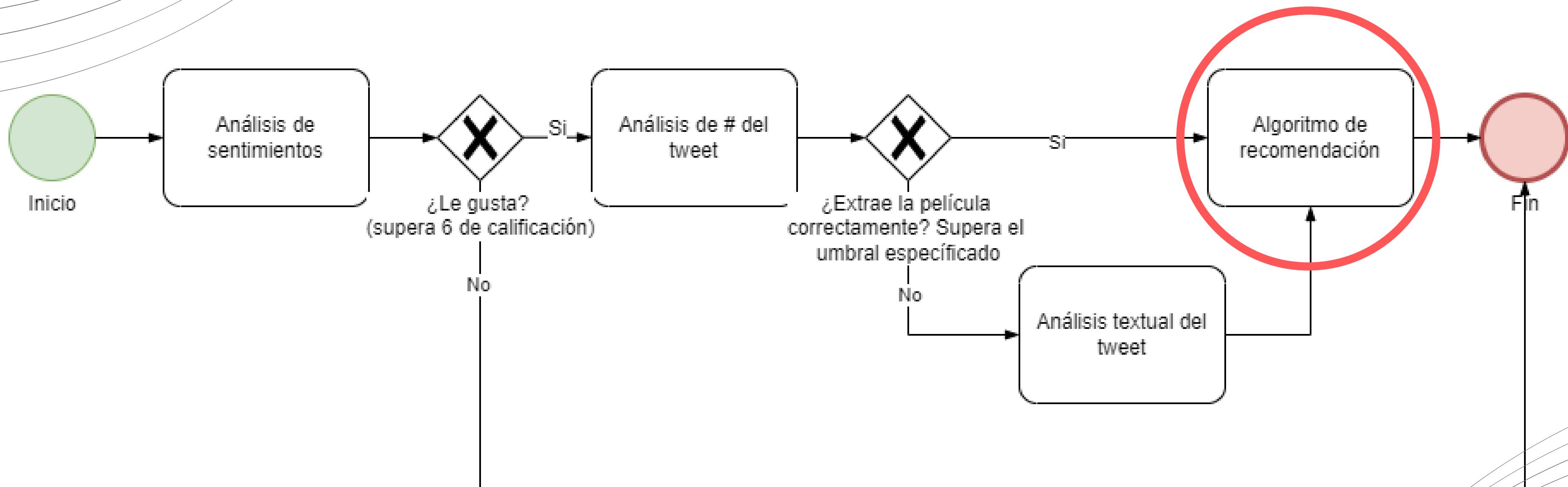
array(['Gisaengchung', '¡Shazam! La furia de los dioses', 'El astronauta',
       'La Liga de la Justicia de Zack Snyder', 'John Wick 4',
       'The iron claw', 'Gisaengchung', 'El imperio del fuego',
       'El Padrino', 'John Wick 4', 'Persona', 'Madame Web',
       'Gisaengchung', 'Apocalipse Now', 'La película de Lego',
       'Perfectos desconocidos', 'Maleficio', 'Eliminado',
       'La película de Lego', 'Persona', 'Fenomeno Secuestro'],
      dtype=object)
```

```
from sklearn.metrics import accuracy_score
y_pred = knn_pelicula.predict(X_peliculas_tweets_bow)
accuracy = accuracy_score(y_peliculas_tweets, y_pred)
Rendimiento=accuracy*21
```

Rendimiento

12.0

PROCESO BPMN DE CLASIFICACIÓN



ALGORITMO DE RECOMENDACIÓN

- A partir de las películas predichas obtener el género comparandolo con el dataset de películas

	Titulo_Predicho	Recomendable	Genre
0	Gisaengchung	positivo-Recomendable	Comedy, Drama, Thriller
1	¡Shazam! La furia de los dioses	positivo-Recomendable	Action, Adventure, Fantasy
2	El astronauta	neutro/negativo-noRecomendable	Drama, Sci-Fi
3	Zack Snyder's Justice League	neutro/negativo-noRecomendable	Action, Adventure, Fantasy
4	John Wick 4	neutro/negativo-noRecomendable	Action, Crime, Thriller

ALGORITMO DE RECOMENDACIÓN

Preprocesamiento dataset películas

- Añadir un movield al dataset de peliculas --> Para poder identificarlas a la hora de la recomendación
 - Hacer un onehotencoding para el genero de tal manera que ponga 1 si la pelicula tiene ese genero y 0 si no lo tiene

movielid	Series_Title	Series_Title_Spanish	Genre	Title-Dependiente	Title-Independiente	Title-DependienteEnglish	Drama	Crime	Drama	...	Western	Mystery	H...
0	1	The Shawshank Redemption	Cadena perpetua	[Drama]	Cadena perpetua	Cadena perpetua	The Shawshank Redemption	1.0	NaN	NaN	...	NaN	NaN
1	2	The Godfather	El Padrino	[Crime, Drama]	El Padrino	El Padrino	The Godfather	NaN	1.0	1.0	...	NaN	NaN
2	3	The Dark Knight	El caballero oscuro	[Action, Crime, Drama]	El caballero oscuro	El caballero oscuro	The Dark Knight	NaN	NaN	1.0	...	NaN	NaN
3	4	The Godfather: Part II	El Padrino: Parte II	[Crime, Drama]	El Padrino: Parte II	El Padrino: Parte II	The Godfather: Part II	NaN	1.0	1.0	...	NaN	NaN
4	5	12 Angry Men	12 hombres enojados	[Crime, Drama]	12 hombres enojados	12 hombres enojados	12 Angry Men	NaN	1.0	1.0	...	NaN	NaN

ALGORITMO DE RECOMENDACIÓN

- Filtrar las filas del dataset de películas de tal manera que coincida con las películas predichas
- Nos quedamos con la columna de movieId, película en español y géneros

	movieId	Series_Title_Spanish	Genre	Drama	Crime	Drama	Action	Crime	Adventure	Biography	...	Western	Mystery	Horror	Musical	Film-Noir	Sport
0	2	El Padrino	[Crime, Drama]	NaN	1.0	1.0	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
1	20	Gisaengchung	[Comedy, Drama, Thriller]	NaN	NaN	1.0	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
2	20	Gisaengchung	[Comedy, Drama, Thriller]	NaN	NaN	1.0	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
3	75	Apocalipsis ahora	[Drama, Mystery, War]	1.0	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
4	1060	Apocalypse Now	[Drama, War]	1.0	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN

ALGORITMO DE RECOMENDACIÓN

- Nos quedamos únicamente con la lista de géneros y si es recomendable o no la película

```
#seguimos limpiando info innecesaria
entrada_peliculas = entrada_peliculas.reset_index(drop = True)
tabla_generos = entrada_peliculas.drop(columns=['movieId','Series_Title_Spanish','Series_Title',
                                                'Genre', 'Titulo_Predicho', 'Title-Dependiente', 'Title-Independiente',
                                                'Title-DependienteEnglish'])
tabla_generos
```

- Asociamos el valor 1 si la pelicula es recomendable y 0 si la pelicula no es recomendable

```
# Convertir la columna 'Recomendable' a valores numéricos
tabla_generos['Recomendable_Num'] = tabla_generos['Recomendable'].map({'positivo-Recomendable': 1,
                                                               'neutro/negativo-noRecomendable': 0})
tabla_generos
```

ALGORITMO DE RECOMENDACIÓN

- Modelamos el algoritmo mediante una matriz de peso --> Se suma 1 por cada genero que le gusta

```
#ya vamos a empezar a modelar nuestro algoritmo, para ello creamos la matriz de peso que basicamente es un producto punto
#llamamos la funcion DOT de la libreria pandas para hacer esto
gustos_usuario = tabla_generos[tabla_generos['Recomendable_Num'] == 1]

# Suma de generos
gustos_usuario = gustos_usuario.drop('Recomendable_Num', axis=1).sum()
```

gustos_usuario

Drama	1.0
Crime	0.0
Drama	2.0
Action	2.0
Crime	1.0
Adventure	1.0
Biography	0.0
History	0.0
Sci-Fi	0.0
Romance	0.0

- Se realiza el mismo proceso para quedarse únicamente con los géneros en el dataset de películas

ALGORITMO DE RECOMENDACIÓN

- Se realiza un promedio ponderado entre los gustos del usuario y las películas del dataset

```
#ahora realizamos el promedio ponderado para recomendar 20 películas que hayan nuestra base de datos
peliculas_recomendadas = ((generos*gustos_usuario).sum(axis=1))/(gustos_usuario.sum())
peliculas_recomendadas.head()
```

```
movieId
1    0.058824
2    0.117647
3    0.294118
4    0.117647
5    0.117647
dtype: float64
```

- Ordenamos en orden descendente para saber que películas se ajustan mas a los gustos del usuario

ALGORITMO DE RECOMENDACIÓN

- Por último, recoger las 20 películas con mayor puntuación y buscamos el nombre de la película mediante el movieId

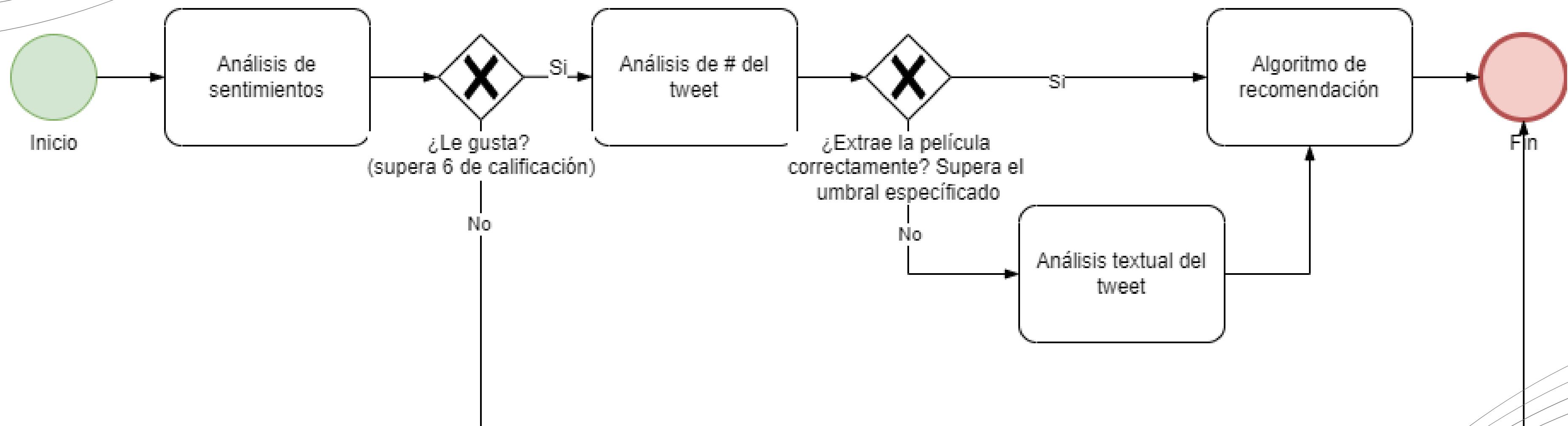
```
#tabla de recomendaciones
nombre_peliculas_recomendadas = dt_BD_peliculas.loc[dt_BD_peliculas['movieId'].isin(peliculas_recomendadas.head(20).keys())]
nombre_peliculas_recomendadas = nombre_peliculas_recomendadas[['Series_Title_Spanish']]
#peliculas recomendadas
```

```
nombre_peliculas_recomendadas
```

Resultado: 20 películas recomendadas por género.

Series_Title_Spanish	
19	Gisaengchung
49	Psicópata
181	Yojinbo
204	Relatos salvajes
353	Nefes: Vatan Sagolsun
362	El ultimátum de Bourne
368	Casino Royale
369	Matar a Bill: vol. 2
511	La identidad Bourne
570	raazi
641	Gongdong gyeongbi guyeok JSA
648	Los elegidos
777	La supremacía de Bourne
826	Barton Fink
847	La toma de Pelham uno dos tres
932	Sierra
948	Los demás
963	Morir duro: con venganza
979	Arma letal
983	Los guerreros

PROCESO BPMN DE CLASIFICACIÓN



PUESTA EN PRODUCCIÓN

prediccion_final('Miguel Araiza @miguelaraizac · 19 mar. Acaban de agregar Drive My Car a Netflix. Uno de los mejores retratos sc
◀ ▶ 100% | 1/1 [00:00<00:00, 73.48it/s]

Tiempo de ejecución: 0.0166015625 segundos

```
[ 'positivo-Recomendable',
  'Gisaengchung',
  'Recomienda?: positivo-Recomendable, por la segunda capa la pelicula es : Gisaengchung, a una distancia de: [[16.46207763]],
674      Dip huet seung colgado
724                  Salir
368          Casino Royale
635      Caminar por la linea
659          Lápida sepulcral
717          Sopa de pato
810          Brillar
1055     El padrino 2
826          Barton Fink
277          vida de brian
Name: Series_Title_Spanish, dtype: object]
```

**GRACIAS
A TODOS**

