Tecnológico de Monterrey

MLOps Project – Phase 1: Online News Popularity

Predictive modeling of article popularity using supervised ML and MLOps principles

Online News Popularity_

Our team (#41)

Steven Sebastian Brutscher Cortez
A01732505
Data Scientist

Felipe de Jesús Gutiérrez Dávila
A01360023
ML Engineer

Ana Karen Estupiñán Pacheco
A01796893
Software Engineer

Angel Iván Ahumada Arguelles
A00398508
Data Engineer

Content Table

1 - Introduction & Problem

Context:

The dataset Online News Popularity (UCI Machine Learning Repository, 2015) contains articles of MAshable with 61 attributres that describer the content, publication day and social media score.

Problem to solve:

Predict an article's popularity prior to publication to optimize editorial and advertising resources.

Modified Dataset:

The instructors supplied a noise-added, internally inconsistent dataset to evaluate the team's ability to clean, validate, and statistically reconstruct the data.

About the Dataset

Property	Description
Name	Online News Popularity
Source	UCI Machine Learning Repository (Fernandes et al., 2015)
Instances	39,644
Attributes	61 total (58 predictive, 2 non-predictive, 1 target)
Target Variable	`shares` (number of times the article was shared)
Objective	Predict or classify the popularity of a news article
Period Covered	January 2013 – January 2015
Data Type	Mixed: numerical (continuous/discrete) and categorical
Languages Used	English articles from Mashable.com

2 - Análisis del problema: Machine Learning Canvas

Task type:

Supervised (Regression and Classification)

Entity:

News article on Mashable

Objective:

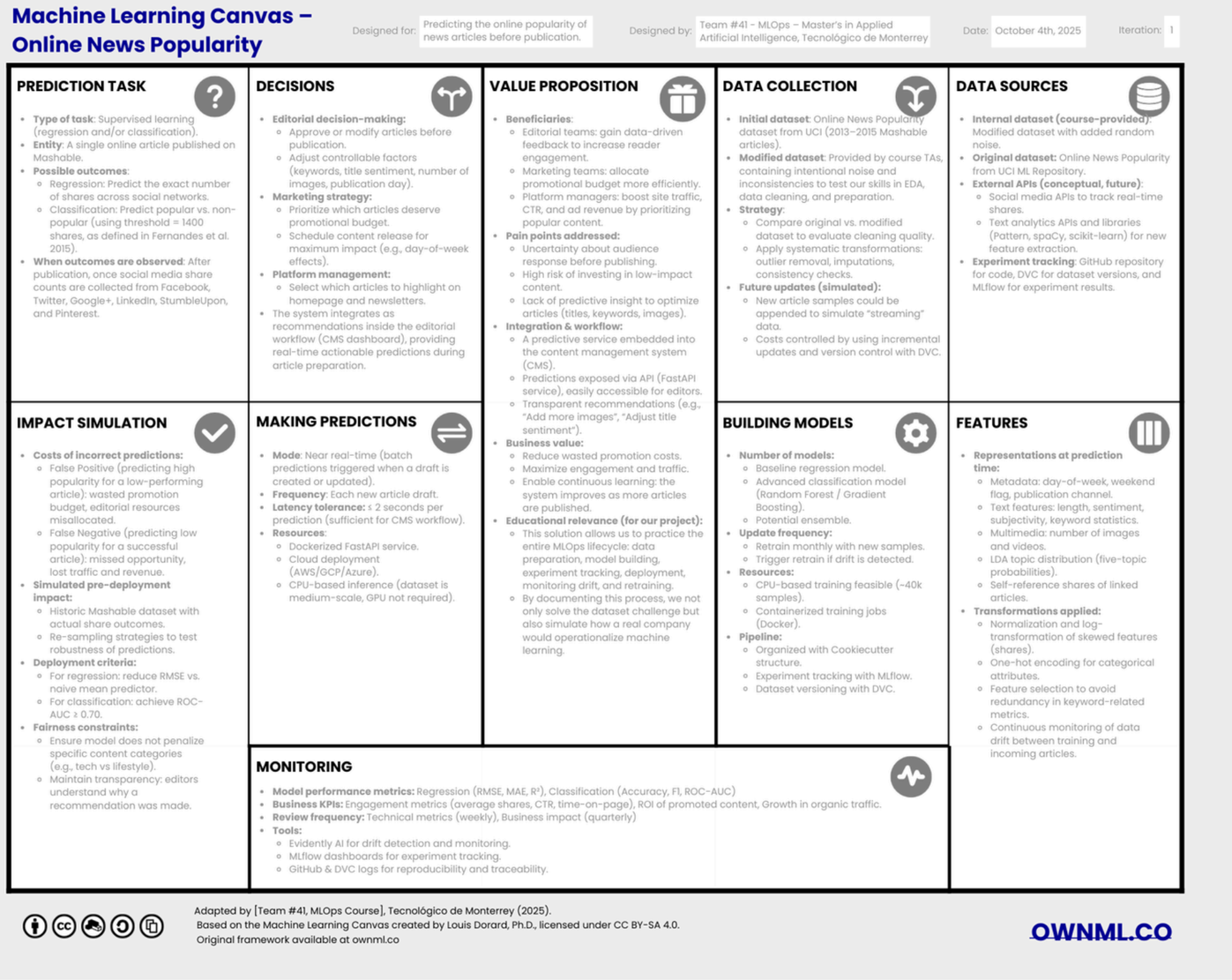
Predict the number of shares or the probability of being popular (> 1,400 shares)

Error costs:

- **False positive:** Over-promoting low-impact articles.
- **False negative:** Losing visibility on potentially viral articles.

Value proposition:

Optimize content planning and maximize efficiency in digital marketing campaigns through data-driven predictions.



Machine Learning Canvas – Online News Popularity

Designed for: Predicting the online popularity of news articles before publication.

Designed by: Team #41 - MLOps – Master's in Applied Artificial Intelligence, Tecnológico de Monterrey

Date: October 4th, 2025

Iteration: 1

PREDICTION TASK

- **Type of task**: Supervised learning (regression and/or classification).
- **Entity**: A single online article published on Mashable.
- **Possible outcomes**:
  - Regression: Predict the exact number of shares across social networks.
  - Classification: Predict popular vs. non-popular (using threshold = 1400 shares, as defined in Fernandes et al. 2015).
- **When outcomes are observed**: After publication, once social media share counts are collected from Facebook, Twitter, Google+, LinkedIn, StumbleUpon, and Pinterest.

DECISIONS

- **Editorial decision-making:**
  - Approve or modify articles before publication.
  - Adjust controllable factors (keywords, title sentiment, number of images, publication day).
- **Marketing strategy:**
  - Prioritize which articles deserve promotional budget.
  - Schedule content release for maximum impact (e.g., day-of-week effects).
- **Platform management:**
  - Select which articles to highlight on homepage and newsletters.
- The system integrates as recommendations inside the editorial workflow (CMS dashboard), providing real-time actionable predictions during article preparation.

VALUE PROPOSITION

- **Beneficiaries**:
  - Editorial teams: gain data-driven feedback to increase reader engagement.
  - Marketing teams: allocate promotional budget more efficiently.
  - Platform managers: boost site traffic, CTR, and ad revenue by prioritizing popular content.
- **Pain points addressed:**
  - Uncertainty about audience response before publishing.
  - High risk of investing in low-impact content.
  - Lack of predictive insight to optimize articles (titles, keywords, images).
- **Integration & workflow:**
  - A predictive service embedded into the content management system (CMS).
  - Predictions exposed via API (FastAPI service), easily accessible for editors.
  - Transparent recommendations (e.g., "Add more images", "Adjust title sentiment").
- **Business value:**
  - Reduce wasted promotion costs.
  - Maximize engagement and traffic.
  - Enable continuous learning: the system improves as more articles are published.
- **Educational relevance (for our project):**
  - This solution allows us to practice the entire MLOps lifecycle: data preparation, model building, experiment tracking, deployment, monitoring drift, and retraining.
  - By documenting this process, we not only solve the dataset challenge but also simulate how a real company would operationalize machine learning.

DATA COLLECTION

- **Initial dataset**: Online News Popularity dataset from UCI (2013–2015 Mashable articles).
- **Modified dataset**: Provided by course TAs, containing intentional noise and inconsistencies to test our skills in EDA, data cleaning, and preparation.
- **Strategy**:
  - Compare original vs. modified dataset to evaluate cleaning quality.
  - Apply systematic transformations: outlier removal, imputations, consistency checks.
- **Future updates (simulated):**
  - New article samples could be appended to simulate "streaming" data.
  - Costs controlled by using incremental updates and version control with DVC.

DATA SOURCES

- **Internal dataset (course-provided)**: Modified dataset with added random noise.
- **Original dataset:** Online News Popularity from UCI ML Repository.
- **External APIs (conceptual, future)**:
  - Social media APIs to track real-time shares.
  - Text analytics APIs and libraries (Pattern, spaCy, scikit-learn) for new feature extraction.
- **Experiment tracking**: GitHub repository for code, DVC for dataset versions, and MLflow for experiment results.

IMPACT SIMULATION

- **Costs of incorrect predictions:**
  - False Positive (predicting high popularity for a low-performing article): wasted promotion budget, editorial resources misallocated.
  - False Negative (predicting low popularity for a successful article): missed opportunity, lost traffic and revenue.
- **Simulated pre-deployment impact:**
  - Historic Mashable dataset with actual share outcomes.
  - Re-sampling strategies to test robustness of predictions.
- **Deployment criteria:**
  - For regression: reduce RMSE vs. naive mean predictor.
  - For classification: achieve ROC-AUC ≥ 0.70.
- **Fairness constraints:**
  - Ensure model does not penalize specific content categories (e.g., tech vs lifestyle).
  - Maintain transparency: editors understand why a recommendation was made.

MAKING PREDICTIONS

- **Mode**: Near real-time (batch predictions triggered when a draft is created or updated).
- **Frequency**: Each new article draft.
- **Latency tolerance:** ≤ 2 seconds per prediction (sufficient for CMS workflow).
- **Resources**:
  - Dockerized FastAPI service.
  - Cloud deployment (AWS/GCP/Azure).
  - CPU-based inference (dataset is medium-scale, GPU not required).

BUILDING MODELS

- **Number of models:**
  - Baseline regression model.
  - Advanced classification model (Random Forest / Gradient Boosting).
  - Potential ensemble.
- **Update frequency:**
  - Retrain monthly with new samples.
  - Trigger retrain if drift is detected.
- **Resources:**
  - CPU-based training feasible (~40k samples).
  - Containerized training jobs (Docker).
- **Pipeline:**
  - Organized with Cookiecutter structure.
  - Experiment tracking with MLflow.
  - Dataset versioning with DVC.

FEATURES

- **Representations at prediction time:**
  - Metadata: day-of-week, weekend flag, publication channel.
  - Text features: length, sentiment, subjectivity, keyword statistics.
  - Multimedia: number of images and videos.
  - LDA topic distribution (five-topic probabilities).
  - Self-reference shares of linked articles.
- **Transformations applied:**
  - Normalization and log-transformation of skewed features (shares).
  - One-hot encoding for categorical attributes.
  - Feature selection to avoid redundancy in keyword-related metrics.
  - Continuous monitoring of data drift between training and incoming articles.

MONITORING

- **Model performance metrics:** Regression (RMSE, MAE, R^2), Classification (Accuracy, F1, ROC-AUC)
- **Business KPIs:** Engagement metrics (average shares, CTR, time-on-page), ROI of promoted content, Growth in organic traffic.
- **Review frequency:** Technical metrics (weekly), Business impact (quarterly)
- **Tools:**
  - Evidently AI for drift detection and monitoring.
  - MLflow dashboards for experiment tracking.
  - GitHub & DVC logs for reproducibility and traceability.

3 - Data Manipulation & Preparation

Owner: Steven Brutscher (Data Scientist)

Main Activities:

- Detect and correct anomalies in numeric and categorical data.
- Validate domains using logical and statistical checks.
- Reconstruct distributions via clipping and smoothing of tails.
- Normalize LDA components and verify inter-feature correlations.

Results:

- df_final_validated.csv is ready for modeling.
- Average correlation-matrix gap |Δ| = 0.0297 → indicates strong structural preservation.
- Binary features are ~identical to the original (JSD ≈ 0.003–0.004).
- Suggested visual: a compact table reporting KS p-values and JSD.

	Metric	Modified Dataset	Final Validated Dataset
0	Rows	40436.00	39235.00
1	Columns	62.00	62.00
2	Missing values (%)	1.27	0.00
3	Numeric columns	0.00	61.00
4	Object columns	62.00	1.00
5	shares — mean	4374.46	2905.62
6	shares — std	35263.99	4065.54
7	shares — min	1.00	1.00
8	shares — max	5185800.00	21513.91

Top 10 numeric columns with largest mean difference (optional v:

	column	mean_mod	mean_final	abs_diff
3	kw_max_max	940137.50	943967.07	3829.56
16	shares	4374.46	2905.62	1468.84
44	kw_min_avg	1500.71	1126.32	374.39
20	kw_avg_max	351611.47	351818.51	207.04
2	timedelta	546.41	354.07	192.34
45	kw_min_max	19542.09	19386.06	156.03
38	n_tokens_content	688.48	537.92	150.56
43	kw_avg_min	412.35	290.44	121.90
29	self_reference_max_shares	12732.15	12624.81	107.34
60	self_reference_avg_sharess	8007.97	7958.57	49.40

4. Exploratory Analysis and Data Preprocessing

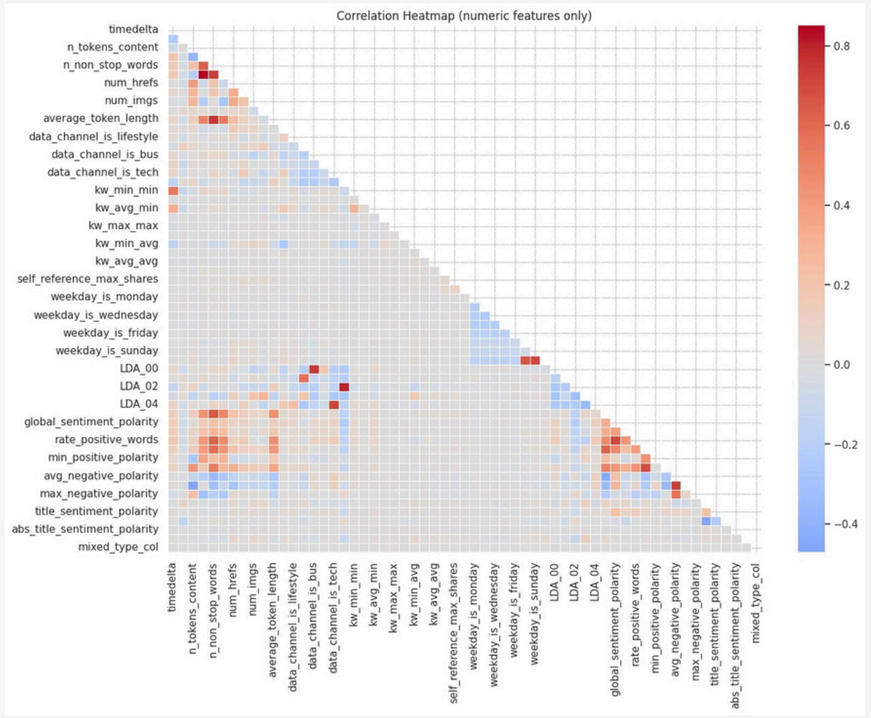
Owner: Felipe (ML Engineer)

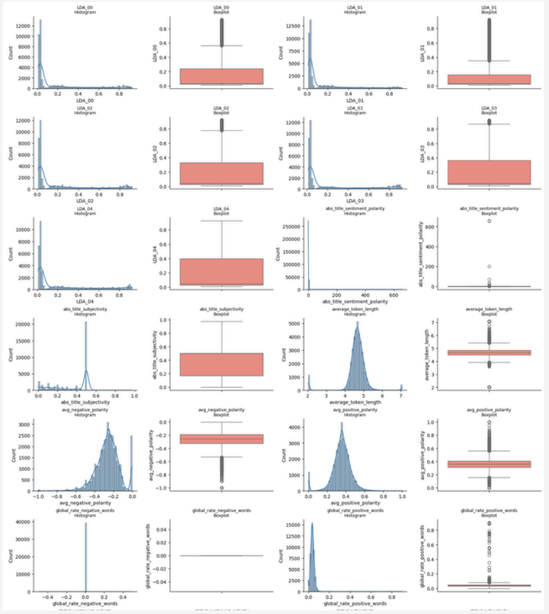
Activities:

- Univariate & bivariate EDA (distribution, correlation, and outlier visuals).
- Normalize and scale numeric features.
- One-hot encode categorical fields.
- Drop redundant variables and reduce dimensionality.

Results:

- Preprocessed & balanced data, ready for the training phase.
- Identify the most impactful/relevant features. (keyword prominence, sentiment polarity, weekday).





5.1 - Data Versioning with DVC
Objective & pipeline

Owner: Ana Karen Estupiñan Pacheco (Software Engineer)

General objective:

Build a reproducible pipeline to predict the reach of online news (shares), ensuring quality control, data and model traceability, and auditable evidence.

Reproducible pipeline with DVC (3 main stages):

- **Clean:** normalizes and validates data, generates a standardized clean.csv.
- **EDA:** produces automated visual reports and descriptive metrics.
- **Train:** trains and versions models, storing artifacts and metrics.

Versioned deliverables:

- HTML/IPYNB reports for cleaning, EDA, and training.
- **Serialized models:** Linear Regression and K-Nearest Neighbors.
- Metrics saved to CSV and JSON for dashboards and comparisons.

5.2 - Data Versioning with DVC Results and Implications

Owner: Ana Karen Estupiñan Pacheco (Software Engineer)

Reproducible execution:
A single, cleaned notebook was consolidated (with no Colab dependencies), executable end-to-end via scripts or DVC commands.

Results (first model iteration):

- **Models tested: Linear Regression and K-Nearest Neighbors (KNN) with hyperparameter search.**
- **Average cross-validation metrics:**
  - **RMSE: ≈ 3,975–4,100 shares**
  - **MAE: ≈ 2,300–2,450 shares**
  - **R^2: low (typical in phenomena with high dispersion)**

Interpretation:
Although the variance of the phenomenon limits precision, the baseline models help identify articles with higher reach potential and provide a solid benchmark for future iterations and feature-engineering improvements.

6 - Model Construction, Tuning and Evaluation (Notebook 03)

Owner: Steven Sebastian Brutscher Cortez (Data Scientist)

Activities:

- Dataset partitioning into train, validation, and test sets.
- Training of multiple baseline regressors: Linear Regression, Ridge, KNN, Decision Tree, Random Forest, and XGBoost.
- Hyperparameter exploration and model validation.
- Evaluation based on MAE, RMSE, and R^2 metrics (on original shares scale).

Results:

- Ensemble models (XGBoost, Random Forest) provided the most consistent and stable performance.
- Best model: XGBoost → RMSE = 4018.48, MAE = 1851.43, R^2 = 0.0116.
- Models confirm high variability and non-linearity in the target variable (shares).
- The selected model serves as the baseline for Phase 2 hyperparameter optimization and explainability analysis.

Additional models (metrics on original shares scale):

	model	features	MAE_train	RMSE_train	R2_train	MAE_test	RMSE_test	R2_test
1	XGBoost	46	1881.58	4058.26	0.0064	1851.43	4018.48	0.0116
0	DecisionTree	46	2759.30	5043.86	-0.5348	2712.27	4930.57	-0.4880

All models summary:

	model	features	MAE_train	RMSE_train	R2_train	MAE_test	RMSE_test	R2_test
4	XGBoost	46	1881.58	4058.26	0.0064	1851.43	4018.48	0.0116
0	RandomForest	46	1896.50	4066.80	0.0022	1878.84	4028.64	0.0066
1	KNN(k=15)	46	1943.93	4153.93	-0.0410	1924.58	4111.83	-0.0348
2	Linear	46	2370.33	61616.47	-228.0446	1911.31	4114.95	-0.0364
3	Ridge(α=5)	46	2370.11	61584.80	-227.8093	1911.31	4114.96	-0.0364
5	DecisionTree	46	2759.30	5043.86	-0.5348	2712.27	4930.57	-0.4880

Current best candidate: XGBoost | RMSE_test=4018.48, MAE_test=1851.43, R2_test=0.0116

7 - Results

Key Insights:

- XGBoost achieved the best overall performance, slightly outperforming all other models.
- Random Forest confirmed the robustness of ensemble methods.
- Simpler models (Linear, Ridge, KNN) struggled to capture non-linear patterns.
- Decision Tree overfitted the data with large error variance.
- Results validate that predicting news virality remains challenging due to strong noise and outlier effects.

Model	Features	MAE (Test)	RMSE (Test)	R^2 (Test)
XGBoost	46	**185.143**	**401.848**	**116**
Random Forest	46	187.884	402.864	66
KNN (k = 15)	46	192.458	411.183	–0.0348
Ridge Regression	46	191.131	411.496	–0.0364
Linear Regression	46	191.131	411.495	–0.0364
Decision Tree	46	271.227	493.057	–0.4880

8 - Role Interaction with MLOps

Data Scientist

- Cleaning and validation of the original data.
- Interaction with a ML Engineer to define feature sets.

ML Engineer

- EDA, preprocessing, and modeling.
- Interaction with a Software Engineer for versioning and logs.

Software Engineer

- DVC, repository management.
- Interaction with a Data Scientist for dataset control.

Data Engineer

- Executive documentation and presentation.
- Interaction with all roles to summarize results.

9 - What’s next?

Phase 2 – Model Versioning and Monitoring

- MLflow for experiment tracking.
- Evidently AI for drift detection.
- Monthly retraining with new data.

Phase 3 – Deployment and Automation

- Containerization with Docker.
- FastAPI service for real-time predictions.
- Integration with a simulated CMS (“editor dashboard”).

10. Conclusions and recommendations

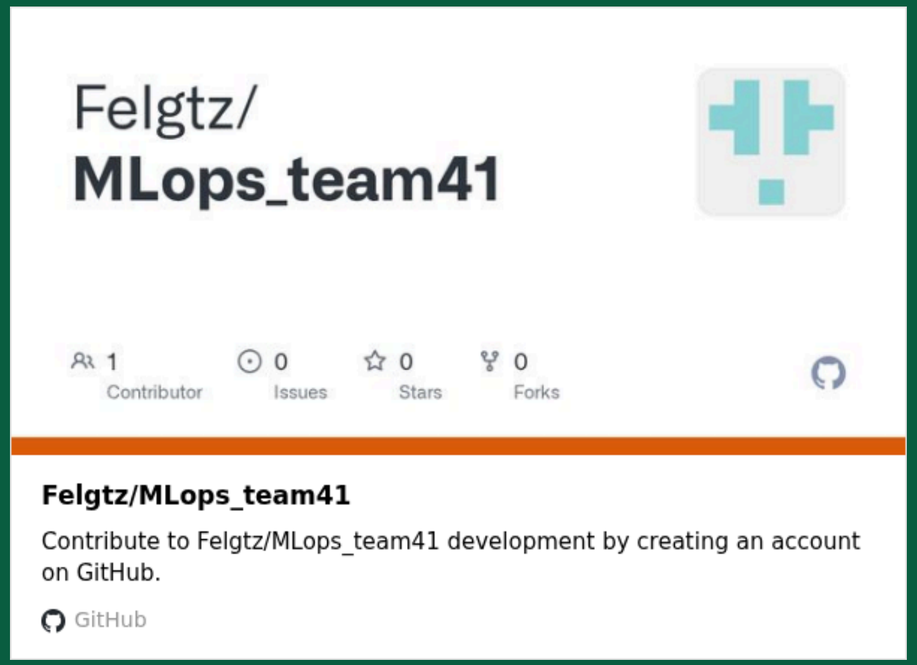
- The dataset was successfully repaired, validated, and preserved, maintaining statistical integrity nearly identical to the original source.
- All datasets, preprocessing pipelines, and models were fully documented and versioned using DVC, ensuring complete reproducibility and traceability.
- Multiple baseline models were trained and evaluated under a consistent MLOps framework, providing reliable performance benchmarks.
- The project is now ready for Phase 2, focused on experiment tracking, model optimization, and deployment to a production-ready environment.

Next Steps:

- Implement MLflow for tracking experiments and model metadata.
- Perform hyperparameter optimization for the XGBoost baseline.
- Integrate explainability tools (e.g., SHAP, LIME) for interpretability.
- Prepare deployment pipeline for cloud or API integration.

Link to our GitHub repository:

https://github.com/Felgtz/MLops_team41



Tecnológico de Monterrey

References

- MLOps Team 41 (2025). Online News Popularity – Phase 1 Repository. GitHub: https://github.com/Felgtz/MLops_team41
- Fernandes, K., Vinagre, J., & Cortez, P. (2015). A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/online+news+popularity
- Dorard, L. (2019). Machine Learning Canvas. Licensed under CC BY-SA 4.0. https://creativecommons.org/licenses/by-sa/4.0/
- Evidently AI. (2024). Data Drift and Model Evaluation Reports. https://docs.evidentlyai.com
- DVC Docs. (2024). Data Version Control (DVC) – Versioning for ML Projects. https://dvc.org/doc
- MLflow Documentation. (2024). Open Source Platform for the Machine Learning Lifecycle. https://mlflow.org/docs/latest/index.html
- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)