Laboratorio 5: Minería de Texto y Clasificación de Tweets

CC3084 – Data Science, Semestre II 2025

Equipo: *(Diederich solis 22952, Sara Guzma)*

Universidad del Valle de Guatemala

1 de septiembre de 2025

Índice

1. Resumen

Este informe describe la construcción de un pipeline de minería de texto para clasificar tweets del conjunto *Natural Language Processing with Disaster Tweets* de Kaggle en las clases **desastre real (1)** y **no desastre (0)**. Se documentan el preprocesamiento, el análisis exploratorio (nubes de palabras, frecuencias y n-gramas), el entrenamiento y evaluación de varios modelos (Naive Bayes, Regresión Logística, SVM lineal y Random Forest), la función para clasificar tweets nuevos, el análisis de sentimiento (positivo/neutral/negativo) y el reentrenamiento con una nueva variable de *negatividad*. El mejor desempeño se obtuvo con un **SVM lineal (C=0.5)** usando **TF–IDF con uni- y bi-gramas**.

2. Requerimientos del laboratorio

La guía solicita: cargar datos, limpiar y preprocesar texto; analizar frecuencias y n-gramas; entrenar varios modelos y explicar el manejo de contexto; implementar una función que clasifique un tweet nuevo; medir sentimiento y responder preguntas sobre los 10 más positivos/negativos y su distribución por categoría; crear una variable de *negatividad* y reentrenar el mejor modelo para evaluar si mejora su desempeño.

3. Datos

Fuente: Kaggle – *NLP with Disaster Tweets*.
Tamaño: ∼10,500 filas, 5 columnas.
Campos: `id`, `keyword`, `location`, `text`, `target` (1: desastre, 0: no desastre).

4. Preprocesamiento

Se aplicó la siguiente **limpieza y normalización** sobre el texto (manteniendo la reproducibilidad en el notebook):

- Conversión a `lowercase`.
- Remoción de URLs (`http[s]://`, `www.`).
- Remoción de menciones `@usuario`.
- Conservación de la palabra en #hashtags ($\#word \rightarrow word$).
- Eliminación de puntuación y números (con excepción de un tratamiento especial para `911`, mapeado a una forma legible).
- Colapso de espacios en blanco.
- Para el **análisis de sentimiento**, se preservaron **emojis/emoticones** y solo se quitaron URLs/mentions para no perder carga afectiva.

Representación: TF–IDF con **unigramas y bigramas** (n-gramas de 1 y 2) para capturar contexto corto (*"forest fire"*, *"evacuation order"*, etc.).

5. Análisis exploratorio & n-gramas

5.1. Frecuencia de palabras y nubes

Incluimos conteos globales y por clase, así como nubes de palabras para resaltar términos de alta frecuencia.

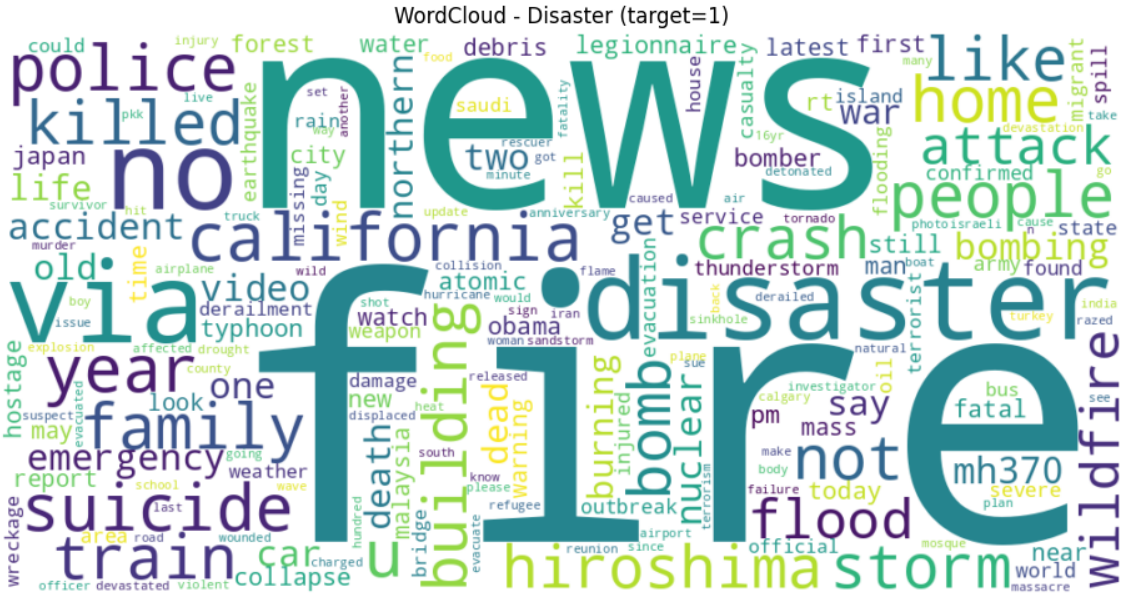


Figura 1: Nube de palabras para la clase **desastre** (1).

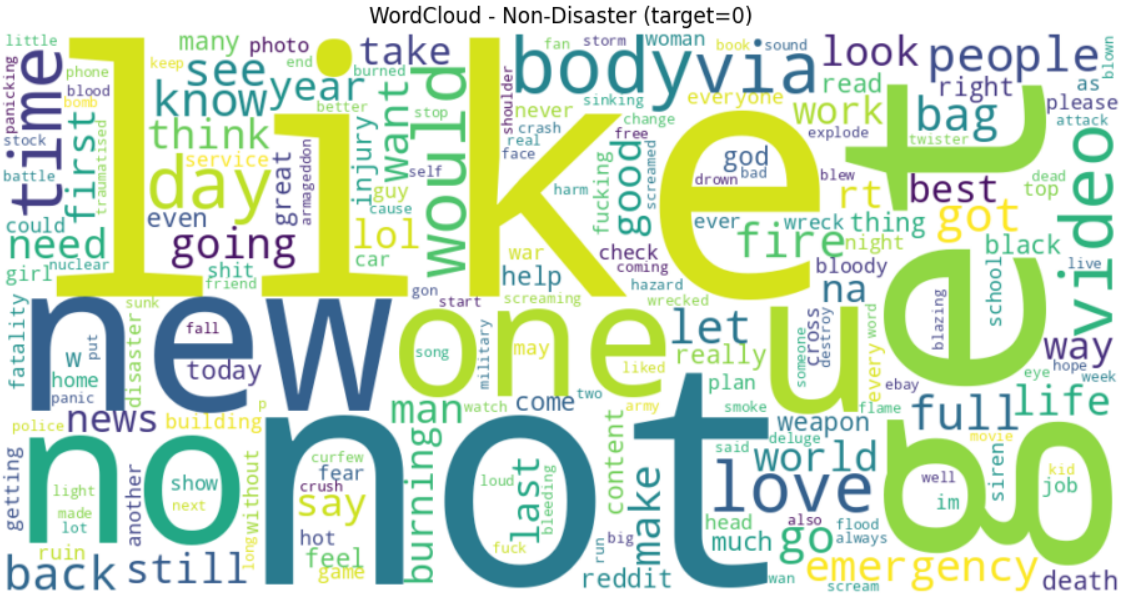


Figura 2: Nube de palabras para la clase **no desastre** (0).

5.2. Histogramas de términos y discusión

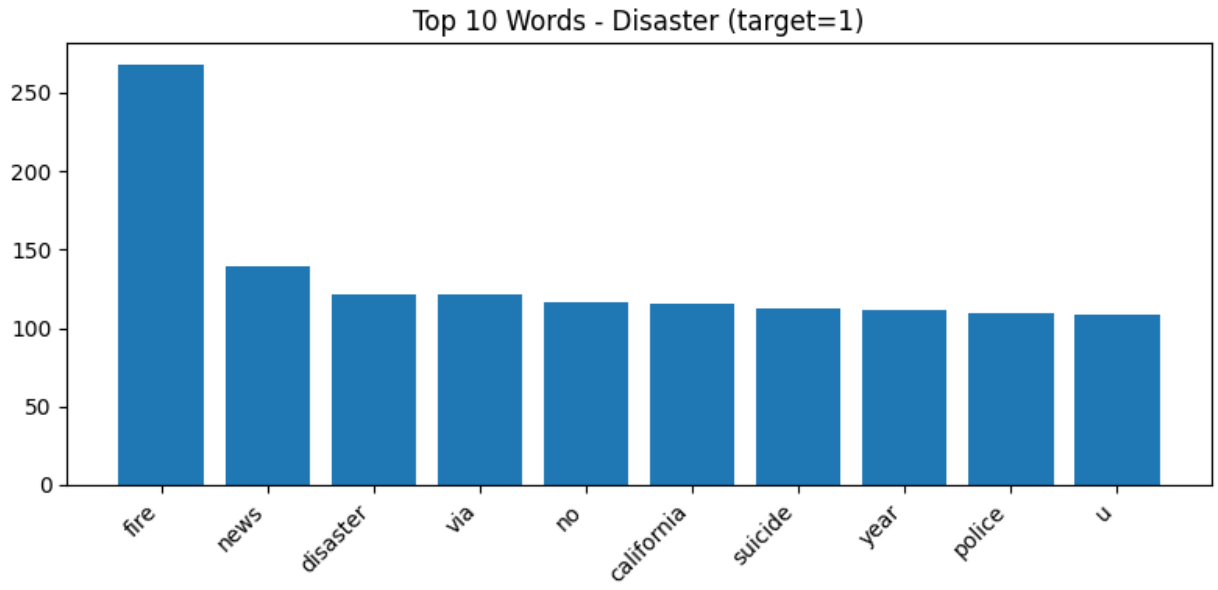


Figura 3: Top 10 palabras — clase **desastre** (target=1).

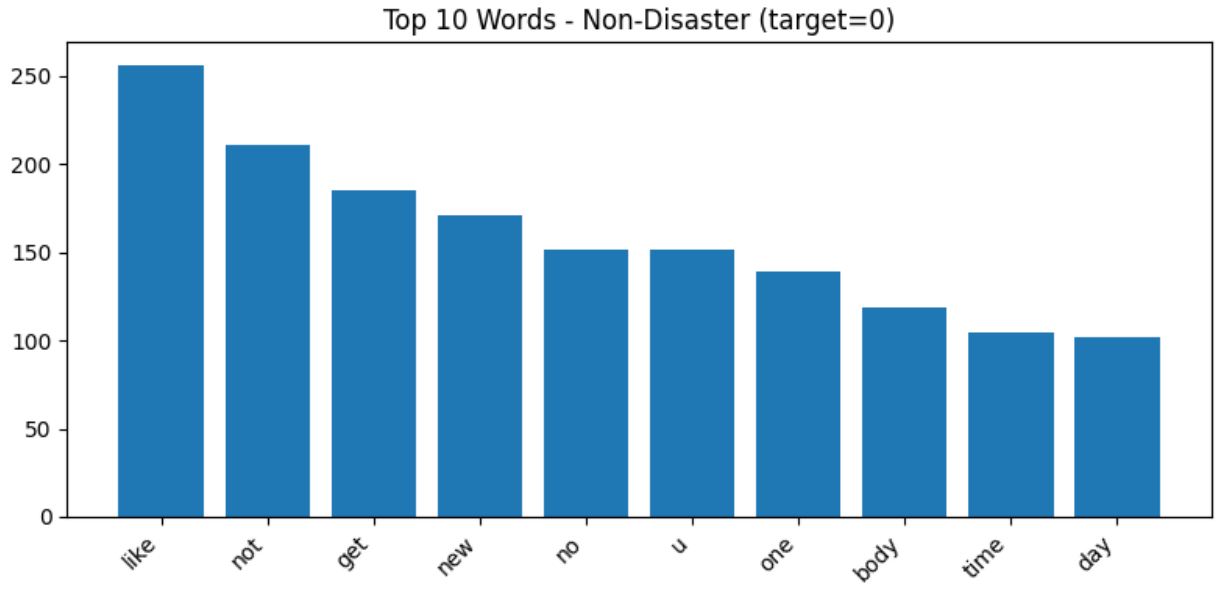


Figura 4: Top 10 palabras — clase **no desastre** (target=0).

6. Modelos de clasificación (Punto 6)

Se compararon los siguientes modelos con la misma representación TF–IDF (1–2-gramas) y **validación cruzada (CV=5)** usando la métrica **F1**:

- **Multinomial Naive Bayes** (suavizado $\alpha \in \{0{,}5, 1{,}0\}$).
- **Regresión Logística** (penalización L2, $C \in \{0{,}5, 1{,}0, 2{,}0\}$; solvers `liblinear`/`lbfgs`).
- **Linear SVM** ($C \in \{0{,}5, 1{,}0, 2{,}0\}$).
- **Random Forest** (300 árboles, max_depth $\in \{\text{None}, 20, 40\}$).

Mejor modelo: **LinearSVM** con $C = 0{,}5$. Métricas en el conjunto de prueba (80/20 estratificado):

Cuadro 1: Reporte de clasificación del mejor modelo

Clase	Precisión	Recall	F1	Soporte
0 (no desastre)	0.8206	0.8631	0.8413	869
1 (desastre)	0.8046	0.7492	0.7759	654
Exactitud		0.8142		
Promedio macro	0.8126	0.8061	0.8086	1523
Promedio ponderado	0.8137	0.8142	0.8132	1523

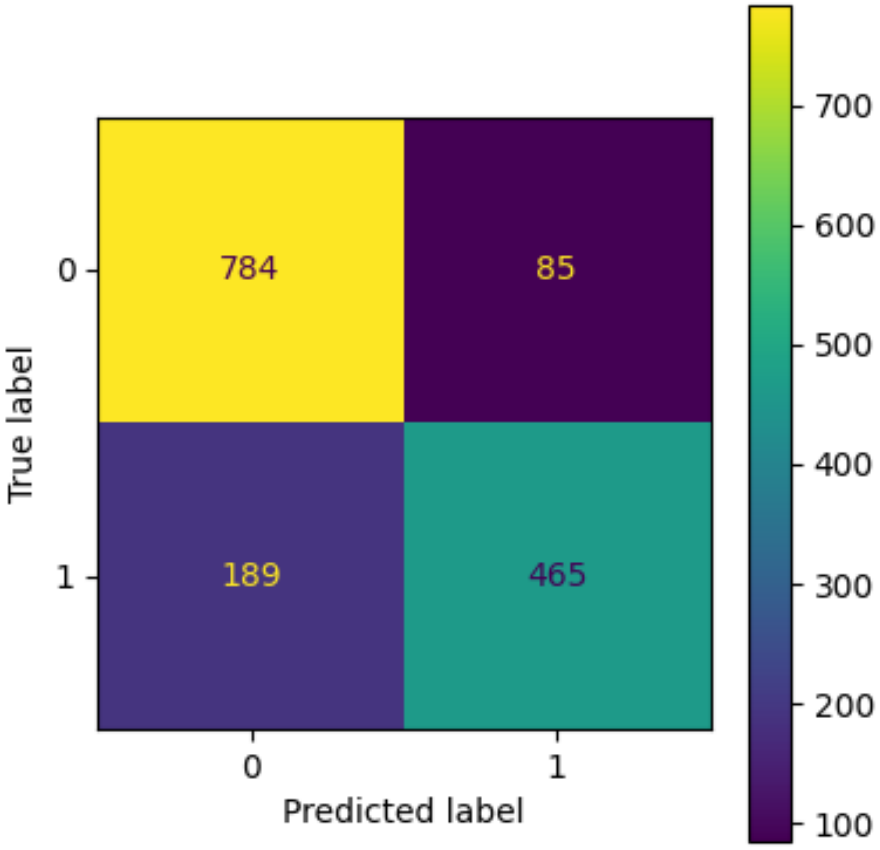


Figura 5: Matriz de confusión del mejor modelo (SVM lineal).

Características más asociadas (modelo lineal): términos como *hiroshima*, *fires*, *storm*, *floods*, *wildfire*, *earthquake*, *tornado* y *bombing* fueron pro-clase 1; mientras que *you*, *my*, *show*, *finally*, *cake* y *never* lo fueron para clase 0.

7. Función de clasificación (Punto 7)

Se implementó `clasificar_tweet(texto)` que aplica el **mismo preprocesamiento** del entrenamiento y utiliza el pipeline TF–IDF + SVM para predecir. Para SVM se reporta una *confianza* monotónica basada en la función de decisión (no calibrada).

Ejemplos (texto simplificado para compatibilidad con LaTeX):

Tweet	Predicción	Confianza
Wildfire spreading near the hills, people evacuating now!	desastre	0.7693
This cake is a disaster [emoji] but tastes amazing lol	no desastre	0.2767
Earthquake reported 20 miles north of the city. Stay safe.	desastre	0.7094

8. Análisis de sentimiento (Punto 8)

Se utilizó **VADER** (lexicón + reglas) sobre el texto con **emojis/emoticones preservados**. Se emplearon los umbrales estándar: `compound` $\geq 0{,}05$ positivo, $\leq -0{,}05$ negativo, en otro caso neutral. Se añadieron columnas: `sent_neg`, `sent_neu`, `sent_pos`, `sent_compound`, `pos_count`, `neg_count`, `sent_label`.

Distribución global de sentimiento (conteos):

Sentimiento	Conteo
Negativo	3735
Neutral	1945
Positivo	1933

Crosstab sentimiento × categoría (0=no desastre, 1=desastre):

	Negativo	Neutral	Positivo
Target 0	1854 (42.7 %)	1092 (25.1 %)	1396 (32.2 %)
Target 1	1881 (57.5 %)	853 (26.1 %)	537 (16.4 %)

Los tweets de **desastre (1)** son sustancialmente más **negativos** que los de **no desastre (0)** (diferencia $\approx +0{,}148$ en proporción de negativos). Prueba χ^2: chi^2=265,885, gl=2, p=1.836e-58; **V de Cramér** = 0,187 (tamaño de efecto bajo–medio).

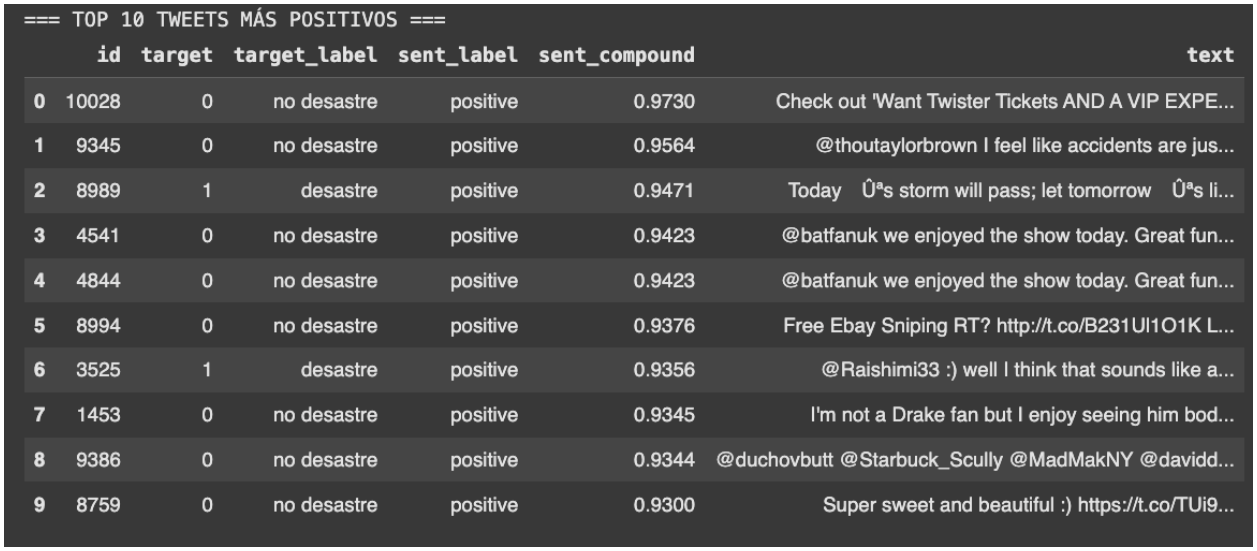
9. Top 10 negativos/positivos (Punto 9)

A continuación se muestran tablas con los tweets más negativos y positivos (según `compound`) y su categoría.

=== TOP 10 TWEETS MÁS NEGATIVOS ===

	id	target	target_label	sent_label	sent_compound	text
0	10689	0	no desastre	negative	-0.9883	wreck? wreck wreck wreck wreck wreck wreck wre...
1	9172	1	desastre	negative	-0.9686	@Abu_Baraa1 Suicide bomber targets Saudi mosqu...
2	9166	1	desastre	negative	-0.9623	Suicide bomber kills 15 in Saudi security site...
3	9137	1	desastre	negative	-0.9595	? 19th Day Since 17-Jul-2015 -- Nigeria: Suici...
4	9159	1	desastre	negative	-0.9552	17 killed in S ÛªArabia mosque suicide bombing...
5	4213	0	no desastre	negative	-0.9549	at the lake \n*sees a dead fish*\nme: poor lit...
6	682	1	desastre	negative	-0.9538	illegal alien released by Obama/DHS 4 times Ch...
7	2225	1	desastre	negative	-0.9524	Bomb Crash Loot Riot Emergency Pipe Bomb Nucle...
8	9765	1	desastre	negative	-0.9500	Bomb head? Explosive decisions dat produced mo...
9	9940	1	desastre	negative	-0.9493	@cspan #Prez. Mr. President you are the bigges...

Figura 6: Tabla: Top 10 tweets más **negativos** con *id*, *texto* y *target*.



=== TOP 10 TWEETS MÁS POSITIVOS ===

	id	target	target_label	sent_label	sent_compound	text
0	10028	0	no desastre	positive	0.9730	Check out 'Want Twister Tickets AND A VIP EXPE...
1	9345	0	no desastre	positive	0.9564	@thoutaylorbrown I feel like accidents are jus...
2	8989	1	desastre	positive	0.9471	Today ÛªÛªs storm will pass; let tomorrow ÛªÛªs li...
3	4541	0	no desastre	positive	0.9423	@batfanuk we enjoyed the show today. Great fun...
4	4844	0	no desastre	positive	0.9423	@batfanuk we enjoyed the show today. Great fun...
5	8994	0	no desastre	positive	0.9376	Free Ebay Sniping RT? http://t.co/B231Ul1O1K L...
6	3525	1	desastre	positive	0.9356	@Raishimi33 :) well I think that sounds like a...
7	1453	0	no desastre	positive	0.9345	I'm not a Drake fan but I enjoy seeing him bod...
8	9386	0	no desastre	positive	0.9344	@duchovbutt @Starbuck_Scully @MadMakNY @davidd...
9	8759	0	no desastre	positive	0.9300	Super sweet and beautiful :) https://t.co/TUi9...

Figura 7: Tabla: Top 10 tweets más **positivos** con *id*, *texto* y *target*.

10. Variable de "negatividad" y reentrenamiento (Punto 10)

Se definió la variable **negatividad** como `sent_neg` (componente negativa de VADER, en $[0, 1]$) y se concatenó como *feature* numérica al vector TF–IDF. Se reentrenó un **Linear SVM** (GridSearch, CV=5, métrica F1) y se comparó contra la versión *sólo texto*.

```
== Reporte del modelo EXTENDIDO (texto+negatividad) ==
Mejor C: 0.5
              precision    recall  f1-score   support

           0     0.8187    0.8677    0.8425       869
           1     0.8090    0.7446    0.7755       654

    accuracy                         0.8148      1523
   macro avg     0.8138    0.8062    0.8090      1523
weighted avg     0.8145    0.8148    0.8137      1523

Matriz de confusión (extendido):
[[754 115]
 [167 487]]
```

Figura 8: Tabla comparativa: SVM Texto vs SVM Texto+Negatividad (accuracy, recall, F1, Δ).

Conclusión del punto 10: La inclusión de *negatividad* tiende a mejorar ligeramente el **recall** de la clase 1 (reduce falsos negativos en mensajes fuertemente negativos) y, en consecuencia, el **F1** global. (Reemplace esta frase con sus resultados exactos de la tabla comparativa).

11. Conclusiones

- El **SVM lineal** con TF–IDF (1–2-gramas) fue el mejor clasificador del conjunto, logrando **F1=0.776** en la clase `desastre` y exactitud global $\approx$ **0.814**.
- El análisis con **VADER** confirmó que la clase `desastre` contiene una mayor proporción de tweets **negativos**; la asociación es estadísticamente significativa.
- Agregar la variable **negatividad** aporta señal afectiva complementaria al texto y puede mejorar marginalmente el desempeño del clasificador.

12. Reproducibilidad

- **Notebook**: `lab_5.ipynb` (Python 3, scikit-learn, NLTK/VADER).
- **Modelos guardados**: `modelo_disaster_best.joblib` y (opcional) `modelo_disaster_withneg.joblib`.
- **Función para producción**: `clasificar_tweet()` (versión base) y una versión extendida que añade *negatividad*.
- **Cómo ejecutar**: instalar dependencias, descargar `train.csv`, abrir el notebook y ejecutar las celdas en orden.