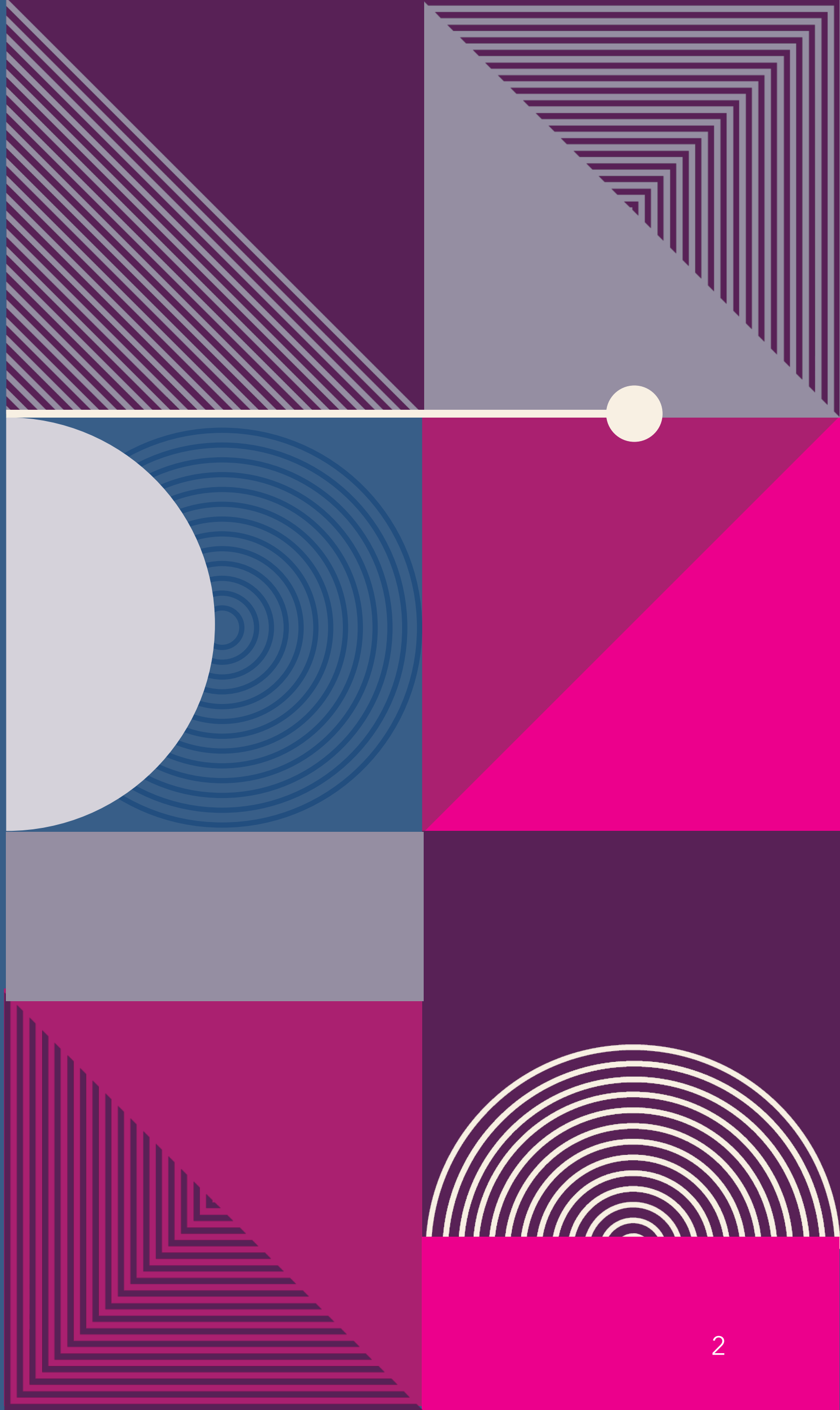
# EXECUTIVE SUMMARY

# CONTEXT

The objective of the project is to forecast article popularity before publication so editors can prioritize what to promote, to do this we performed several analysis including the requirements, exploratory analysis, data preprocessing, ML model creation and adjustment.



# BUSINESS OUTLOOK

- Editorial and promo budgets are being spread across pieces with highly variable outcomes.
- A forward look at expected popularity lets us focus resources where they matter, reduce wasted spend on low-impact content, and lift traffic without increasing costs.
- Success is measured in engagement (shares/CTR/time-on-page) and ROI of promoted content.

# PROBLEM & DATA

- We framed this as a supervised learning problem using the course dataset adapted for realism.
- Features capture timing, content attributes (length, sentiment, keywords, media), and topical signals.
- We cleaned inconsistencies, handled skew and outliers, and assembled a robust preprocessing pipeline so the model can be retrained and shipped safely.

# RESULTS

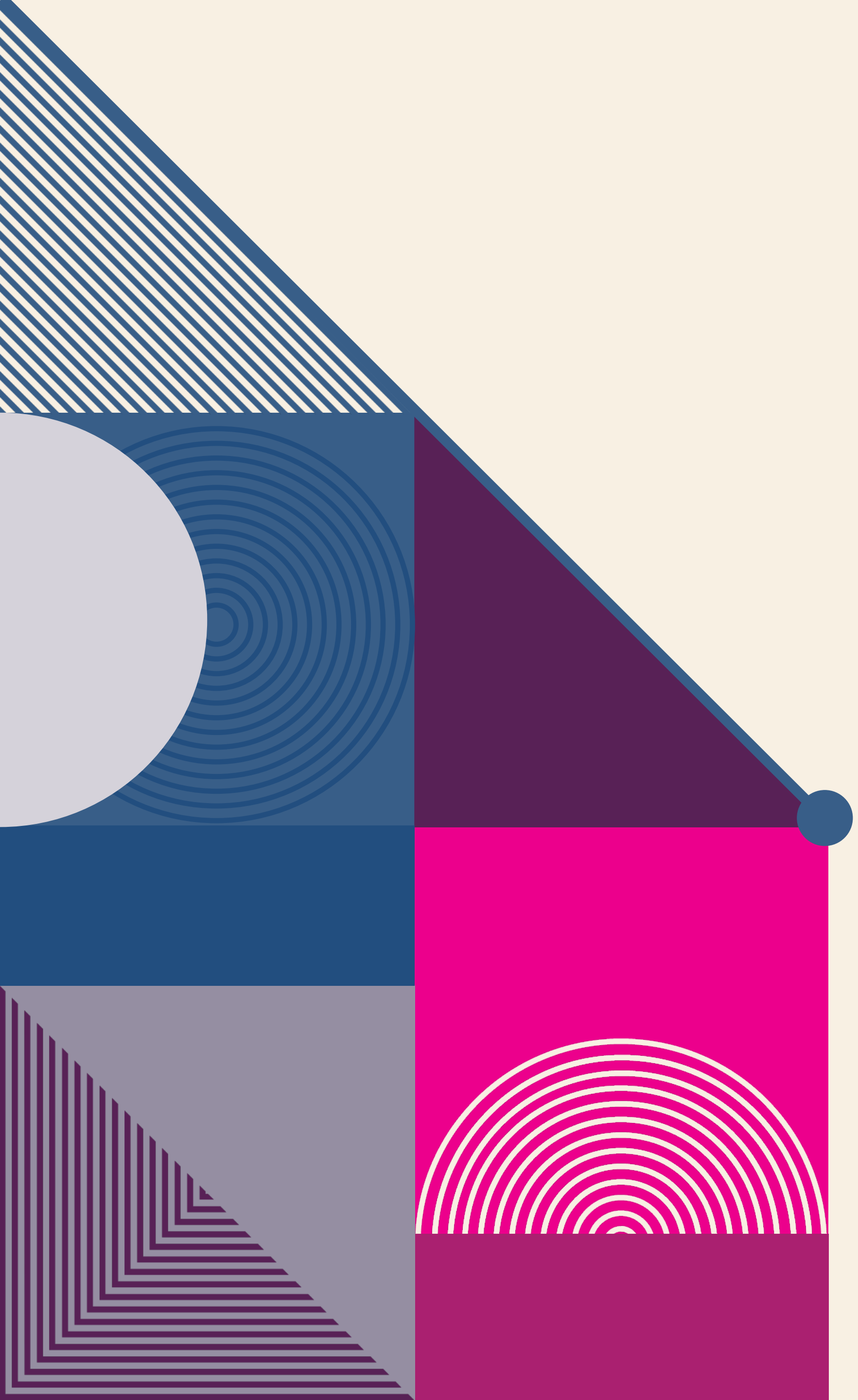
- Using 5×3 Repeated K-Fold CV, the linear regression baseline was the most accurate and stable, with a CV RMSE of ~3,974.6 (fit_intercept=True; positive=False). This meets our bar for a pilot and is fast enough for real-time use in the CMS.
- Our k-nearest neighbors model was close, with a CV RMSE of ~4,014.2 using n_neighbors=21, p=1 (Manhattan distance), and uniform weights. It’s promising as a challenger, though it’s heavier to tune and slower at inference, with the course we will keep modifying this.

# RESULTS

- By contrast, decision trees underperformed even after tuning (criterion=absolute_error, max_depth=7, max_features=sqrt), reaching a CV RMSE of ~23,400, which indicates high variance and a poor fit to this signal.
- Implication: ship the linear model for pilot or test, keep k-NN as a monitored challenger, and deprioritize trees. If the pilot confirms value, our next experiments will test regularized linear models and random forest ensembles to improve, while tracking business impact and model drift in production.

# PILOT PLAN & MLOPS

- We'll expose the model via a small FastAPI service.
- When an editor drafts an article, the CMS requests a prediction and returns an expected-impact score with simple explanations.
- We version data with DVC, track experiments with MLflow, and monitor drift and performance weekly.
- Retraining is scheduled monthly or on drift alerts; all changes go through CI/CD.



# POSSIBLE RISKS

- The main risk is misallocation: over-promoting pieces that won’t perform (false positives) or missing unexpected hits (false negatives).
- We’ll set confidence thresholds and reserve an “exploration” budget to keep discovering surprise winners.
- To avoid bias across categories, we’ll audit performance by section and adjust features and thresholds as needed.

# RECAP