Instituto Politécnico Nacional
Escuela Superior de Cómputo

PRÁCTICA 4: FIRST STAGE - CLICKBAIT DETECTION

Natural Language Processing

Integrantes:

- Ortega Prado Mauricio
- Palmerin García Diego
- Pérez Gómez Andres

Grupo: 7CM2

Profesor: Joel Omar Juárez Gambino

21 DE MAYO DE 2025

Tarea por resolver.

El objetivo de la práctica es desarrollar un modelo que identificara si un título de noticia es clickbait o no lo era. Para lograr esto, se utilizó un corpus en español con etiquetas (Clickbait y No), se aplicó un proceso de limpieza, vectorización, entrenamiento y evaluación de los modelos de clasificación supervisada. El modelo con el mejor f1-macro se utilizó para ser entrenado con el corpus de entrenamiento completo y utilizado para predecir las etiquetas de un conjunto de desarrollo.

Métodos de aprendizaje automático seleccionados.

Se seleccionaron y probaron los diversos algoritmos de clasificación, los cuales son los siguientes:

- Naive Bayes (MultinomialNB).
- Regresión Logística (LogisticRegression).
- Máquinas de Vectores de Soporte (SVM).

Cada uno de estos modelos fue probado modificando sus hiperparametros, estos ajustes se describen en la siguiente sección.

Hiperparámetros ajustados.

Para cada modelo de clasificación, se exploraron diferentes combinaciones de hiperparámetros utilizando una búsqueda exhaustiva con validación cruzada (GridSearchCV) y particionado estratificado (StratifiedKFold con n_splits=5). Los siguientes son los hiperparámetros probados para cada clasificador:

- **Multinomial Naive Bayes (MultinomialNB)**
  - alpha: [0.1, 0.5, 1.0]

- **Regresión Logística (LogisticRegression)**
  - C: [0.1, 1.0, 10]
  - solver: ["liblinear"]

- **Máquinas de Vectores de Soporte (SVM - SVC)**
  - C: [0.1, 1.0, 10]
  - kernel: ["linear", "rbf"]

Representación de texto probada

Vectorizadores:

- TfidfVectorizer
- CountVectorizer (frecuencia y binario)

n-gramas utilizados:

- Unigramas (1, 1)
- Bigramas (2, 2)
- Trigramas (3, 3)
- Unigramas + Bigramas (1, 2)
- Unigramas + Trigramas (1, 3)
- Bigramas + Trigramas (2, 3)

Reporte de clasificación de cada experimento.

ML Method	ML Hyperparameters	Text Normalization	Text Representation	Balance Methods	Average F-Score Macro
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Binario unigramas	Oversampling	0.7566
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	TF-IDF uni_bi	Oversampling	0.7562
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	TF-IDF unigramas	Oversampling	0.7490
SVM	'C': 10, 'kernel': 'rbf'	Eliminar URLs y conservar #””	Binario unigramas	Oversampling	0.7469
SVM	'C': 10, 'kernel': 'linear'	Eliminar URLs y conservar #””	TF-IDF unigramas + bigramas	Sin Balanceo	0.7465
SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	TF-IDF uni_bi	Oversampling	0.7459
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Binario unigramas + bigramas	Sin Balanceo	0.7455
SVM	'C': 10, 'kernel': 'rbf'	Eliminar URLs y conservar #””	Binario uni_bi	Oversampling	0.7439
SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	Binario unigramas + bigramas	Sin Balanceo	0.7412
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Binario unigramas + trigramas	Oversampling	0.7407
SVM	'C': 10, 'kernel': 'rbf'	Eliminar URLs y conservar #””	Binario unigramas + trigramas	Oversampling	0.7401
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Binario uni_bi	Oversampling	0.7401
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Binario unigramas	Sin Balanceo	0.7387

LogisticRegression	'C': 1.0, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Frecuencia unigramas	Oversampling	0.7367
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Frecuencia unigramas + bigramas	Sin Balanceo	0.7360
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	TF-IDF unigramas + trigramas	Oversampling	0.7344
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Frecuencia unigramas + bigramas	Oversampling	0.7344
SVM	'C': 10, 'kernel': 'linear'	Eliminar URLs y conservar #””	TF-IDF unigramas + trigramas	Oversampling	0.7335
SVM	'C': 10, 'kernel': 'linear'	Eliminar URLs y conservar #””	TF-IDF unigramas + trigramas	Sin Balanceo	0.7335
SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	Frecuencia unigramas	Sin Balanceo	0.7327
SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	Binario unigramas + trigramas	Sin Balanceo	0.7300
SVM	'C': 10, 'kernel': 'linear'	Eliminar URLs y conservar #””	TF-IDF unigramas	Sin Balanceo	0.7292
LogisticRegression	'C': 1.0, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Frecuencia unigramas	Sin Balanceo	0.7286
SVM	'C': 10, 'kernel': 'rbf'	Eliminar URLs y conservar #””	Frecuencia unigramas + trigramas	Oversampling	0.7257
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	TF-IDF unigramas	Sin Balanceo	0.7247
SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	Binario unigramas	Sin Balanceo	0.7243
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Frecuencia unigramas + trigramas	Oversampling	0.7227

SVM	'C': 10, 'kernel': 'rbf'	Eliminar URLs y conservar #””	Frecuencia unigramas + bigramas	Oversampling	0.7222
SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	Frecuencia unigramas + bigramas	Sin Balanceo	0.7189
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Binario unigramas + trigramas	Sin Balanceo	0.7115
SVM	'C': 0.1, 'kernel': 'linear'	Eliminar URLs y conservar #””	Frecuencia unigramas + trigramas	Sin Balanceo	0.7086
SVM	'C': 10, 'kernel': 'rbf'	Eliminar URLs y conservar #””	Frecuencia unigramas	Oversampling	0.7056
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Frecuencia unigramas + trigramas	Sin Balanceo	0.7012
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	TF-IDF uni_bi	Oversampling	0.6994
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	TF-IDF unigramas + bigramas	Sin Balanceo	0.6985
SVM	'C': 1.0, 'kernel': 'rbf'	Eliminar URLs y conservar #””	TF-IDF unigramas	Oversampling	0.6956
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	TF-IDF unigramas + trigramas	Oversampling	0.6925
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	Binario unigramas + trigramas	Oversampling	0.6849
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	Binario uni_bi	Oversampling	0.6824
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	Binario unigramas	Oversampling	0.6716
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	Frecuencia unigramas + trigramas	Oversampling	0.6710

LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	TF-IDF unigramas + trigramas	Sin Balanceo	0.6710
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	Frecuencia unigramas	Oversampling	0.6704
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	Frecuencia unigramas + bigramas	Oversampling	0.6698
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	TF-IDF unigramas	Oversampling	0.6672
NaiveBayes	'alpha': 1.0	Eliminar URLs y conservar #””	Binario unigramas	Sin Balanceo	0.6645
SVM	'C': 10, 'kernel': 'linear'	Eliminar URLs y conservar #””	TF-IDF bigramas	Sin Balanceo	0.6600
SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	TF-IDF bigramas	Oversampling	0.6580
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	TF-IDF bigramas	Oversampling	0.6574
NaiveBayes	'alpha': 1.0	Eliminar URLs y conservar #””	Frecuencia unigramas	Sin Balanceo	0.6564
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	Binario bigramas	Oversampling	0.6479
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	Frecuencia bigramas	Oversampling	0.6421
NaiveBayes	'alpha': 1.0	Eliminar URLs y conservar #””	Binario bigramas	Sin Balanceo	0.6419
LogisticRegression	'C': 1.0, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Binario bigramas	Oversampling	0.6402
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	TF-IDF bigramas	Oversampling	0.6402
NaiveBayes	'alpha': 1.0	Eliminar URLs y conservar #””	Binario unigramas + bigramas	Sin Balanceo	0.6395

NaiveBayes	'alpha': 1.0	Eliminar URLs y conservar #””	Frecuencia unigramas + bigramas	Sin Balanceo	0.6312
NaiveBayes	'alpha': 1.0	Eliminar URLs y conservar #””	Frecuencia bigramas	Sin Balanceo	0.6309
LogisticRegression	'C': 1.0, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Frecuencia bigramas	Oversampling	0.6287
NaiveBayes	'alpha': 1.0	Eliminar URLs y conservar #””	Binario unigramas + trigramas	Sin Balanceo	0.6275
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	TF-IDF unigramas	Sin Balanceo	0.6242
SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	Binario bigramas	Sin Balanceo	0.6237
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	TF-IDF unigramas + bigramas	Sin Balanceo	0.6169
SVM	'C': 10, 'kernel': 'rbf'	Eliminar URLs y conservar #””	Binario bigramas	Oversampling	0.6139
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	TF-IDF bigramas	Sin Balanceo	0.6127
NaiveBayes	'alpha': 1.0	Eliminar URLs y conservar #””	Frecuencia unigramas + trigramas	Sin Balanceo	0.6103
SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	Frecuencia bigramas	Sin Balanceo	0.6077
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Binario bigramas	Sin Balanceo	0.6020
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	TF-IDF unigramas + trigramas	Sin Balanceo	0.5962
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Frecuencia bigramas	Sin Balanceo	0.5947
SVM	'C': 10, 'kernel': 'rbf'	Eliminar URLs y conservar #””	Frecuencia bigramas	Oversampling	0.5898

NaiveBayes	'alpha': 1.0	Eliminar URLs y conservar #””	Binario trigramas	Sin Balanceo	0.5887
NaiveBayes	'alpha': 1.0	Eliminar URLs y conservar #””	Frecuencia trigramas	Sin Balanceo	0.5885
SVM	'C': 10, 'kernel': 'linear'	Eliminar URLs y conservar #””	TF-IDF trigramas	Oversampling	0.5680
SVM	'C': 10, 'kernel': 'linear'	Eliminar URLs y conservar #””	TF-IDF trigramas	Sin Balanceo	0.5680
LogisticRegression	'C': 1.0, 'solver': 'liblinear'	Eliminar URLs y conservar #””	TF-IDF trigramas	Oversampling	0.5649
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	TF-IDF trigramas	Oversampling	0.5392
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	Frecuencia trigramas	Oversampling	0.5334
NaiveBayes	'alpha': 0.1	Eliminar URLs y conservar #””	Binario trigramas	Oversampling	0.5262
NaiveBayes	'alpha': 0.5	Eliminar URLs y conservar #””	TF-IDF bigramas	Sin Balanceo	0.4968
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	TF-IDF trigramas	Sin Balanceo	0.4885
LogisticRegression	'C': 1.0, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Binario trigramas	Oversampling	0.4815
LogisticRegression	'C': 1.0, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Frecuencia trigramas	Oversampling	0.4767
SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	Binario trigramas	Oversampling	0.4727
SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	Binario trigramas	Sin Balanceo	0.4727
SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	Frecuencia trigramas	Sin Balanceo	0.4727

SVM	'C': 1.0, 'kernel': 'linear'	Eliminar URLs y conservar #””	Frecuencia trigramas	Oversampling	0.4727
NaiveBayes	'alpha': 0.5	Eliminar URLs y conservar #””	TF-IDF trigramas	Sin Balanceo	0.4649
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Frecuencia trigramas	Sin Balanceo	0.4529
LogisticRegression	'C': 10, 'solver': 'liblinear'	Eliminar URLs y conservar #””	Binario trigramas	Sin Balanceo	0.4428

Conclusión del modelo seleccionado

Para la detección de titulares tipo clickbait, se optó por utilizar el modelo de LogisticRegression debido a su excelente desempeño observado durante los experimentos con validación cruzada. Este modelo es particularmente eficaz para problemas de clasificación binaria, como el presente caso, ya que busca una separación lineal óptima entre las clases (Clickbait y No), maximizando la precisión y generalización.

Además, se utilizó una representación binaria de texto con unigramas (CountVectorizer(binary=True, ngram_range=(1,1)), lo que permite capturar la presencia o ausencia de palabras clave individuales que suelen ser indicativas del estilo clickbait, sin verse afectado por la frecuencia con la que aparecen. Esta representación es sencilla pero efectiva, especialmente cuando los títulos son cortos y contienen términos distintivos.

La combinación de un modelo lineal robusto y una representación simple pero expresiva del texto demostró ser una de las configuraciones con mejor balance entre rendimiento y eficiencia, obteniendo un alto puntaje en la métrica f1_macro sin requerir procesamiento o complejidad adicional.