Dataverse4.0 Hackathon

Viral Classification Challenge

Technical Brief

Organized by DATAVERSE 4.0

Theme: *From Data to AI – Unlocking the Future*

Event Dates: December 8–9, 2025

Location: ENSA Khouribga, Morocco

This document provides all technical details, dataset structure, staged delivery, and evaluation methodology for the Dataverse4.0 Hackathon.

1 Challenge Overview

Objective. Build a supervised machine learning model that predicts whether a social media post will be *viral* based on available metadata (platform, engagement metrics, time) and textual information (pseudo caption). Participants are expected to produce robust, reproducible pipelines, include visualizations, provide a model report in notebook markdowns, and clearly explain their feature engineering and modeling choices.

2 Dataset Description

The dataset consists of approximately 5,000 anonymized social media posts collected from multiple platforms and regions. Each record corresponds to a single post and contains engagement, contextual, and text-related fields.

Columns

- **Post_ID** – Unique identifier for each post.
- **Pseudo_Caption** – A text caption associated with the post (to be used for NLP / embeddings).
- **Post_Date** – Date of publication (YYYY-MM-DD).
- **Platform** – Name of the platform (e.g., TikTok, YouTube, Instagram).
- **Hashtag** – Main hashtag linked to the post.
- **Content_Type** – Type of content (e.g., Video, Reel, Short, Tweet).
- **Region** – Geographic region where the content is published.
- **Views** – Total number of views.
- **Likes** – Total number of likes.
- **Shares** – Total number of shares.
- **Comments** – Total number of comments.
- **Viral** – Target label (1 = Viral, 0 = Not Viral). *(present only in the training file)*

- **Note on Engagement Metrics:** All engagement metrics (Views, Likes, Shares, Comments) are collected within the first 24 hours after the post is published. Participants should treat these as early indicators of potential virality. Using metrics measured after the 24-hour window or final cumulative totals is not allowed to ensure a fair prediction task.

Participants will receive:

- a **training file** containing all features and the `Viral` label,
- a **public validation file** with features only (for inference).

3 Hackathon Phases (30 Hours)

Part 1 – Exploratory Data Analysis (15 pts)

- Inspect missing values and overall data quality.
- Summarize engagement metrics (views, likes, shares, comments).
- Visualize key relationships (e.g., views vs. viral, platform vs. viral).

Evaluation: quality of insights, clarity of visuals, and correctness of basic preprocessing.

Part 2 – Feature Engineering (20 pts)

- Encode categorical variables (`Platform`, `Region`, `Content_Type`).
- Derive date-based features from `Post_Date`: day-of-week, month, year.
- Keep transformations reproducible (**seeded, scripted**).

Evaluation: relevance, originality, and robustness of engineered features.

Part 3 – Model Development & Optimization (35 pts)

- Train multiple supervised models: Logistic Regression, Decision Tree, Random Forest, XGBoost, CatBoost, SVM, etc.
- Optimize primarily for **macro F1** (dataset is not perfectly balanced).
- Include a **Model Report in notebook markdowns only: metrics for all models, hyperparameters, and rationale for selecting the final model**.

Evaluation: predictive performance, interpretability, stability, and quality of the model report.

Part 4 – Embedding Integration (10 pts)

- Generate sentence embeddings from `Pseudo_Caption` using Sentence Transformers.
- Optionally reduce dimensionality (e.g., PCA to 50–100) before merging with tabular features.
- Show the uplift in performance after adding textual features.

Evaluation: quality of integration and demonstrated contribution.

Part 5 – Best Day to Post Analysis (10 pts)

- Use date features to estimate optimal posting days per platform and content type.
- Present results using clear charts or tables.

Evaluation: correctness, clarity, and actionability.

Part 6 – Model Explainability (10 pts)

- Use feature-importance tools (e.g., SHAP, LIME, permutation importance) to explain how features influence predictions.
- Highlight which variables most affect virality across platforms and regions.
- Visualize key drivers using summary plots or heatmaps.

Evaluation: depth of analysis, clarity of explanations, and interpretability.

4 Staged Delivery and Requirements

Phase 1 – Notebook Submission (Required, 8 hours before the hackathon ends)

- Submit a single notebook containing:
  - Include all steps in your notebook:
    - Data exploration (EDA, missing values, summary statistics)
    - Feature engineering pipeline
    - Model training and evaluation with multiple models
    - Model report in notebook markdowns only: metrics, hyperparameters, and rationale for selecting the final model
  - Generate a predictions CSV using your selected model with the following columns: $\textbf{Post}_ID, Viral$
  - **Compress all files (notebook and CSV) into a zip file and name it with your team name**
- **Notebook and CSV submission is mandatory. Teams failing to submit cannot proceed.**

Submission Rules and Timeline

- All notebooks (.ipynb) must be submitted with **executed output cells**.
- Participants are allowed to **start working on Phase 2 elements** (UI, app structure, improved models) even before Phase 1 results are announced.
- The evaluation of Phase 1 is quick. Results will be announced within **30–40 minutes** after the submission deadline.

Only the top 5 notebook submissions will qualify for Phase 2.

Phase 2 – Web App + Presentation

- Build an interactive web application (any stack allowed)
- Dashboard displaying dataset insights and prediction interface
- **Optional bonus features:** admin upload, retrain, model versioning, SHAP/LIME explainability
- pptx presentation

- 10-minute presentation per team covering:
  - Insights from EDA and visualizations
  - Model selection rationale (from markdowns + discussion)
  - Demo of the app
  - Optional bonus features

5 Scoring Summary

Category	Description	Points
Part 1	Exploratory Data Analysis	15
Part 2	Feature Engineering	20
Part 3	Model Development & Optimization	35
Part 4	Embedding Integration	10
Part 5	Best Day to Post Analysis	10
Part 6	Model Explainability	10
Total		**100**

Note: Phase 2 (Web App & Presentation) will account for **40% of the total hackathon score**.

6 Data Processing & Reproducibility Guidelines

- Fit preprocessing components (e.g., `Imputer`, `Encoder`, `PCA`, or embedding generator) **only on the training set**, then apply (`transform`) the same components on validation or test data; Never re-fit preprocessing steps on validation data.
- Maintain consistent feature order, naming, and data types between training and validation.
- Use fixed random seeds for all random operations (e.g., splits, sampling, PCA, model initialization).
- Clearly separate "data preparation" and "model training" sections within your notebook for transparency.

Remember: Today's code might be messy, but tomorrow it might just go viral!