



**REPORTE 2 - PROYECTO FINAL
CATEGORIZACIÓN Y ANÁLISIS DE SENTIMIENTOS
DE ARTÍCULOS DE NOTICIAS**

GRUPO 16

Andrés Felipe Gualdrón Gutiérrez
Juan Manzano Barona
Jersson Hernán Morales Hernández
Carina Lizebeth Ordoñez Araque

1 INTRODUCCIÓN

Este proyecto propone un enfoque innovador que integra técnicas de aprendizaje automático supervisado y no supervisado para la categorización y re-categorización de noticias, complementado por el uso del análisis de sentimientos mediante VADER (Valence Aware Dictionary and sEntiment Reasoning). Esta metodología se presenta como una herramienta robusta y adaptable para navegar el complejo panorama informativo contemporáneo.

El sistema resultante ofrecerá a los usuarios una visión integral y matizada del entorno de noticias en constante evolución, facilitando una comprensión más profunda de las diversas categorías de artículos. Al utilizar un amplio conjunto de datos que abarca 210,000 artículos y sus respectivas categorías, junto con el análisis de sentimientos proporcionado por VADER, este proyecto busca ofrecer insights más precisos y detallados.

Con el fin de mitigar el impacto de la polarización en la opinión pública, esta propuesta permitirá a las partes interesadas tomar decisiones más informadas y promover una difusión de noticias más equilibrada y consciente. La combinación de métodos de aprendizaje automático y análisis de sentimientos no solo optimiza el filtrado de información relevante, sino que también contribuye a un consumo informativo más equilibrado y reflexivo, esencial en un contexto mediático tan dinámico y desafiante.

2 PROBLEMA Y SU CONTEXTO

El acceso a la información por parte de la población ha tenido un gran cambio durante las últimas décadas, en las que se han ido desarrollando mecanismos que facilitan el acceso a los datos, noticias e información general de una forma mucho más directa y rápida. Las nuevas tecnologías, la creación y desarrollo de dispositivos móviles y el gran avance en la conectividad han propiciado un entorno de creación y difusión de todo tipo de contenidos de una forma inmediata y global, lo que supone un reto importante en cuanto a la filtración de la información a la que se accede.

Una de las consecuencias de este modelo, que se ha reconocido como una problemática social en muchos países, está relacionada con la polarización de la opinión pública. La proliferación de creadores y difusores de contenidos ha contribuido a alimentar esta polarización mucho más allá de la política o la religión, llegando a prácticamente cualquier aspecto social, cultural o deportivo.

En este contexto, cobra una especial relevancia el filtrado de la información a la que accedemos y, para ello, la validación y el refinamiento de la categorización de los artículos de noticias y opinión se convierten en un aspecto fundamental.

3 PREGUNTA DE NEGOCIO Y ALCANCE DEL PROYECTO

Este proyecto tiene como objetivo el desarrollo de un sistema automatizado que permita la validación, el refinamiento y una posible re-categorización de artículos de noticias, utilizando como fuente una base de datos del portal HuffPost que incluye, aproximadamente, 210.000 artículos.

Este sistema se llevará a cabo mediante la realización de un análisis de sentimientos o minería de opiniones que permite identificar el tono emocional presente en un texto. Para ello, se utilizará un enfoque híbrido de aprendizaje supervisado y no supervisado para realizar la categorización de la información, además del algoritmo VADER (Valence Aware Dictionary for Sentiment Reasoning) o BERT-base-uncased (Bidirectional Encoder Representations from Transformers) para el análisis de sentimientos.

3.1 Metodología (Revisada)

- Categorización y Re-categorización de Noticias
 - Análisis No Supervisado:

- Implementar la técnica Latent Dirichlet Allocation (LDA) para el modelado de temas.
 - Descubrir patrones temáticos latentes en el conjunto de datos.
- Análisis Supervisado:
 - Utilizar las etiquetas de categoría existentes para entrenar un modelo de clasificación supervisada (SVM, Random Forest, o XGBoost).
 - Evaluar la precisión del modelo en comparación con las categorías existentes.
- Análisis de Sentimientos
 - Implementar VADER o BERT para el análisis de sentimientos.
 - Aplicar VADER/BERT a cada artículo para obtener puntuaciones de sentimiento (positivo, negativo, neutral).
 - Agregar los resultados de sentimiento por categoría y a lo largo del tiempo.
- Desarrollo de un Panel de Control
 - Crear un panel interactivo para visualizar:
 - Distribución y evolución de categorías y subcategorías identificadas
 - Tendencias de sentimiento para diferentes categorías de noticias utilizando los resultados de VADER y/o de BERT.
 - Artículos clave para temas específicos
 - Análisis de frecuencia de palabras por categoría y subcategoría
 - Indicadores de polarización basados en la distribución de sentimientos de VADER

3.2 Aplicaciones Potenciales

- Asistir a los medios de comunicación en la comprensión de temas tendencia y el sentimiento público, en general, con una categorización más precisa.
- Ayudar a los legisladores a medir el sentimiento público sobre diversos temas y a desarrollar políticas para mitigar la polarización, respaldados por análisis de sentimientos adaptados al contexto noticioso.
- Ayudar a las empresas a monitorear el sentimiento de las noticias relacionadas con su industria a través de fuentes de noticias, con una comprensión de las categorías y subcategorías relevantes.

4 CONJUNTO DE DATOS

Los datos corresponden a artículos de noticias del portal HuffPost en Estados Unidos que fueron recopilados entre 2012 y 2022. El dataset cuenta con cerca de 210 mil titulares de noticias (HEADLINE) junto con:

- CATEGORY: Categoría en la que el artículo fue publicado
- SHORT_DESCRIPTION: Resumen del artículo
- AUTHORS: Lista de los autores quienes contribuyeron en el artículo
- DATE: Fecha de publicación del artículo
- LINK: Hipervínculo del artículo original de la noticia

Los datos están contenidos en un archivo tipo *.JSON, de 87.3MB, disponible para acceso público permanentemente en la plataforma Kaggle con las siguientes citas:

- <https://www.kaggle.com/datasets/rmisra/news-category-dataset>
- Misra, Rishabh. "News Category Dataset." arXiv preprint arXiv: 2209.11429 (2022).
- Misra, Rishabh and Jigyasa Grover. "Sculpting Data for ML: The first act of Machine Learning." ISBN 9798585463570 (2021).

5 MODELOS DESARROLLADOS

5.1 Pre-procesamiento y Procesamiento de los Datos.

Para continuar con la etapa de pruebas para los modelos en desarrollo se agruparon las categorías existentes de acuerdo con el criterio del grupo considerando que algunos de categorías eran redundantes y otras tenían relación entre sí permitiendo. De esta manera, el número de categorías se redujo significativamente de 42 a 27, como parte del pre-procesamiento de los datos.

```
# Se agrupan las categorías similares.
category_map = {
    'PARENTING': 'PARENTS',
    'HEALTHY LIVING': 'WELLNESS',
    'THE WORLDPOST': 'WORLD NEWS',
    'WORLDPOST': 'WORLD NEWS',
    'U.S. NEWS': 'WORLD NEWS',
    'ARTS': 'CULTURE & ARTS',
    'TASTE': 'FOOD & DRINK',
    'COLLEGE': 'EDUCATION',
    'MONEY': 'BUSINESS',
    'STYLE': 'STYLE & BEAUTY',
    'GREEN' : 'ENVIRONMENT',
    'BLACK VOICES' : 'DIVERSITY VOICES',
    'LATINO VOICES': 'DIVERSITY VOICES',
    'QUEER VOICES': 'DIVERSITY VOICES',
    'WEIRD NEWS': 'VARIETY',
    'GOOD NEWS': 'VARIETY',
    'FIFTY': 'VARIETY',
}
```

Siguiendo con el pre-procesamiento de los datos, se ejecutó una limpieza de caracteres, signos de puntuación (solo para el modelo de categorización; para el análisis de sentimientos se mantuvieron los signos de puntuación en el texto), extensión de contracciones, tokenización, detección y rotulación del tipo de palabra (las negativas se mantuvieron para mejorar tanto la categorización como el análisis de sentimientos), y lematización.

Todo se condensó en un script de *Python* para ser usado de manera flexible en un notebook de *Jupyter*, y guardado en un archivo *.csv.

- *news_processor.py*
- *pre-processing&engg-feature.ipynb*

Para la fase de experimentos se usó una muestra equivalente al 1% del total de los datos pre-procesados para acelerar el entrenamiento y prueba de los modelos.

Al dataset pre-procesado y guardado en archivo *.csv se le determinó la cantidad de palabras (mediante *CountVectorizer*) y su frecuencia inversa (mediante *TfidfVectorizer*) para obtener un matriz numérica que junto con los resultados del modelo No Supervisado, es el set de datos para entrenar el modelo Supervisado. La preparación de los datos para su uso en los modelos se condensó en un script de *Python* para ser usado de manera flexible para el entrenamiento y prueba de los modelos en un notebook de *Jupyter*.

- *topic_modeling_prep.py*

5.2 No Supervisado (Latent Dirichlet Allocation - LDA)

Se usó LDA como un modelo No Supervisado para detectar temas (topics) que pudieran agregar información particular a los artículos para mejorar el desempeño del modelo supervisado. Se determinaron los parámetros: *n_topics* y *learning_offset* como variables para los experimentos.

Los experimentos con el modelo no supervisado LDA se ejecutaron junto con el modelo supervisado.

5.3 Supervisado (Random Forest - RF)

Un modelo de clasificación de ensamble Random Forest fue tomado como primera opción para el desarrollo del modelo supervisado. Los parámetros que se determinaron para los experimentos fueron: *n_estimators*, *criterion*, *max_samples*, y *class_balance*.

Para una primera aproximación, se creó un script en Python para correrlo en un notebook de *Jupyter*, e identificar el desempeño preliminar del modelo. A partir de este, se elaboró un script en Python para ser usado en un ambiente de experimentos MLFlow en una instancia EC2 de AWS.

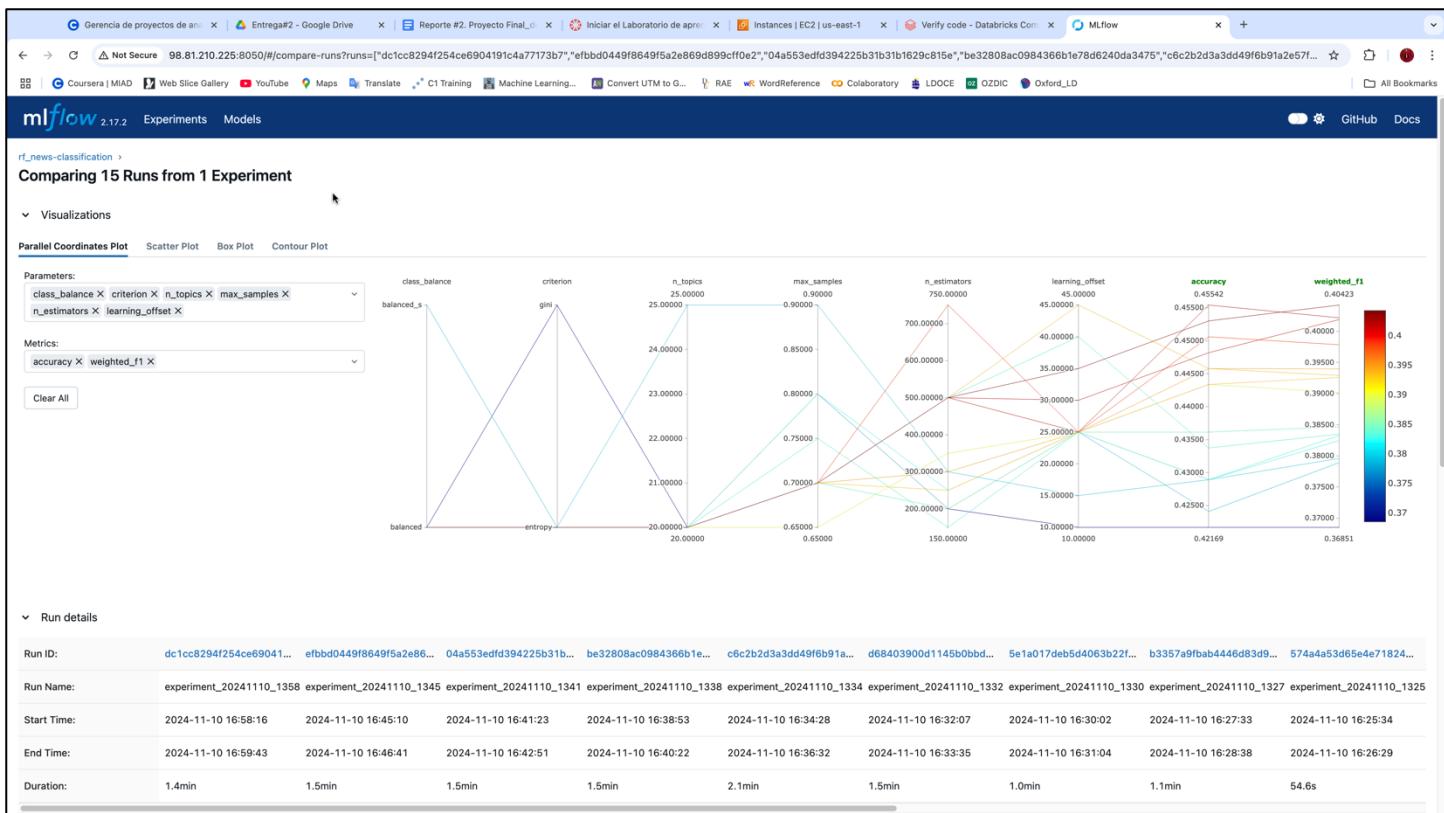
- *rf_news_classifier.py*
- *rf_topic&category_model.ipynb*
- *mlflow_rf_news_classification.py*

Por restricciones de la cuenta de carácter académico, no se pudo lanzar una instancia con suficiente capacidad para correr los experimentos ágilmente. Para el caso del modelo RF con una base de datos de entrenamiento de 2,074 artículos de noticias, el tiempo promedio fue de 1,05 minutos por experimento.

Los valores de los parámetros fueron cambiados para cada experimento desde la conexión local a una instancia EC2 (**news categorization instance**) del AWS, usando el comando *nano*. (refiérase al Anexo -1).

Se ejecutaron un total de 15 experimentos que determinaron que, para el LDA, 20 temas y un learning_offset de 35, y para el RF, una cantidad de árboles igual a 500, una muestra del 70% de los datos por árbol, un ajuste en las clases -balanced-, y un criterio de división -entropy-, resulta en un modelo con el mayor *Weighted_F1* (0.404), y un Accuracy (0.453) muy alto. Los soportes se muestran a continuación y sus estadísticas en el Anexo -2:

Run Name	Created	Dataset	Duration	Source	Models	accuracy	weighted_f1
experiment_20241110...	1 minute ago	-	1.4min	mlflow...	sklearn	0.4481927...	0.4018882...
experiment_20241110...	14 minutes ago	-	1.5min	mlflow...	sklearn	0.4337349...	0.3834896...
experiment_20241110...	18 minutes ago	-	1.5min	mlflow...	sklearn	0.4457831...	0.3929193...
experiment_20241110...	21 minutes ago	-	1.5min	mlflow...	sklearn	0.4530120...	0.4042251...
experiment_20241110...	25 minutes ago	-	2.1min	mlflow...	sklearn	0.4506024...	0.3978516...
experiment_20241110...	27 minutes ago	-	1.5min	mlflow...	sklearn	0.4554216...	0.4021729...
experiment_20241110...	30 minutes ago	-	1.0min	mlflow...	sklearn	0.4457831...	0.3939579...
experiment_20241110...	32 minutes ago	-	1.1min	mlflow...	sklearn	0.4433734...	0.3901855...
experiment_20241110...	34 minutes ago	-	54.6s	mlflow...	sklearn	0.4433734...	0.3925928...
experiment_20241110...	36 minutes ago	-	56.3s	mlflow...	sklearn	0.4289156...	0.3824572...
experiment_20241110...	38 minutes ago	-	40.8s	mlflow...	sklearn	0.4289156...	0.3832709...
experiment_20241110...	44 minutes ago	-	1.1min	mlflow...	sklearn	0.4289156...	0.3795798...
experiment_20241110...	46 minutes ago	-	46.2s	mlflow...	sklearn	0.4361445...	0.3846469...
experiment_20241110...	56 minutes ago	-	47.9s	mlflow...	sklearn	0.4216867...	0.3685115...
experiment_20241110...	1 hour ago	-	52.4s	mlflow...	sklearn	0.4240963...	0.3789067...



5.4 Supervisado (Support Vector Machine - SVM)

Un modelo de clasificación SVM fue tomado como segunda opción para el desarrollo del modelo supervisado. Los parámetros que se determinaron para los experimentos fueron: *C (Regularization parameter)*, *gamma*, *kernel*, y *decision_function_shape*.

Se creó un script en Python para correrlo en un notebook de *Jupyter*, e identificar el desempeño preliminar del modelo SVM. A partir de este, se elaboró un script en Python para ser usado en un ambiente de experimentos MLFlow en una instancia EC2 de AWS.

- *svm_news_classifier.py*
- *svm_topic&category_model.ipynb*
- *mlflow_svm_news_classification.py*

Por restricciones de la cuenta de carácter académico, no se pudo lanzar una instancia con suficiente capacidad para correr los experimentos ágilmente. Para el caso del modelo SVM con una base de datos de entrenamiento de 2,074 artículos de noticias, el tiempo promedio fue de 24.3 minutos por experimento, haciendo esta opción inviable considerando que la base de datos completa es 100 veces más grande.

Los valores de los parámetros fueron cambiados para cada experimento desde la conexión local a una instancia EC2 (**news_categorization_instance**) del AWS, usando el comando *nano* (refiérase al Anexo - 1).

Se ejecutaron un total de 10 experimentos que determinaron que para el LDA, 20 temas y un learning_offset de 35, y para el SVM, un factor de regularización igual a 10, un kernel -rbf- (radial basis function), una ajuste en las clases -balanced-, un gamma -scale-, y una función de decisión -ovr-, resultan en un modelo con el mayor Accuracy (0.397) , y *Weighted_F1* (0.352) muy alto. Los soportes se muestran a continuación y sus estadísticas en el Anexo -3:

Screenshot of the mlflow interface showing the 'Experiments' page for the 'svm_news-classification' experiment. The page displays a table of 10 runs, each with details like Run Name, Created, Duration, Source, Models, accuracy, and weighted_f1. A search bar at the top filters runs by metrics.rmse < 1 and params.model = "tree".

	Run Name	Created	Dataset	Duration	Source	Models	accuracy	weighted_f1
experiment_20241110...	46 minutes ago	-	Dataset	24.2min	mlflow...	sklearn	0.387951...	0.342874...
experiment_20241110...	1 hour ago	-	Dataset	24.2min	mlflow...	sklearn	0.3951807...	0.3516806...
experiment_20241110...	1 hour ago	-	Dataset	24.1min	mlflow...	sklearn	0.2096385...	0.1339318...
experiment_20241110...	2 hours ago	-	Dataset	24.3min	mlflow...	sklearn	0.3951807...	0.3516806...
experiment_20241110...	2 hours ago	-	Dataset	24.2min	mlflow...	sklearn	0.3951807...	0.3516806...
experiment_20241110...	3 hours ago	-	Dataset	24.3min	mlflow...	sklearn	0.3951807...	0.3514148...
experiment_20241110...	3 hours ago	-	Dataset	24.2min	mlflow...	sklearn	0.3975903...	0.3536319...
experiment_20241110...	4 hours ago	-	Dataset	23.8min	mlflow...	sklearn	0.392771...	0.346444...
experiment_20241110...	4 hours ago	-	Dataset	25.7min	mlflow...	sklearn	0.2481927...	0.1192014...
experiment_20241110...	5 hours ago	-	Dataset	23.9min	mlflow...	sklearn	0.3975903...	0.3503378...

10 matching runs

Screenshot of the mlflow interface showing the 'Comparing 10 Runs from 1 Experiment' page for the 'svm_news-classification' experiment. It features a parallel coordinates plot comparing parameters like kernel, decision_function_shape, gamma, C, n_topics, learning_offset, accuracy, and weighted_f1 across 10 runs. The plot shows a complex relationship between these variables, with accuracy and weighted_f1 generally increasing as other parameters like C and n_topics increase.

Parallel Coordinates Plot

Parameters: kernel X decision_function_shape X gamma X C X n_topics X learning_offset X

Metrics: accuracy X weighted_f1 X

Run details:

Run ID:	142123e2ae5443238d6c9c3377f10349...	d6106a69384d463a35...	98c417de5c87442ca67a...	f7e3ae0c2bca4809b7f79...	b4cc070622f64b639f7d...	933667465e3e45d9afdf...	ec4604ac44a24198afcc...	e7cac180344044388f55...	c59cf39aea94baabc3e...
Run Name:	experiment_20241110_1845	experiment_20241110_1819	experiment_20241110_1753	experiment_20241110_1714	experiment_20241110_1647	experiment_20241110_1621	experiment_20241110_1550	experiment_20241110_1502	experiment_20241110_1434
Start Time:	2024-11-10 21:45:26	2024-11-10 21:19:09	2024-11-10 20:53:16	2024-11-10 20:14:45	2024-11-10 19:47:18	2024-11-10 19:21:14	2024-11-10 18:50:12	2024-11-10 18:02:26	2024-11-10 17:34:48
End Time:	2024-11-10 22:09:39	2024-11-10 21:43:21	2024-11-10 21:17:24	2024-11-10 20:39:00	2024-11-10 20:11:28	2024-11-10 19:45:33	2024-11-10 19:14:26	2024-11-10 18:26:16	2024-11-10 18:00:32
Duration:	24.2min	24.2min	24.1min	24.3min	24.2min	24.3min	24.2min	23.8min	25.7min

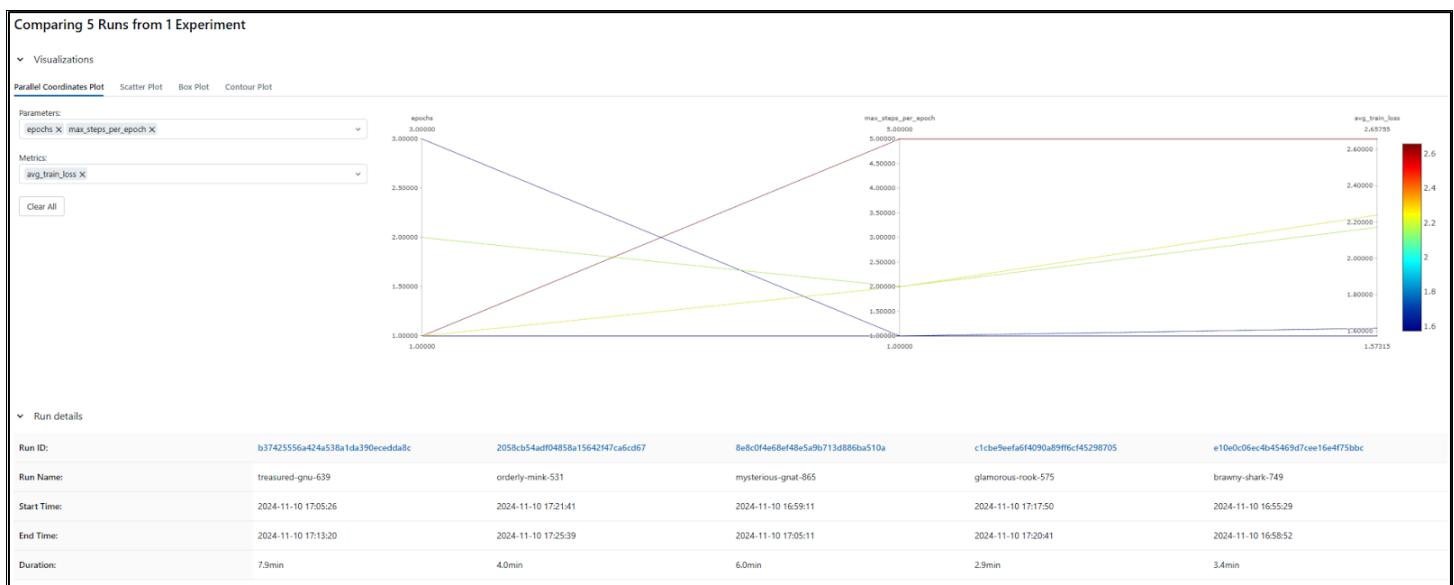
En este caso, el modelo LDA-SVM no tiene ventaja alguna sobre el modelo LDA-RF, por lo que se descarta. El modelo LDA-RF se seguirá desarrollando.

5.5 No Supervisado – BERT

Para el análisis de sentimientos, se exploró *BERT-base-uncased* de Hugging Face como modelo pre-entrenado para asignar un sentimiento a cada uno de los artículos de noticias con el propósito de tener una comparación con el algoritmo VADER que se usó durante la fase de exploración de los datos.

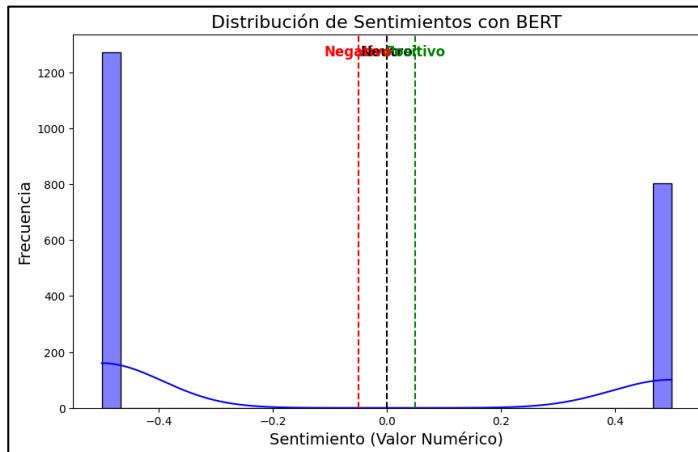
Inicialmente, se creó un script en Python para llevar a cabo una evaluación preliminar en un entorno de Jupyter Notebook. Posteriormente, este script fue adaptado para realizar experimentos en MLflow, ejecutado en la plataforma de *Databricks* para facilitar la gestión de experimentos y el análisis de resultados. Los parámetros definidos para los experimentos incluyeron: *epochs*, *max_steps_per_epoch*, y *sample_fraction*, mientras que las pruebas se llevaron a cabo con un 10% del dataset para reducir el tiempo de procesamiento.

Un total de 5 experimentos fueron realizados variando los valores de *epochs* y *max_steps_per_epoch* para un total de 20.704 artículos. Los resultados de las métricas de precisión (*avg_val_accuracy*) y pérdida promedio de entrenamiento (*avg_train_loss*) indican que a mayor número de *epochs* y *max_steps_per_epoch*, el modelo logra una precisión de validación ligeramente superior, aunque esto aumenta el tiempo de ejecución considerablemente.

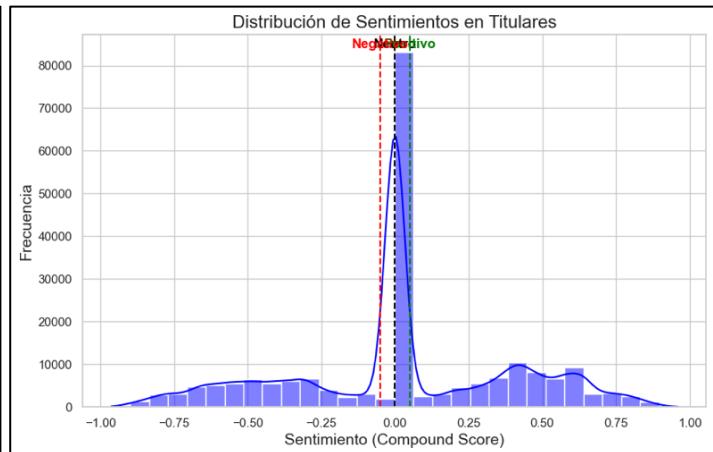


5.6 Comparación de Modelos de Análisis de Sentimientos: BERT vs. VADER

BERT



VADER



La distribución de los sentimientos asignados por BERT muestra una tendencia clara a clasificar los textos en categorías más extremas de "positivo" o "negativo". Esto se puede observar en la gráfica de BERT, donde la mayoría de los puntajes se concentran en los extremos, con pocos valores cercanos a cero (neutro). Esto indica que BERT, al ser un modelo pre entrenado específicamente para análisis de sentimiento, tiende a identificar y clasificar los sentimientos de manera más categórica y directa, dejando poco espacio para la ambigüedad o los valores neutros.

Por el contrario, VADER muestra una distribución de puntajes más continua y con una mayor concentración de valores alrededor de cero. Esto significa que VADER clasifica muchos textos como "neutros" y ofrece una gama más matizada de puntajes. La gráfica de VADER muestra claramente este comportamiento, con un pico pronunciado en el centro y una disminución gradual hacia los extremos positivo y negativo. Esto se debe a que VADER está diseñado para capturar una gama más amplia de matices en el sentimiento, en lugar de hacer una categorización binaria.

6 REPOSITORIOS

6.1 DVC

Los datos procesados fueron almacenados en el servicio AWS-S3 para su acceso por los integrantes del grupo y el registro fue guardado en el repositorio.

```
jerssonm@Jerssons-MacBook-Pro ~ % source venv/bin/activate
[venv]: no such file or directory: venv/bin/activate
jerssonm@Jerssons-MacBook-Pro ~ % cd Documents/GitHub/miad2024_dsa_G16
jerssonm@Jerssons-MacBook-Pro ~/Documents/GitHub/miad2024_dsa_G16 % source venv/bin/activate
(venv) jerssonm@Jerssons-MacBook-Pro ~/Documents/GitHub/miad2024_dsa_G16 % dvc add processed/processed_news_dataset.csv
ERROR: stage working dir '/Users/jerssonm/Documents/GitHub/miad2024_dsa_G16/processed' does not exist
(venv) jerssonm@Jerssons-MacBook-Pro ~/Documents/GitHub/miad2024_dsa_G16 % cd data
(venv) jerssonm@Jerssons-MacBook-Pro ~/Documents/GitHub/miad2024_dsa_G16 % dvc add processed/processed_news_dataset.csv
100% Adding... [1/1 [00:00, 7.01file/s]

To track the changes with git, run:
  git add processed/.gitignore processed/processed_news_dataset.csv.dvc

To enable auto staging, run:
  dvc config core.autostage true

(venv) jerssonm@Jerssons-MacBook-Pro ~/Documents/GitHub/miad2024_dsa_G16 % dvc add processed/processed_news_dataset.cs_sample.csv
Adding...
ERROR: output 'processed/processed_news_dataset.cs_sample.csv' does not exist: [Errno 2] No such file or directory: '/Users/jerssonm/Documents/GitHub/miad2024_dsa_G16/data/processed/processed_news_dataset.cs_sample.csv'
(venv) jerssonm@Jerssons-MacBook-Pro ~/Documents/GitHub/miad2024_dsa_G16 % dvc add processed/processed_news_dataset_sample.csv
100% Adding... [1/1 [00:00, 89.02file/s]

To track the changes with git, run:
  git add processed/.gitignore processed/processed_news_dataset_sample.csv.dvc

To enable auto staging, run:
  dvc config core.autostage true

[[venv]] jerssonm@Jerssons-MacBook-Pro ~/Documents/GitHub/miad2024_dsa_G16 % cd data
[[venv]] jerssonm@Jerssons-MacBook-Pro ~/data % git add processed/.gitignore processed/processed_news_dataset_sample.csv.dvc
[[venv]] jerssonm@Jerssons-MacBook-Pro ~/data % git add processed/processed_news_dataset.csv.dvc
[[venv]] jerssonm@Jerssons-MacBook-Pro ~/data % git commit -m "Adding processed data files to processed dir."
[main bba6659] Adding processed data files to processed dir.
 2 files changed, 10 insertions(+)
 create mode 100644 data/processed/processed_news_dataset.csv.dvc
 create mode 100644 data/processed/processed_news_dataset_sample.csv.dvc
[[venv]] jerssonm@Jerssons-MacBook-Pro ~/data % git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 767 bytes | 767.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/jhermoher/miad2024_dsa_G16.git
 6a315b9..bba6659 main -> main
```

6.2 GIT. https://github.com/jhermoher/miad2024_dsa_G16.git

El repositorio del proyecto fue actualizado con los scripts, notebooks y registro de datos almacenados en el servicio S3. Un -screenshot- de la estructura del repositorio se puede observar en el Anexo -4.

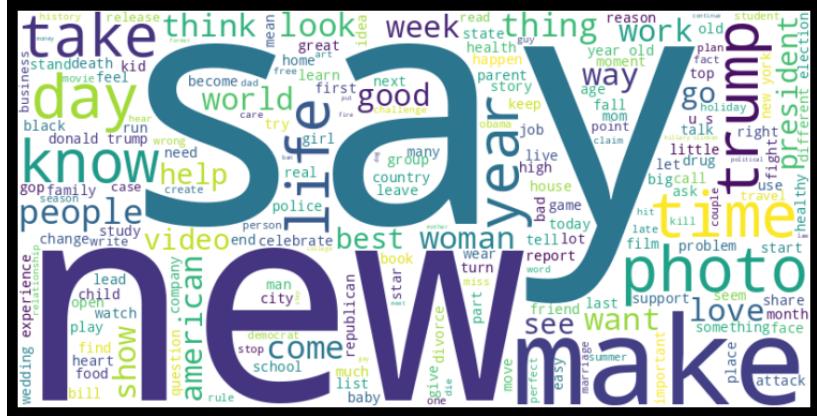
7 DASHBOARD

Este proyecto implementa un tablero interactivo de manera que se facilite la visualización de la predicción de la categoría del artículo de noticias y el resultado del análisis de sentimientos.

Para la implementación de este, utilizamos las librerías *Dash* para construir el tablero, *Plotly Express* para generar gráficos interactivos, *Pandas* para la manipulación y análisis de datos, y *WordCloud* junto a *Matplotlib* para crear visualizaciones de nubes de palabras a partir de los textos analizados.

Se logró la construcción de algunas visualizaciones para ser usadas una vez el modelo final este disponible y la estructura final de tablero esté completamente definida.





8 REPORTE TRABAJO EN EQUIPO

El primer reporte relacionado con el proyecto final del curso busca documentar el avance alcanzado en las semanas 4 y 5.

Semana 4: Durante esta semana, nuestro equipo se enfocó en el desarrollo inicial de los modelos para el proyecto. Para ello, desarrollamos los scripts para el pre-procesamiento de los datos y para su acondicionamiento para usarlos posteriormente, en los modelos. Desarrollamos las primeras versiones de los modelos para correrlos en un IDE local y así, tener una idea de su desempeño con respecto a los datos procesados. Y paralelamente, trabajamos en el desarrollo del tablero siguiendo las especificaciones de la maqueta diseñada previamente.

Semana 5: En esta semana, el equipo se concentró en la evolución y refinamiento de los modelos. Desarrollamos nuevas versiones, y utilizamos MLflow para su versionamiento y el análisis de los resultados experimentales. Realizamos análisis comparativos y seleccionamos las mejores alternativas basadas en los resultados. Además, avanzamos en el desarrollo del tablero según la maqueta establecida, asegurándonos de documentar el proceso.

Para avanzar significativamente en las diferentes actividades de las semanas 4 y 5, las tareas fueron distribuidas a cada integrante de la siguiente manera:

- Andrés Gualdrón. Experimentos modelo no supervisado basado en BERT (sentimientos).
- Juan Manzano. Generación del reporte correspondiente a la segunda entrega.
- Jersson Morales. Experimentos modelos supervisados basados en RF & SVM (categorías).
- Lizebeth Ordoñez. Creación del tablero.

Las mismas se realizaron completamente con responsabilidad y calidad, como se muestran en este reporte.

ANEXO -1. EXPERIMENTACIÓN DESDE LA TERMINAL

```
-/Desktop -- ubuntu@ip-172-31-26-71: ~ ssh -i dsa_llave2.pem ubuntu@54.236.209.242 ... -/Documents/Github/miad2024_dsa_G16/data -- zsh -/Desktop -- ubuntu@ip-172-31-26-71: ~ ssh -i dsa_llave2.pem ubuntu@54.236.209.242 ... +
```

Longitud promedio del texto: 105.6 caracteres
Cantidad promedio de palabras: 15.4 palabras

Top 5 de categorías por frecuencia:

```
category
POLITICS 338
WELLNESS 258
ENTERTAINMENT 171
PARENTS 123
DIVERSITY VOICES 123
```

Name: count, dtype: int64
Fitting LDA model...
Training SVM classifier...
2024/11/18 18:16:57 WARNING miflow.models.model: Model logged without a signature and input example. Please set 'input_example' parameter when logging the model to auto infer the model signature.

Experiment Results:
Run ID: 98c417de5cb7442c67acab6f7575048
Parameters: {'n_topics': 20, 'learning_offset': 35.0, 'kernel': 'poly', 'C': 10.0, 'gamma': 'auto', 'class_balance': 'balanced', 'decision_function_shape': 'ovr'}
Metrics: {'accuracy': 0.39518872289156625, 'weighted_f1': 0.1339313808649420376, 'macro_f1': 0.48852846466119611, 'avg_confidence': np.float64(0.21613934492348027), 'min_confidence': np.float64(0.088731485168464606}, 'max_confidence': np.float64(0.753188605195966)}
2024/11/18 18:17:24 INFO miflow.tracking_tracking_service.client: \ View run experiment_20241118_170310 at: http://localhost:8050/#/experiments/702903838871205330/runs/98c417de5cb7442ca67cab6f755048.
2024/11/18 18:17:24 INFO miflow.tracking_tracking_service.client: \ View run experiment at: http://localhost:8050/#/experiments/702903838871205330.
(env-miflow) ubuntu@ip-172-31-26-71:~\$ nano miflow_svm_news_classification.py
(env-miflow) ubuntu@ip-172-31-26-71:~\$ python3 miflow_svm_news_classification.py

Resumen de validación:
Total de artículos: 2,874
Número de categorías: 27
Longitud promedio del texto: 105.6 caracteres
Cantidad promedio de palabras: 15.4 palabras

Top 5 de categorías por frecuencia:

```
category
POLITICS 338
WELLNESS 258
ENTERTAINMENT 171
PARENTS 123
DIVERSITY VOICES 123
```

Name: count, dtype: int64
Fitting LDA model...
Training SVM classifier...
2024/11/18 18:43:03 WARNING miflow.models.model: Model logged without a signature and input example. Please set 'input_example' parameter when logging the model to auto infer the model signature.

Experiment Results:
Run ID: d0aa998a4dca3e343e2d62d44e2ef
Parameters: {'n_topics': 20, 'learning_offset': 35.0, 'kernel': 'rbf', 'C': 50.0, 'gamma': 'auto', 'class_balance': 'balanced', 'decision_function_shape': 'ovr'}
Metrics: {'accuracy': 0.39518872289156625, 'weighted_f1': 0.2397694238913268, 'macro_f1': 0.297406452984638085, 'min_confidence': np.float64(0.092489702505777879), 'max_confidence': np.float64(0.7523247080088511)}
2024/11/18 18:43:21 INFO miflow.tracking_tracking_service.client: \ View run experiment_20241118_181968 at: http://localhost:8050/#/experiments/702903838871205330.
2024/11/18 18:43:21 INFO miflow.tracking_tracking_service.client: \ View run experiment at: http://localhost:8050/#/experiments/702903838871205330.
(env-miflow) ubuntu@ip-172-31-26-71:~\$ nano miflow_svm_news_classification.py
(env-miflow) ubuntu@ip-172-31-26-71:~\$ python3 miflow_svm_news_classification.py

Resumen de validación:
Total de artículos: 2,874
Número de categorías: 27
Longitud promedio del texto: 105.6 caracteres
Cantidad promedio de palabras: 15.4 palabras

Top 5 de categorías por frecuencia:

```
category
POLITICS 338
WELLNESS 258
ENTERTAINMENT 171
PARENTS 123
DIVERSITY VOICES 123
```

Name: count, dtype: int64
Fitting LDA model...
Training SVM classifier...
2024/11/18 19:09:21 WARNING miflow.models.model: Model logged without a signature and input example. Please set 'input_example' parameter when logging the model to auto infer the model signature.

Experiment Results:
Run ID: 142123e2aed44238d6c9c3377f10349
Parameters: {'n_topics': 20, 'learning_offset': 30.0, 'kernel': 'rbf', 'C': 10.0, 'gamma': 'scale', 'class_balance': 'balanced', 'decision_function_shape': 'ovr'}
Metrics: {'accuracy': 0.39518872289156625, 'weighted_f1': 0.234293911238581205, 'macro_f1': 0.36439811250581205, 'min_confidence': np.float64(0.092489702505777879), 'max_confidence': np.float64(0.7523484656731665)}
2024/11/18 19:09:39 INFO miflow.tracking_tracking_service.client: \ View run experiment_20241118_184526 at: http://localhost:8050/#/experiments/9867222269886334.
2024/11/18 19:09:39 INFO miflow.tracking_tracking_service.client: \ View run experiment at: http://localhost:8050/#/experiments/702903838871205330.
(env-miflow) ubuntu@ip-172-31-26-71:~\$

```
-/Desktop -- ubuntu@ip-172-31-26-71: ~ ssh -i dsa_llave2.pem ubuntu@98.81.210.225 ... -/Desktop -- ubuntu@ip-172-31-26-71: ~ ssh -i dsa_llave2.pem ubuntu@98.81.210.225 ... +
```

Longitud promedio del texto: 105.6 caracteres
Cantidad promedio de palabras: 15.4 palabras

Top 5 de categorías por frecuencia:

```
category
POLITICS 338
WELLNESS 258
ENTERTAINMENT 171
PARENTS 123
DIVERSITY VOICES 123
```

Name: count, dtype: int64
Fitting LDA model...
Training Random Forest classifier...
2024/11/18 13:42:58 WARNING miflow.models.model: Model logged without a signature and input example. Please set 'input_example' parameter when logging the model to auto infer the model signature.

Experiment Results:
Run ID: 844553edfd39424b31b31b164269156
Parameters: {'n_estimators': 500, 'learning_offset': 40.0, 'criterion': 'entropy', 'max_samples': 0.7, 'class_balance': 'balanced'}
Metrics: {'accuracy': 0.4457831325381295, 'weighted_f1': 0.3929193158346347, 'macro_f1': 0.23648281404878257}
2024/11/18 13:42:51 INFO miflow.tracking_tracking_service.client: \ View run experiment_20241118_134123 at: http://localhost:8050/#/experiments/9867222269886334.
2024/11/18 13:42:51 INFO miflow.tracking_tracking_service.client: \ View run experiment at: http://localhost:8050/#/experiments/9867222269886334.
(env-miflow) ubuntu@ip-172-31-26-71:~\$ nano miflow_rf_news_classification.py
(env-miflow) ubuntu@ip-172-31-26-71:~\$ python3 miflow_rf_news_classification.py

Resumen de validación:
Total de artículos: 2,874
Número de categorías: 27
Longitud promedio del texto: 105.6 caracteres
Cantidad promedio de palabras: 15.4 palabras

Top 5 de categorías por frecuencia:

```
category
POLITICS 338
WELLNESS 258
ENTERTAINMENT 171
PARENTS 123
DIVERSITY VOICES 123
```

Name: count, dtype: int64
Fitting LDA model...
Training Random Forest classifier...
2024/11/18 13:44:39 WARNING miflow.models.model: Model logged without a signature and input example. Please set 'input_example' parameter when logging the model to auto infer the model signature.

Experiment Results:
Run ID: efbdb0449f8649f5a2e869d899cf0e2
Parameters: {'n_estimators': 20, 'learning_offset': 50.0, 'criterion': 'entropy', 'max_samples': 0.7, 'class_balance': 'balanced'}
Metrics: {'accuracy': 0.4373493979936315, 'weighted_f1': 0.3834969098985141, 'macro_f1': 0.2599468489760633}
2024/11/18 13:44:41 INFO miflow.tracking_tracking_service.client: \ View run experiment_20241118_134518 at: http://localhost:8050/#/experiments/9867222269886334.
2024/11/18 13:44:41 INFO miflow.tracking_tracking_service.client: \ View run experiment at: http://localhost:8050/#/experiments/9867222269886334.
(env-miflow) ubuntu@ip-172-31-26-71:~\$ nano miflow_rf_news_classification.py
(env-miflow) ubuntu@ip-172-31-26-71:~\$ python3 miflow_rf_news_classification.py

Resumen de validación:
Total de artículos: 2,874
Número de categorías: 27
Longitud promedio del texto: 105.6 caracteres
Cantidad promedio de palabras: 15.4 palabras

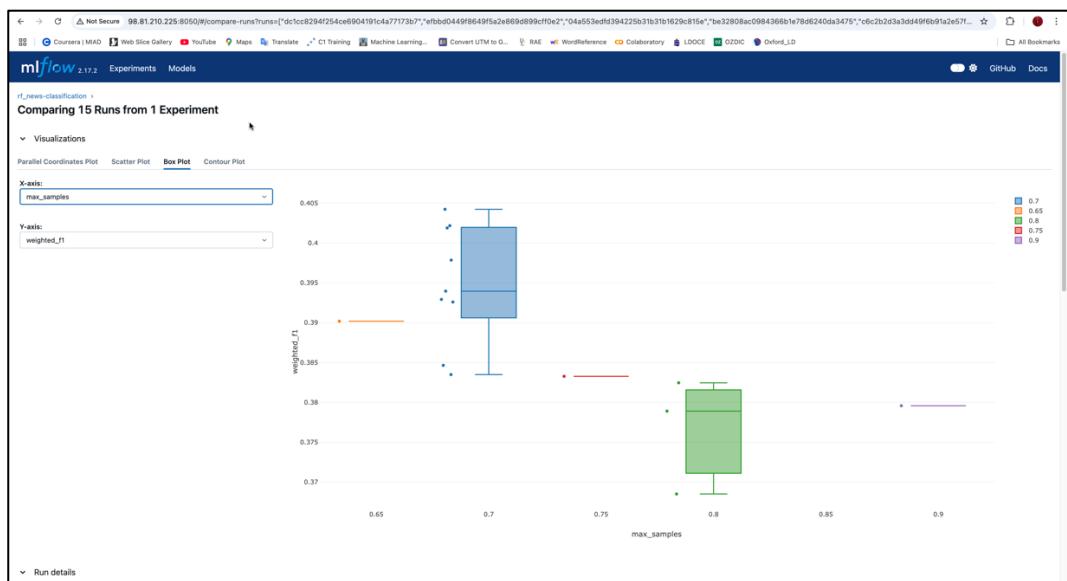
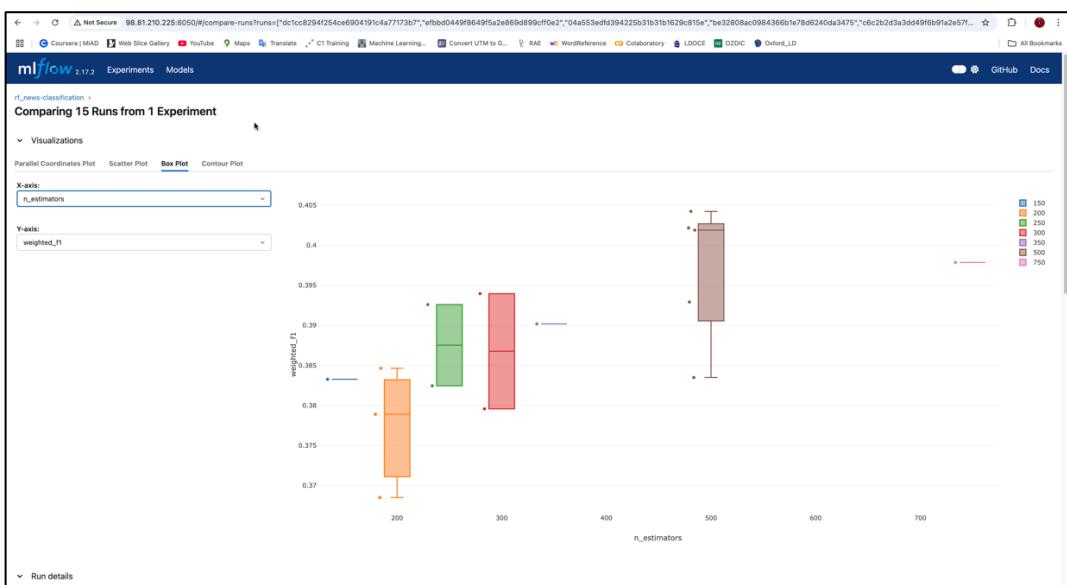
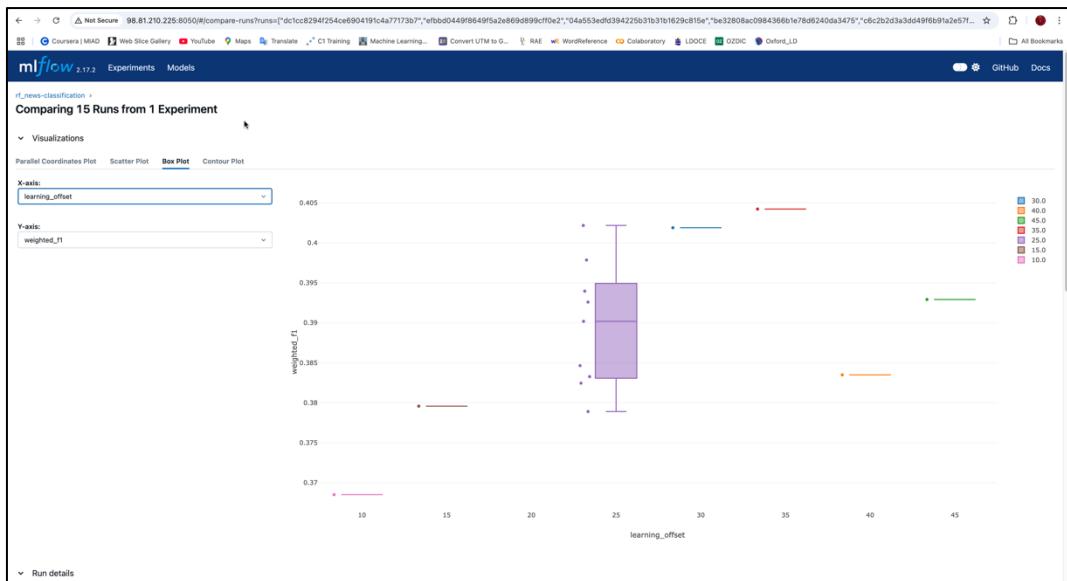
Top 5 de categorías por frecuencia:

```
category
POLITICS 338
WELLNESS 258
ENTERTAINMENT 171
PARENTS 123
DIVERSITY VOICES 123
```

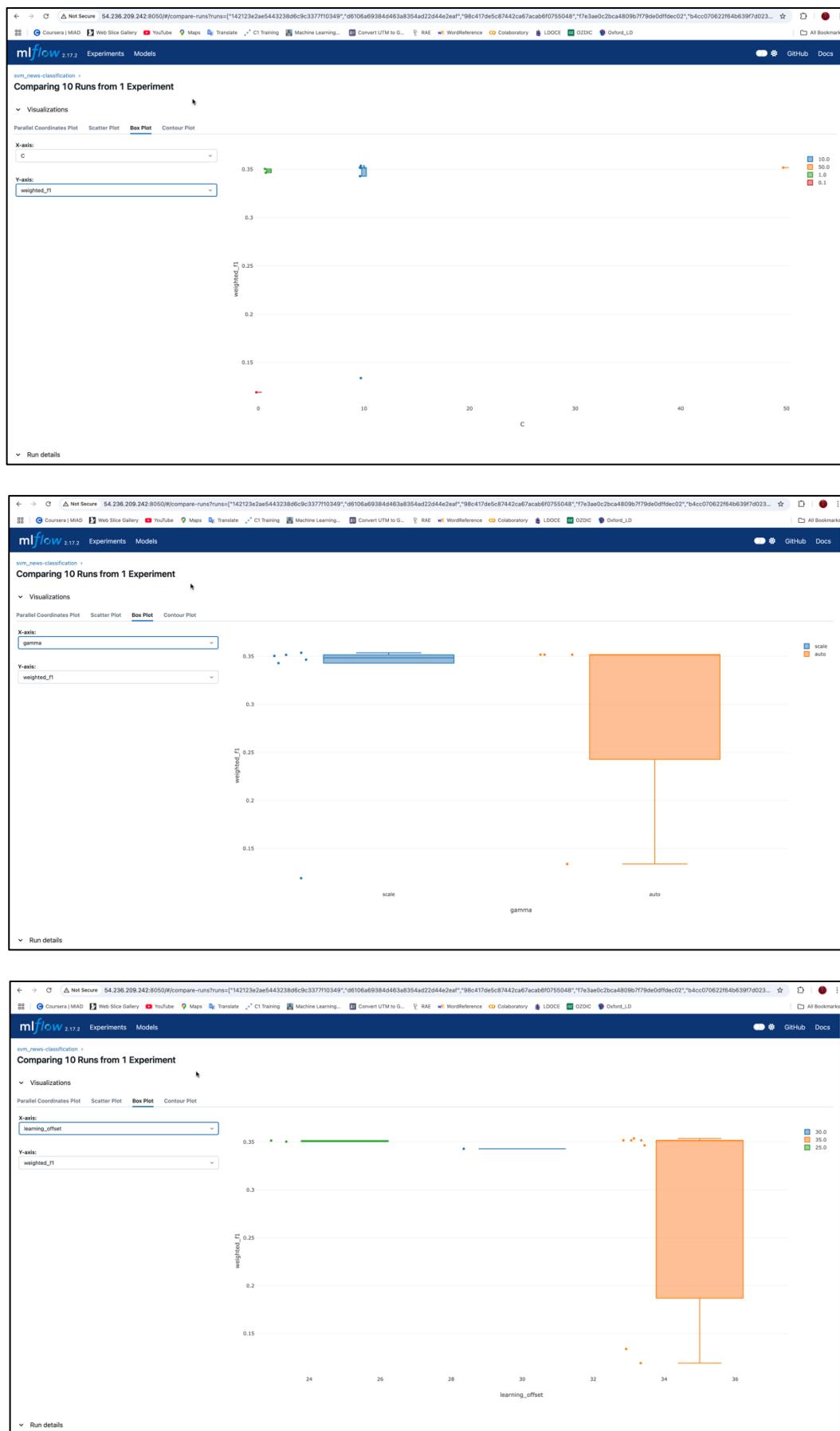
Name: count, dtype: int64
Fitting LDA model...
Training Random Forest classifier...
2024/11/18 13:59:41 WARNING miflow.models.model: Model logged without a signature and input example. Please set 'input_example' parameter when logging the model to auto infer the model signature.

Experiment Results:
Run ID: dc1cc2942f54de6984191c4a77173b7
Parameters: {'n_estimators': 20, 'learning_offset': 50.0, 'criterion': 'entropy', 'max_samples': 0.7, 'class_balance': 'balanced'}
Metrics: {'accuracy': 0.4373493979936315, 'weighted_f1': 0.3834969098985141, 'macro_f1': 0.2599468489760633}
2024/11/18 13:59:43 INFO miflow.tracking_tracking_service.client: \ View run experiment_20241118_135815 at: http://localhost:8050/#/experiments/9867222269886334.
2024/11/18 13:59:43 INFO miflow.tracking_tracking_service.client: \ View run experiment at: http://localhost:8050/#/experiments/9867222269886334.
(env-miflow) ubuntu@ip-172-31-26-71:~\$

ANEXO -2. ESTADISTICAS DE LOS RESULTADOS DE LOS EXPERIMENTOS MODELO LDA-RF



ANEXO -3. ESTADISTICAS DE LOS RESULTADOS DE LOS EXPERIMENTOS MODELO LDA-SVM



ANEXO -4. REPOSITORIO GIT DEL PROYECTO

The screenshot shows a GitHub repository page for 'miad2024_dsa_G16'. The repository is public and owned by 'jhermoher'. It has 2 branches and 0 tags. The main branch contains 14 commits from 'jhermoher' with various commit messages related to project structure and model experiments. A file named 'README' is present. The repository description is 'Proyecto final del curso de despliegue de soluciones analíticas. 2024-15'. The 'About' section includes links to Readme, Activity, and Contributors. The 'Contributors' section lists 'jhermoher', 'andfelipe1', and 'jerssonMH'. The 'Languages' section shows a horizontal bar with orange and blue segments.

jhhermoher / miad2024_dsa_G16

Code Issues Pull requests Actions Projects Security Insights Settings

Type to search

Unpin Unwatch 1 Fork 0 Star 0

miad2024_dsa_G16 Public

main 2 Branches 0 Tags Go to file Add file Code

jhermoher venv/ dir removal 5413d9d · now 14 Commits

.dvc Adición S3 como remote 2 weeks ago

dashboard Add project structure with folders 2 weeks ago

data Models LDA-RF & LDA-SVM. Experiments 9 minutes ago

docs Add project structure with folders 2 weeks ago

notebooks Models LDA-RF & LDA-SVM. Experiments 9 minutes ago

src Models LDA-RF & LDA-SVM. Experiments 9 minutes ago

submittals Models LDA-RF & LDA-SVM. Experiments 9 minutes ago

test Models LDA-RF & LDA-SVM. Experiments 9 minutes ago

.dvcignore Initialize DVC 2 weeks ago

.gitignore Initial project setup 2 weeks ago

README.md Update README.md 2 weeks ago

README

About

Proyecto final del curso de despliegue de soluciones analíticas. 2024-15

Readme Activity 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Contributors 3

jhermoher andfelipe1 jerssonMH

Languages

CATEGORIZACIÓN Y ANÁLISIS DE SENTIMIENTOS DE ARTÍCULOS DE NOTICIAS 📺

ANEXO -5. INSTANCIA EC2 EN AWS.

The screenshot shows the AWS Management Console interface for an EC2 instance. The left sidebar contains navigation links for Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Load Balancers, Target Groups, and Trusted Advisors. The main content area displays the instance details for 'i-008278d7b63a2a072' (news_categorization_instance). The 'Details' tab is selected, showing the following configuration:

Setting	Value	Setting	Value
Instance ID	i-008278d7b63a2a072	Public IPv4 address	54.236.209.242 open address
IPv6 address	-	Instance state	Running
Hostname type	IP name: ip-172-31-26-71.ec2.internal	Private IP DNS name (IPv4 only)	ip-172-31-26-71.ec2.internal
Savings Plans	IPv4 (A)	Instance type	t2.large
Reserved Instances	Answer private resource DNS name	VPC ID	vpc-03cce35ec3bdc2957
Dedicated Hosts	Auto-assigned IP address	Subnet ID	subnet-0a688ef0ffac6e0fb
Capacity	54.236.209.242 [Public IP]	Instance ARN	arn:aws:ec2:us-east-1:541512327398:instance/i-008278d7b63a2a072
Reservations	New		
Images	IAM Role		
AMIs	-		
AMI Catalog	IMDSv2 Required		
Elastic Block Store	Details Status and alarms Monitoring Security Networking Storage Tags		
Volumes	Instance details Info		
Snapshots	Platform Ubuntu	AMI ID	ami-0866a3c8686eaeeba
Lifecycle Manager	Platform details Linux/UNIX	AMI name	ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20240927
Network & Security	Stop protection Disabled	Launch time	Sun Nov 10 2024 18:45:35 GMT+0300 (Arabian Standard Time) (about 4 hours)
Security Groups	Instance auto-recovery Default	Lifecycle	normal
Elastic IPs	AMI Launch index 0	Key pair assigned at launch	dsalave2
Placement Groups			
Key Pairs			
Network Interfaces			
Load Balancing			
Load Balancers			
Target Groups			
Trusted Advisors			

At the bottom right of the main content area, there are links for 'AWS Compute Optimizer finding', 'Opt-in to AWS Compute Optimizer for recommendations.', 'Learn more', 'Auto Scaling Group name', and 'Amazon CloudWatch Metrics'. The footer of the page includes links for 'CloudShell', 'Feedback', '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.