




TruthFinder

“In a world of lies, the truth is your only weapon”



Alex Nakamura Díaz Francés
David Hernández Uriostegui
Diego Javier Padilla Lara
Erick Daniel Arroyo Martínez
Sebastián Alamina Ramírez

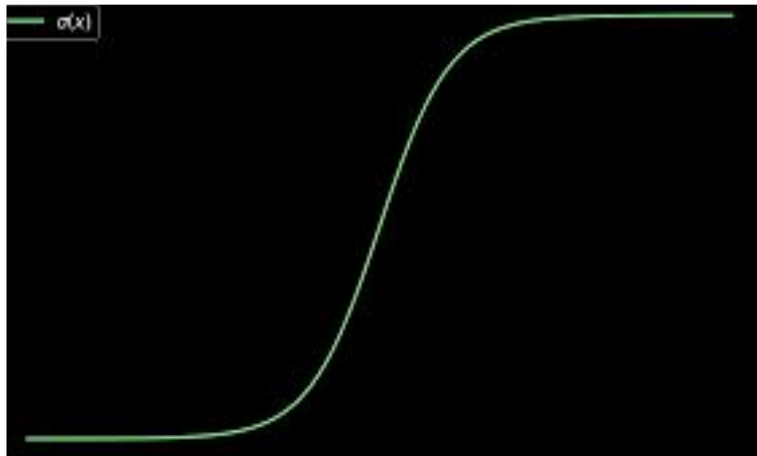
INTRODUCCIÓN

- Las **noticias falsas** son un tipo de periodismo amarillista que consta de piezas de noticias que suelen ser engaños transmitidos a través de las redes sociales y otros medios en línea.
- Contienen afirmaciones falsas y/o exageradas, y son fácilmente viralizadas por algoritmos, por lo que es importante detectarlas de manera eficiente.
- Compararemos el desempeño de diferentes estrategias y métodos para atacar el problema.

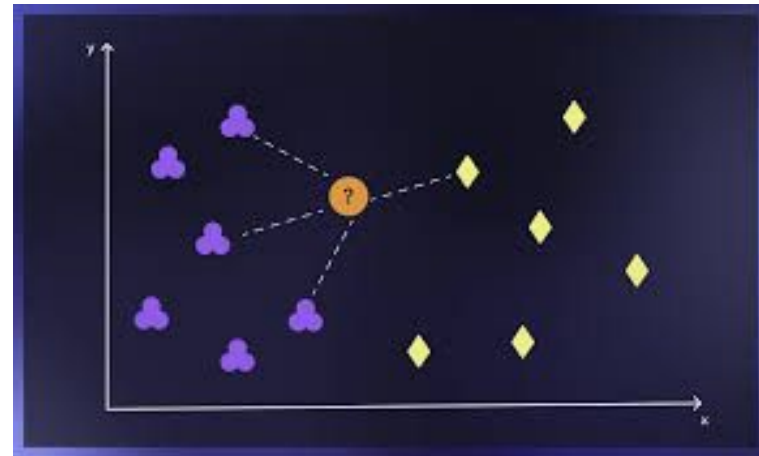


PREGUNTA DE INVESTIGACIÓN

¿Cuál es el rendimiento comparativo de la regresión logística y el algoritmo de K nearest neighbors para la detección y clasificación de noticias falsas en un conjunto de datos etiquetado para la detección y clasificación de Fake News en medios sociales, en términos de precisión y eficiencia en la clasificación?



VS



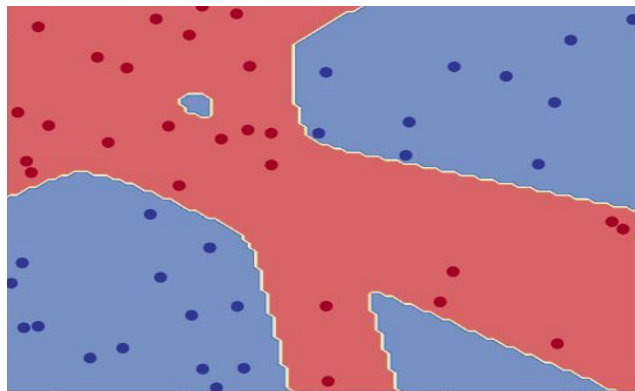
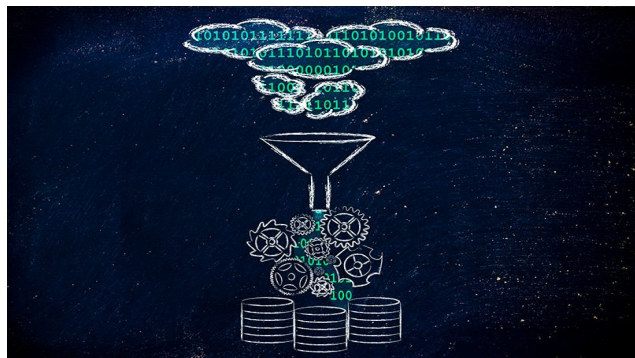
OBJETIVOS

- Comparar estrategias de preprocesamiento.
- Mostrar mejor combinación de preprocesamiento + clasificación.
- Contrastar el uso de la reducción de dimensiones con su ausencia.



METODOLOGÍA

- Preprocesamiento de datos
 - Selección de datos
 - Limpieza de datos
 - Vectorización
 - Reducción dimensional
 - Entrenamiento y prueba de modelos.
 - Evaluación de modelos
- Algoritmos de clasificación
 - Regresión Logística y KNN
 - Regresión Logística y KNN + TFIDF
 - Regresión Logística y KNN + CountVectorizer
 - Regresión Logística y KNN + (TFIDF y SVD)
 - Regresión Logística y KNN + (CountVectorizer y SVD)
 - Regresión Logística + (Countvectorizer y Word2Vec)
 - Regresión Logística + (Countvectorizer y Word2Vec y SVD)




DATASET

<https://www.kaggle.com/datasets/jainpooja/fake-news-detection>

Data Explorer

Version 1 (116.37 MB)

 Fake.csv

 True.csv

Summary

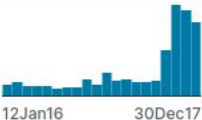
▼		2 files	
		.csv	2
▼		8 columns	
		String	6
		DateTime	2

DATASET

<https://www.kaggle.com/datasets/jainpooja/fake-news-detection>

True.csv (53.58 MB)

DetailCompactColumn

▲ title	▲ text	▲ subject	📅 date
20826 unique values	21192 unique values	politicsNews 53% worldnews 47%	
House Speaker Ryan mulls retirement after 2018 elections: Politico	WASHINGTON (Reuters) - Republican House Speaker Paul Ryan has told confidants he would like to retir...	politicsNews	December 14, 2017
Ford messages back	NEW YORK (Reuters)	politicsNews	December

Summary

2 files

.csv

2

8 columns

String

6

DateTime

2

Data Explorer

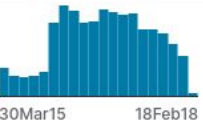
Version 1 (116.37 MB)

Fake.csv

True.csv

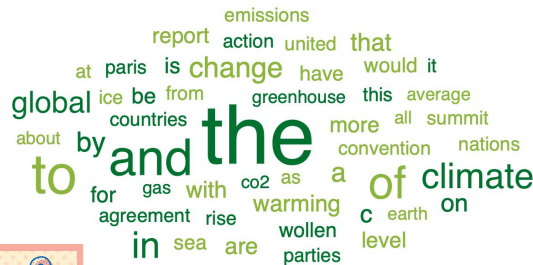
Fake.csv (62.79 MB)

DetailCompactColumn

▲ title	▲ text	▲ subject	📅 date
17903 unique values	[empty] 3% AP News The regul... 0% Other (22851) 97%	News 39% politics 29% Other (7590) 32%	
Donald Trump Sends Out Embarrassing New Year's Eve Message; This is Disturbing	Donald Trump just couldn't wish all Americans a Happy New Year and leave it at that. Instead, he had...	News	December 31, 2017
Break: Breaking Trump	House Intelligence	News	December 31, 2017

LIMPIEZA DE LOS DATOS

- Símbolos especiales y stopwords.
- Palabras de cierta longitud.
- Lematización.
- Tokenización.
- Palabras por frecuencia



“ ”



"This is the first step in the NLP pipeline"

Tokenizer

'This' 'is' 'the' 'first' 'step' 'in' 'the' 'NLP' 'pipeline'

Tokenize on rules

Let 's tokenize ! Is n't this easy ?

Tokenize on punctuation

Let ' s tokenize ! Isn ' t this easy ?

Tokenize on white spaces

Let's tokenize! Isn't this easy?

Let's tokenize! Isn't this easy?

VECTORIZACIÓN

CountVectorizer y TfidfVectorizer

Count Vectorizer

	blue	bright	sky	sun
Doc1	1	0	1	0
Doc2	0	1	0	1

TD-IDF Vectorizer

	blue	bright	sky	sun
Doc1	0.707107	0.000000	0.707107	0.000000
Doc2	0.000000	0.707107	0.000000	0.707107

CountVectorizer

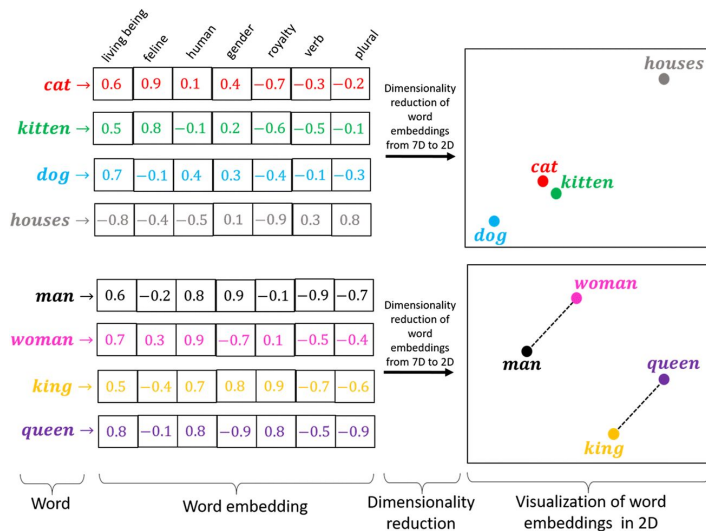
- Crea una matriz donde cada fila representa un documento y cada columna representa una palabra del vocabulario.
- Los valores en la matriz indican la cantidad de veces que una palabra específica aparece en cada documento.
- No tiene en cuenta la importancia relativa de las palabras en el texto.

TfidfVectorizer

- Calcula un peso que refleja la importancia de una palabra en un documento específico y en todo el texto.
- Multiplica la frecuencia de una palabra en un documento por el inverso de la frecuencia de documentos que contienen esa palabra.
- Genera una matriz donde los valores indican el peso de cada palabra en el texto.
- Resalta las palabras clave importantes y filtra palabras comunes.

Word2Vec

- Word2Vec es un algoritmo de aprendizaje automático para generar representaciones vectoriales de palabras.
- **Objetivo principal:** capturar relaciones semánticas y sintácticas entre palabras en un corpus de texto.
- Representa cada palabra como un vector denso en un espacio vectorial.
- Los vectores capturan la información contextual y semántica de las palabras.



ESTRATEGIAS DE REDUCCIÓN DE DIMENSIÓN

SVD

Funciona descomponiendo una matriz de datos en tres matrices más pequeñas, permitiendo así representar los datos originales de manera más compacta. Al encontrar los patrones básicos en los datos y descartar los menos importantes, SVD puede ayudar a simplificar los datos y facilitar el trabajo con ellos.

Singular decomposition
analysis(SVD)

$$\boxed{C_{m \times n}} = \boxed{U_{m \times r}} \times \boxed{\Sigma_{r \times r}} \times \boxed{V_{r \times n}^1}$$

TruncatedSVD ¿En qué difiere con SVD?

La principal diferencia entre la **SVD** y **TruncatedSVD** es que la SVD factoriza una matriz en tres matrices. De igual manera, **TruncatedSVD** es una variación de la SVD que conserva sólo un subconjunto de los valores singulares y los vectores singulares asociados, lo que da lugar a una aproximación de menor rango de la matriz original.

PCA vs Truncated SVD

PCA vs Truncated SVD

Truncated SVD

1. Utiliza la descomposición de valores singulares de una matriz para extraer componentes principales.
2. Las componentes son ortogonales y no están relacionadas con la varianza de los datos, sino con la cantidad de información capturada.
3. Es más eficiente computacionalmente, especialmente para matrices grandes o dispersas.

PCA vs Truncated SVD

Truncated SVD

1. Utiliza la descomposición de valores singulares de una matriz para extraer componentes principales.
2. Las componentes son ortogonales y no están relacionadas con la varianza de los datos, sino con la cantidad de información capturada.
3. Es más eficiente computacionalmente, especialmente para matrices grandes o dispersas.

PCA

1. Encuentra las componentes principales que explican la mayor varianza en los datos originales.
2. Las componentes son combinaciones lineales de las características originales.
3. Tiende a preservar la varianza total de los datos originales.
4. Es adecuado cuando se busca retener la mayor cantidad de información posible al reducir la dimensionalidad.

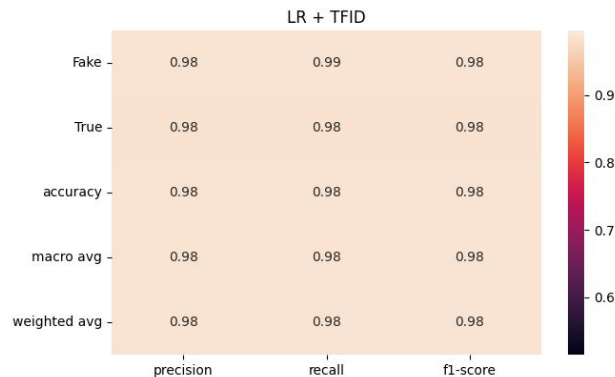
¿Por qué TruncatedSVD en lugar de PCA?

¿Por qué TruncatedSVD en lugar de PCA?

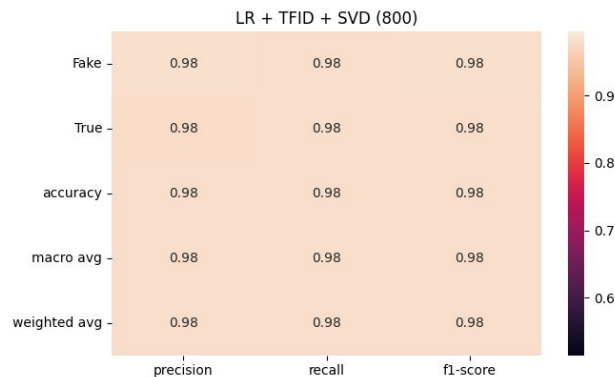
COSTO COMPUTACIONAL BASTANTE ALTO

Resultados preliminares: LR + TFIDF

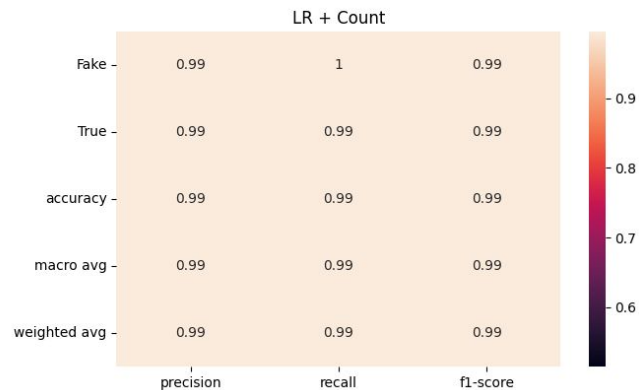
No SVD



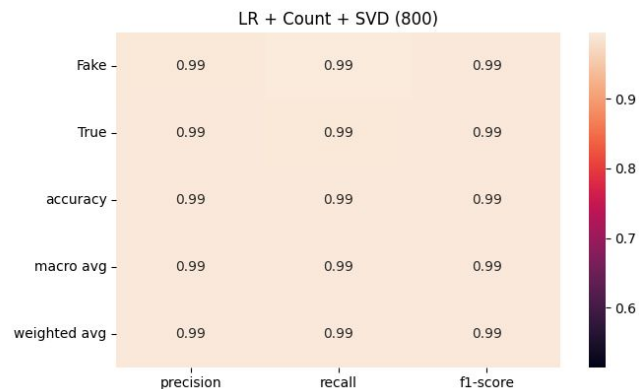
SVD



Resultados preliminares: LR + CountVectorizer



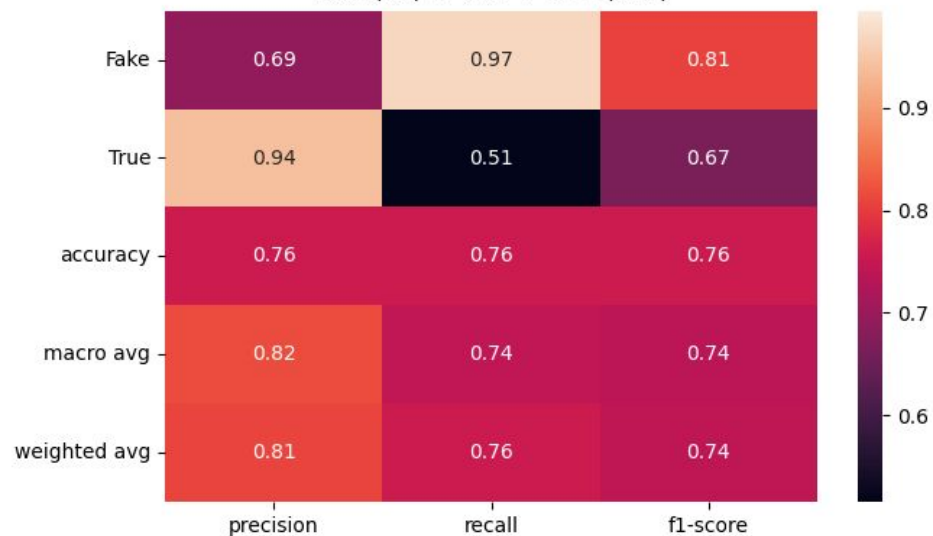
No SVD



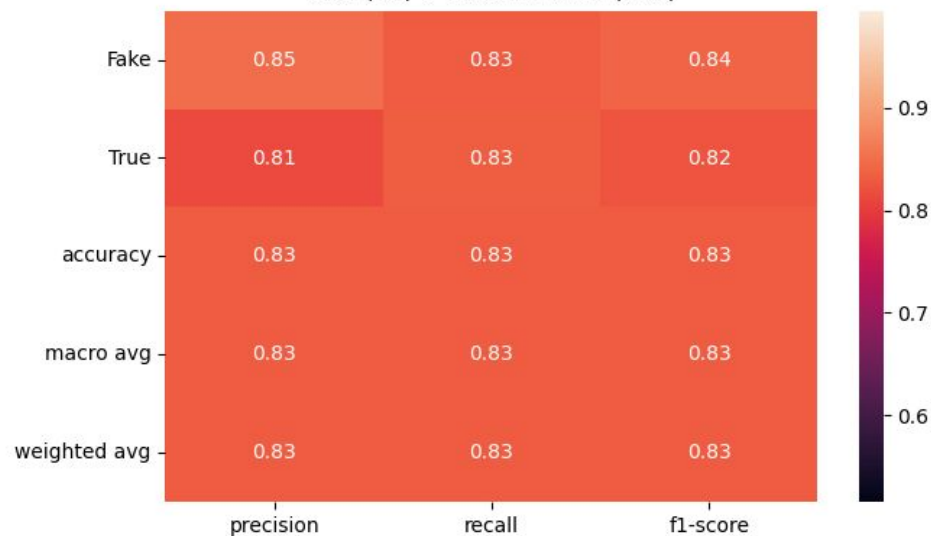
SVD

Resultados preliminares: KNN

KNN (50) + TFID + SVD (800)



KNN (50) + Count + SVD (800)



Regresión Logística + Count + Word2Vec

¿Qué podría salir mal?

¿Qué riesgos y beneficios asumimos?

¿Era necesario?

¿Por qué no TFIDF y Word2Vec?

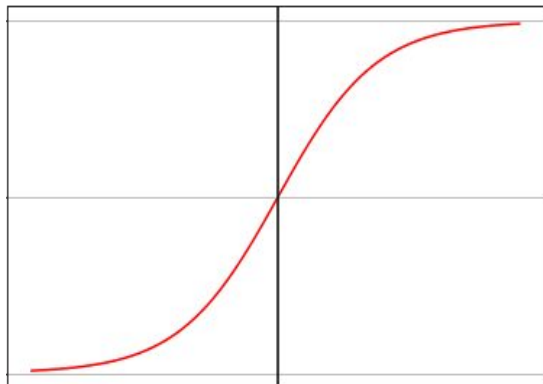
¿Lo sabemos todo?

¿Se siguió algún orden?

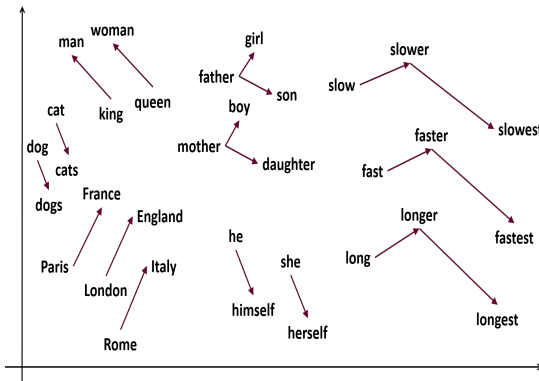
¿Por qué no KNN?

¿Por qué Regresión Logística?

¿Por qué Count y Word2Vec?



+



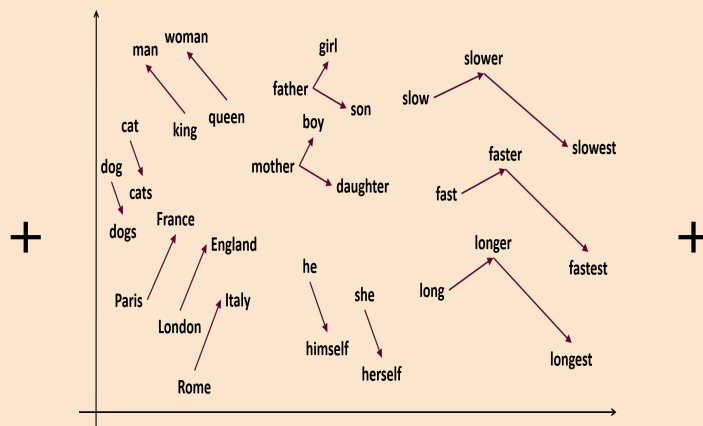
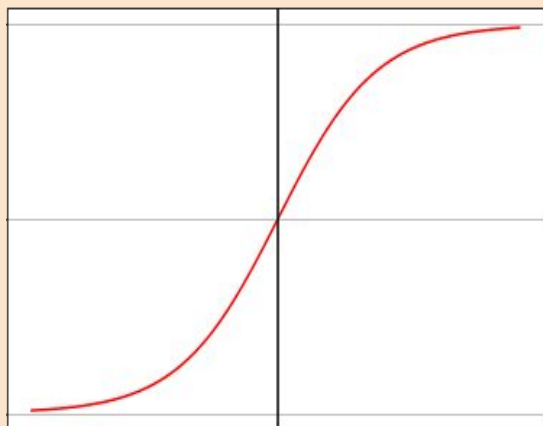
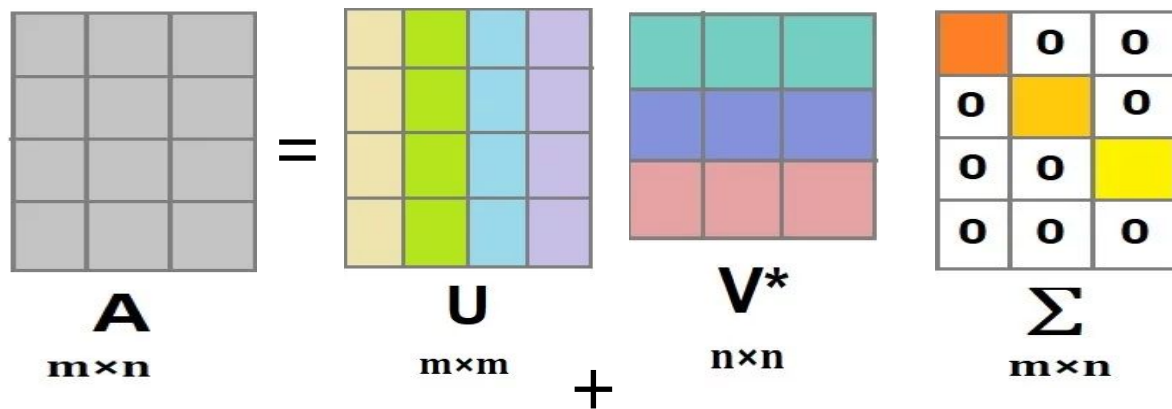
+

Data = ['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']



	The	quick	brown	fox	jumps	over	lazy	dog
Data	2	1	1	1	1	1	1	1

Regresión Logística + Count + Word2Vec + SVD



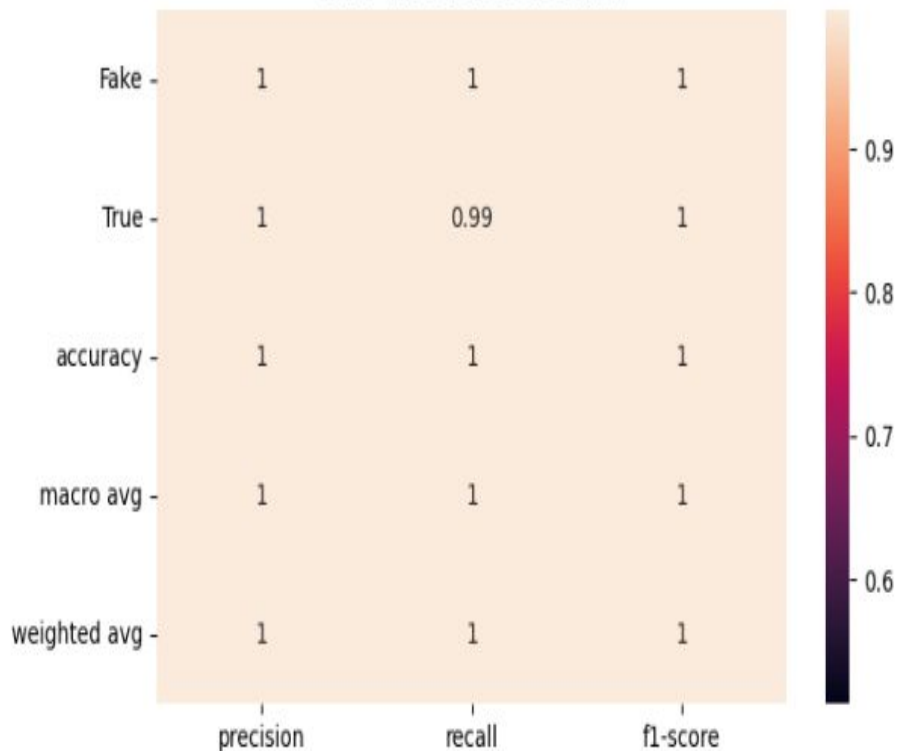
Data = ['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']



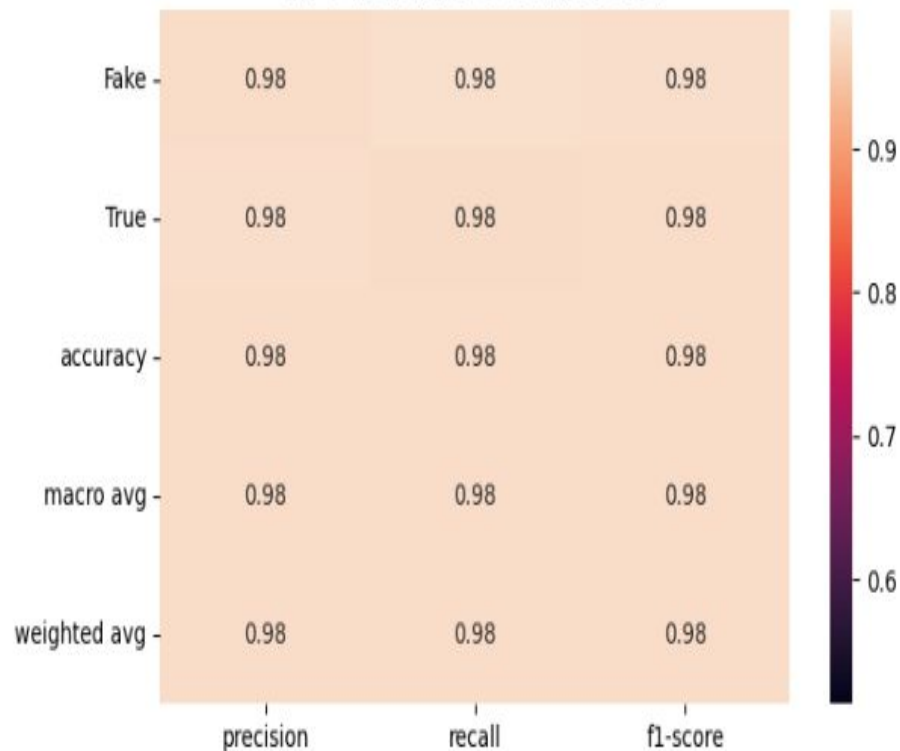
	The	quick	brown	fox	jumps	over	lazy	dog
Data	2	1	1	1	1	1	1	1

Resultados

LR + (Word2Vec & Count)



LR + (Word2Vec & Count & SVD)



Probando diferentes valores para k en KNN

- Observar el comportamiento para valores de k mayores a 50

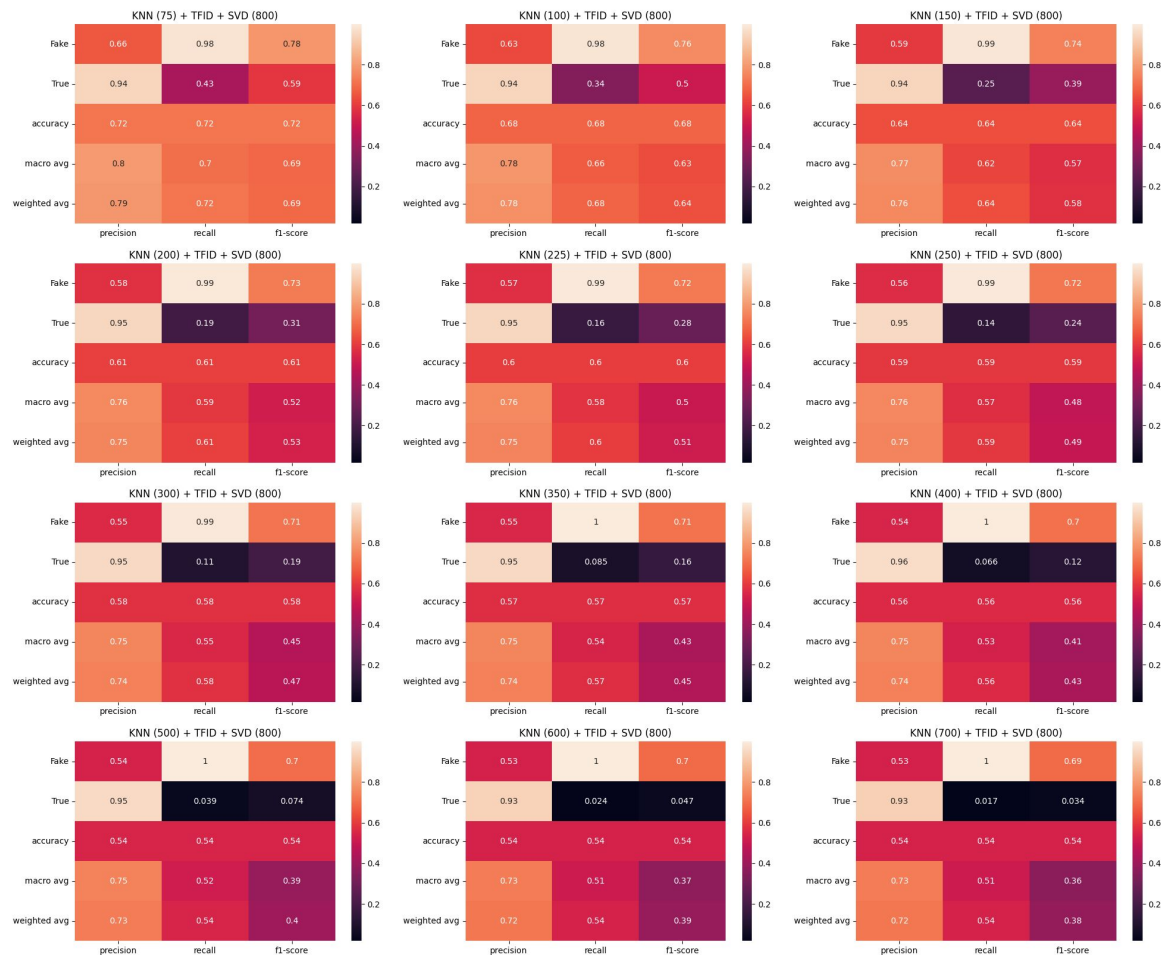
{75, 100, 150, 200, 225, 250, 300, 350, 400, 500, 600, 700}

- Probado para TFID + SVD y para CountVectorizer + SVD

¿Cuál fue el comportamiento observado?

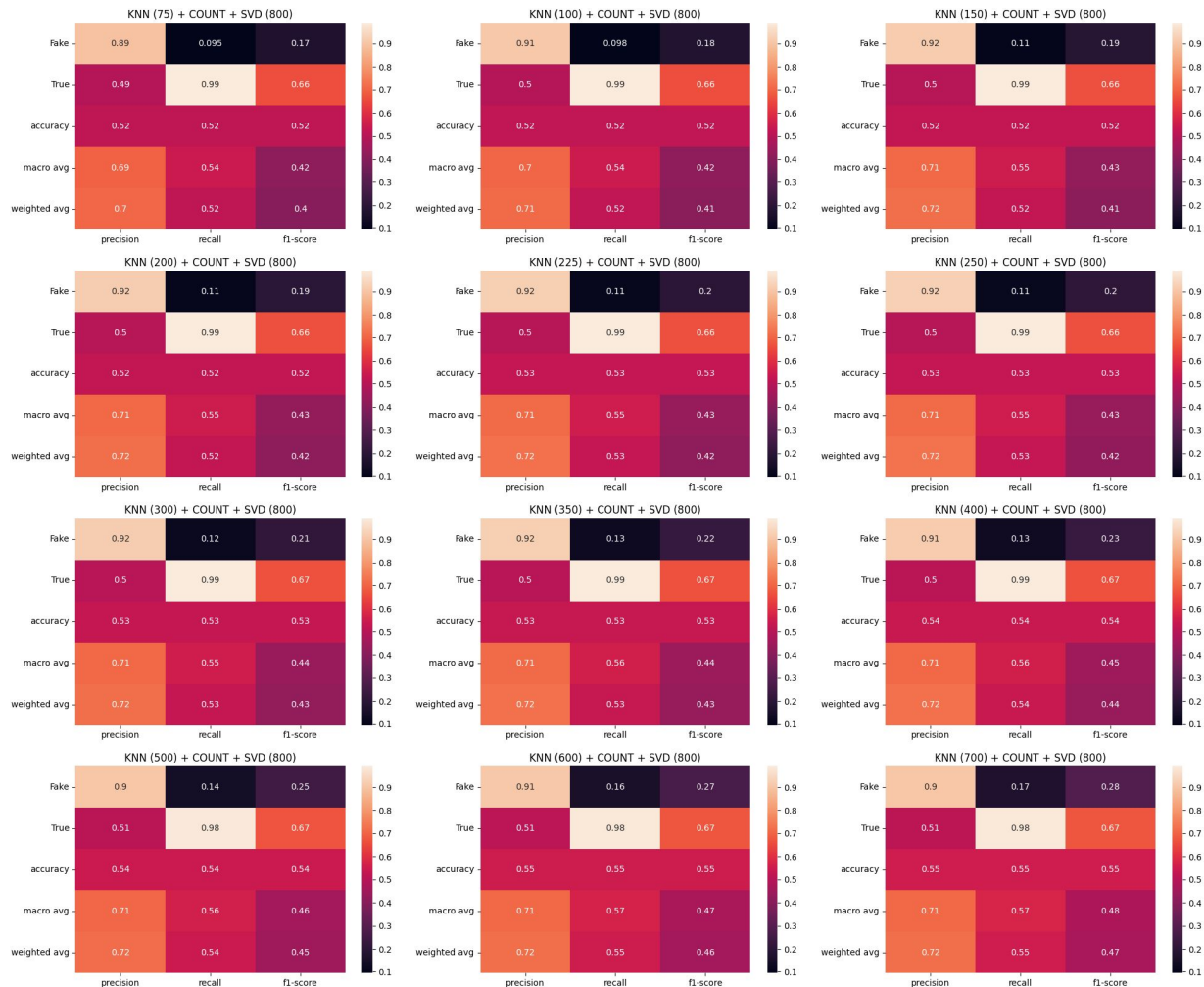
Resultados

TFDI + SVD



Resultados

CountVectorizer+SVD



CONCLUSIONES

Algoritmo	Vectorización	Reducción	Exactitud	Precisión	recall	F1
RL	Count+Word2Vec	--	1	1	1	1
RL	Count+Word2Vec	SVD	0.98	0.98	0.98	0.98
RL	Count	--	0.99	0.99	0.99	0.99
RL	TFIDF	--	0.98	0.98	0.98	0.98
KNN	Count	SVD	0.83	0.83	0.83	0.83
KNN	TFIDF	SVD	0.76	0.76	0.76	0.76