Antony Wei

May 12, 2025

Data bootcamp

Professor Kohler

# Predicting Share Scores of Facebook

## Introduction to Problem & Data

#### Problem Statement:

Over the past decade, social-media virality has become a decisive currency for digital publishers: a single surge of shares can elevate a story’s reach, advertising revenue, and brand stature overnight. For my final project, I will develop a predictive model that forecasts the number of Facebook shares an article will attract, using the Online News Popularity dataset compiled from two years of Mashable content (acquisition date — 8 Jan 2015). The dataset distills each article into 60-plus numeric and categorical attributes—including lexical richness, sentiment polarity, topic-channel flags, and timing features—while omitting the full text to respect Mashable’s content rights.

Accurate share predictions serve multiple stakeholders. Editors and headline writers gain evidence-based guidance on wording, sentiment, and topic mixes that amplify reach. Audience-development teams can time publication to exploit high-impact windows, and advertisers can align campaigns with stories poised for rapid diffusion. Beyond tactical benefits, the model will clarify the broader mechanics of online attention: Which linguistic cues matter most? How do technology or lifestyle beats differ from entertainment in viral potential?

By translating heterogeneous article statistics into actionable forecasts, this project aims to furnish newsrooms and digital-media strategists with a data-driven lens on virality—helping them allocate creative effort and marketing spend where it is most likely to convert into social-network visibility and, ultimately, reader engagement.

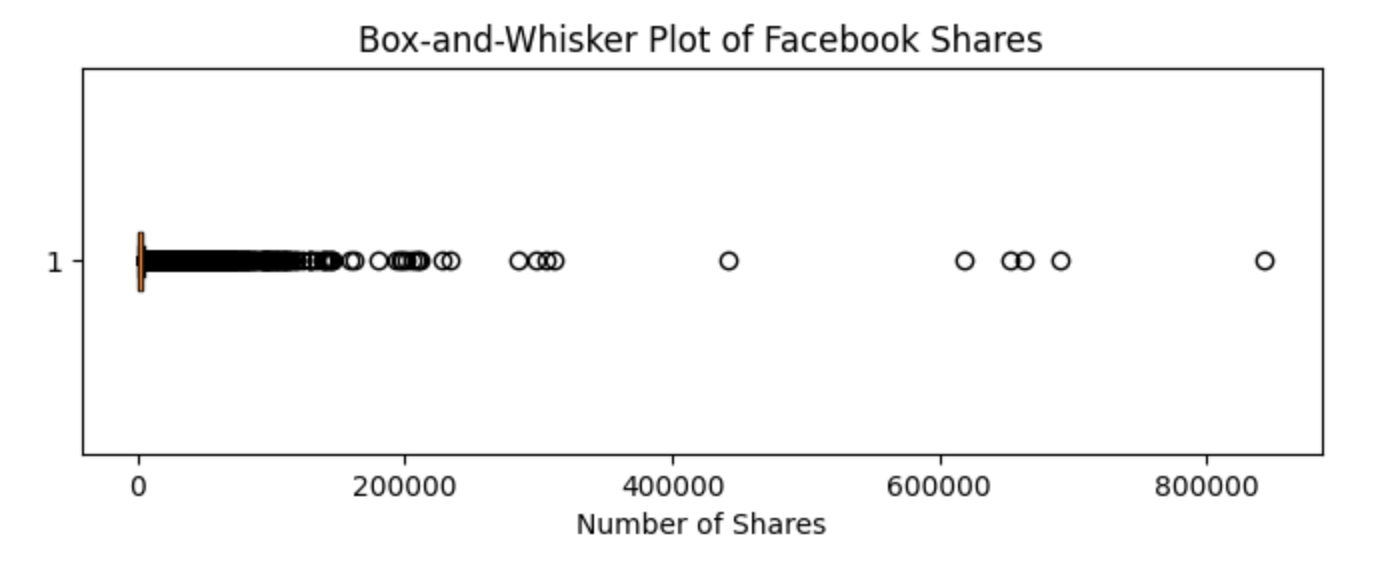
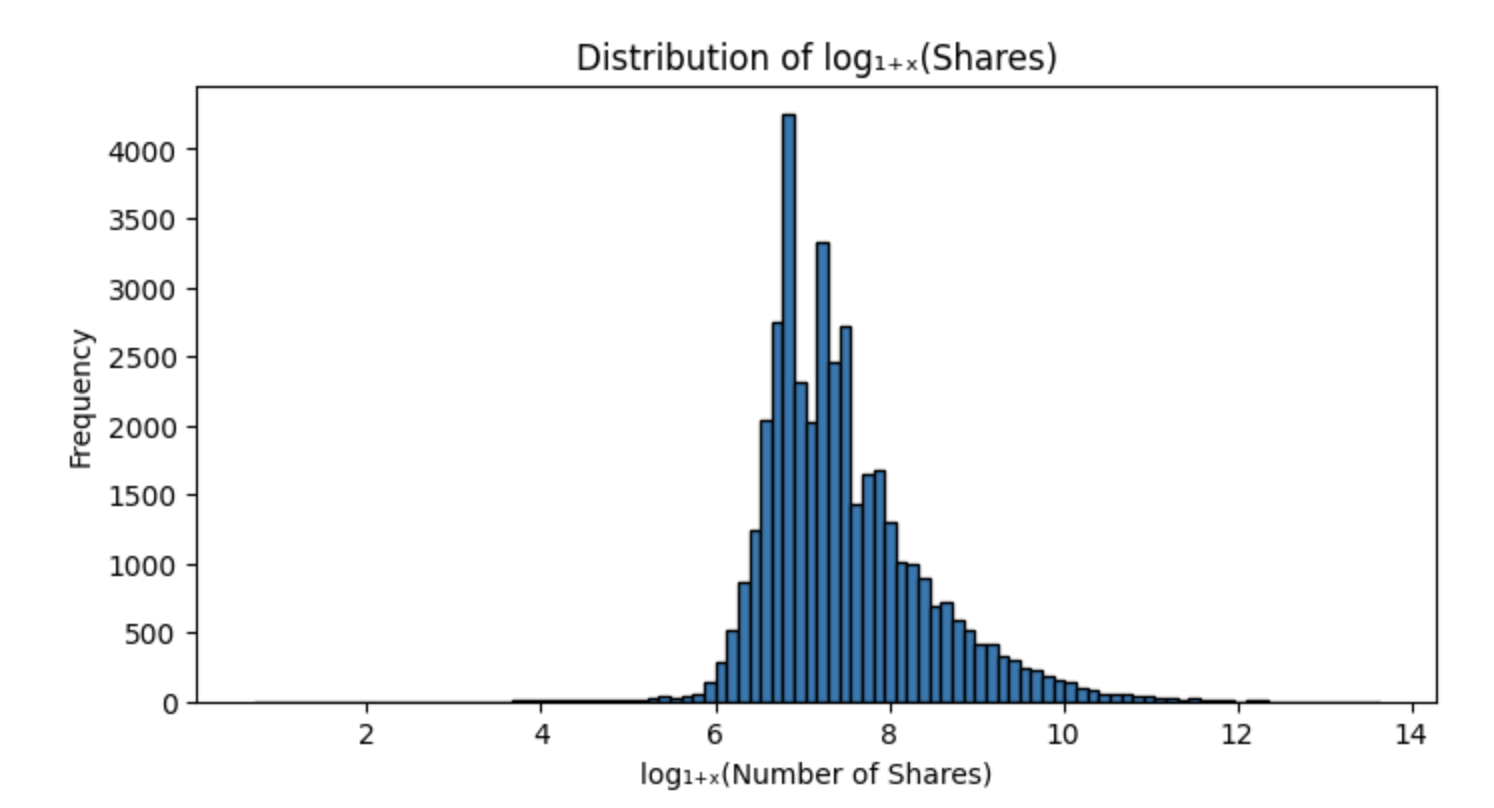
#### Dataset Description:

Data for this project is sourced from the UCI Machine-Learning Repository in CSV format and contains summary statistics for every article Mashable published over a two-year span. Because the original text is under copyright, the file keeps only engineered features—token counts, sentiment scores, topic flags, and publication timing—so some initial wrangling will involve dropping non-informative identifiers such as the raw URL and handling a handful of missing values in a few lexical columns. Modeling popularity is also tricky: Facebook shares are extremely right-skewed (a tiny share of stories “go viral”), so log-transformations or a binary “viral / not viral” label will be considered to counter heavy-tail effects and yield more stable predictions.

The dataset comprises 39,644 articles described by 61 variables: 60 potential predictors plus the target variable shares. Predictors span topical channels (lifestyle, tech, entertainment, etc.), fine-grained lexical and sentiment metrics, keyword statistics, and weekday / weekend indicators. This rich mix of linguistic, topical, and temporal attributes should capture the ingredients of online virality and allow the model to predict each article’s share count—or likelihood of surpassing a “viral” threshold—with useful accuracy.

The revised dataset that I will be working with contains information on 39 644 online news articles published by Mashable between January 2013 and January 2015. Each story is represented only by engineered statistics (to respect copyright): titles span 2 – 18 words, full-text length ranges roughly 45 – 850 words, and a rich set of sentiment, topic, and timing features accompanies every record. The articles’ popularity, measured by Facebook shares, varies dramatically—from a single share to 843 300—giving the project a wide target spectrum for modelling virality.

Exploratory Data Analysis



The log₁₊ₓ transformation has compressed the extreme upper tail and produced a roughly bell-shaped distribution, though a slight right skew remains. Practically, this means the vast majority of articles attract on the order of ≈ 400 – 3 000 shares (log values 6–8), while only a small minority break past log values above 10—i.e., tens of thousands of shares—reflecting true “viral” outliers.

#### Initial Visualizations

Exploratory analysis shows that an article’s historical keyword performance is the single most consistent signal of virality: features such as kw\_avg\_avg and related self-reference metrics display the strongest positive correlations with Facebook shares, while world-news and LDA-topic-2 pieces trend in the opposite direction. Timing and topic interact meaningfully with that signal—publishing lifestyle, tech, or social-media stories on weekends lifts median reach by 30-50 %, and a “sweet spot” emerges where moderately long, mildly positive articles that include a balanced mix of images and outbound links outperform sparse or overly dense pages. Taken together, these findings suggest that share count is driven by a blend of proven keywords, topical focus, and publication window, with length, sentiment, and media richness providing secondary refinements. Moving forward, we will treat shares on a log scale and build a series of predictive pipelines that prioritise the keyword metrics, topic flags, and weekend indicator, augment them with interaction terms (e.g., topic × weekend), and test increasingly expressive models—starting with regularised linear regression and logistic classification for a viral threshold, then advancing to tree-based ensembles that can capture non-linear effects while guarding against over-fitting via cross-validated hyper-parameter tuning

## Modeling & Interpretations

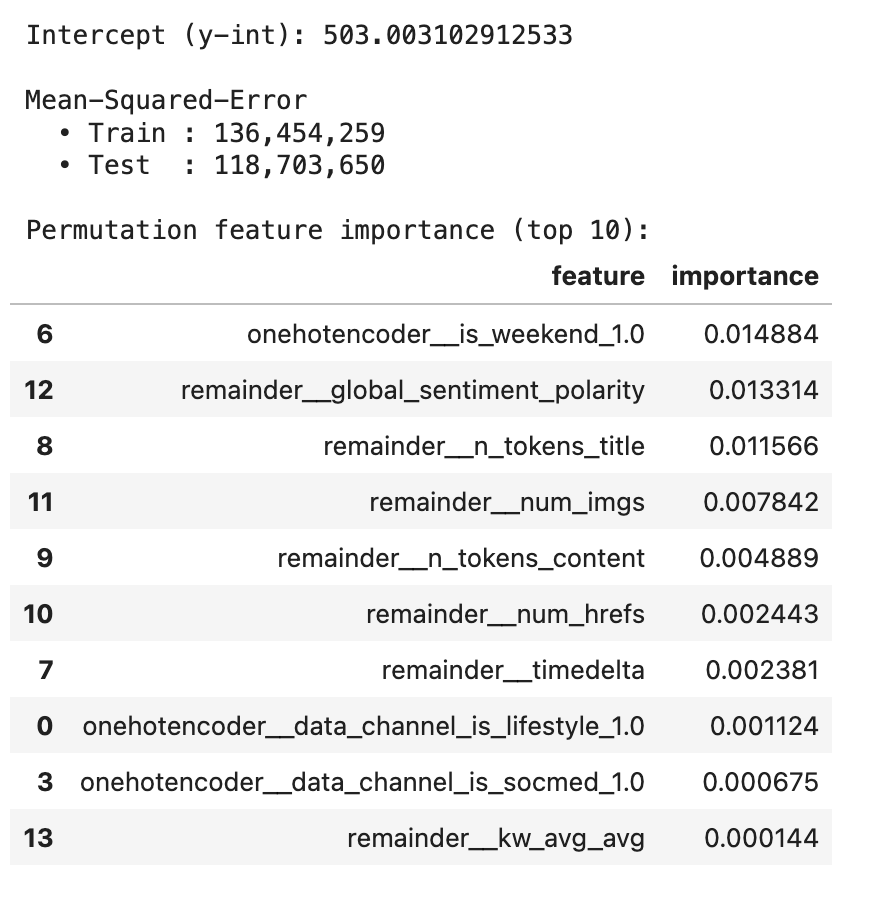
#### Baseline Model

## To establish a reference point, I built a naïve model that predicts the mean number of Facebook shares in the training set for every article; its mean-squared-error serves as the benchmark that all subsequent models must beat.

## 

#### Multiple Regression Model

I chose to build a multiple-regression model so I could use a range of article-level features—timing, length, sentiment, topic flags, and more—to predict the number of Facebook shares, trusting that these factors operate together to shape virality. Multiple linear regression lets me quantify each feature’s individual contribution while capturing their combined influence on an article’s expected reach

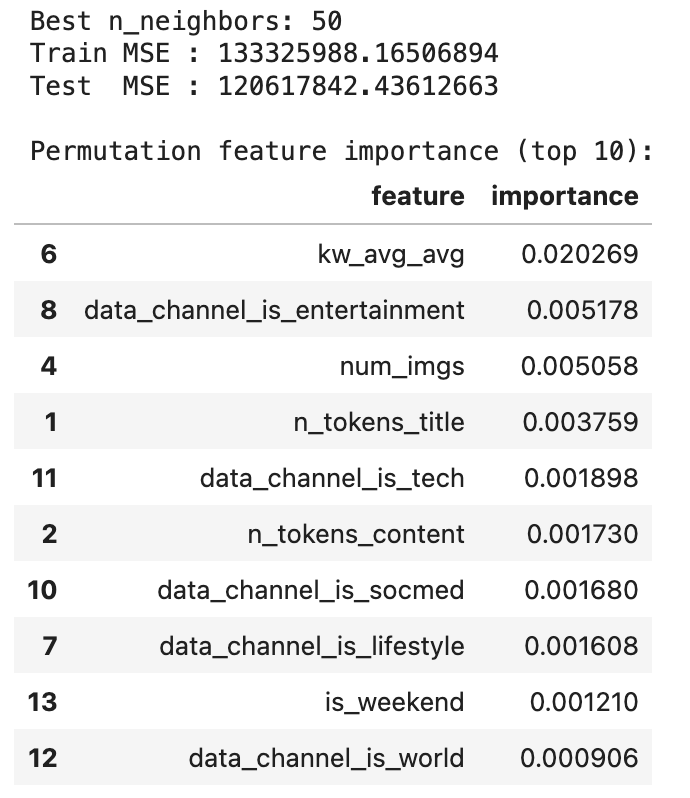
.

I opted for a multiple-linear-regression pipeline so the model could weigh several story attributes at once—timing, length, sentiment, media richness, topic flags—rather than fall back on a single grand average. Compared with the naïve baseline (MSE ≈ 120.7 million), the regression improves modestly on the untouched hold-out set (test MSE ≈ 118.7 million), signalling that the chosen predictors capture a slice—though not the majority—of the huge variance in Facebook-share counts. The training error (≈ 136.5 million) sits slightly above the test error, suggesting the linear model is not over-fitting but simply bumping into the limits of a heavy-tailed target that a straight line can’t fully tame. Permutation importance shows the weekend indicator is the single most influential feature, followed closely by overall sentiment polarity, title length, and the count of images; keyword history, surprisingly, ranks much lower in this linear setting. In short, the regression edges past the mean-only benchmark by leveraging timing and surface-level content cues, yet the still-large MSE hints that richer, non-linear methods (or a log-transformed target) will be needed for a substantial performance jump.

#### 

#### K-Nearest Neighbors Regression Model

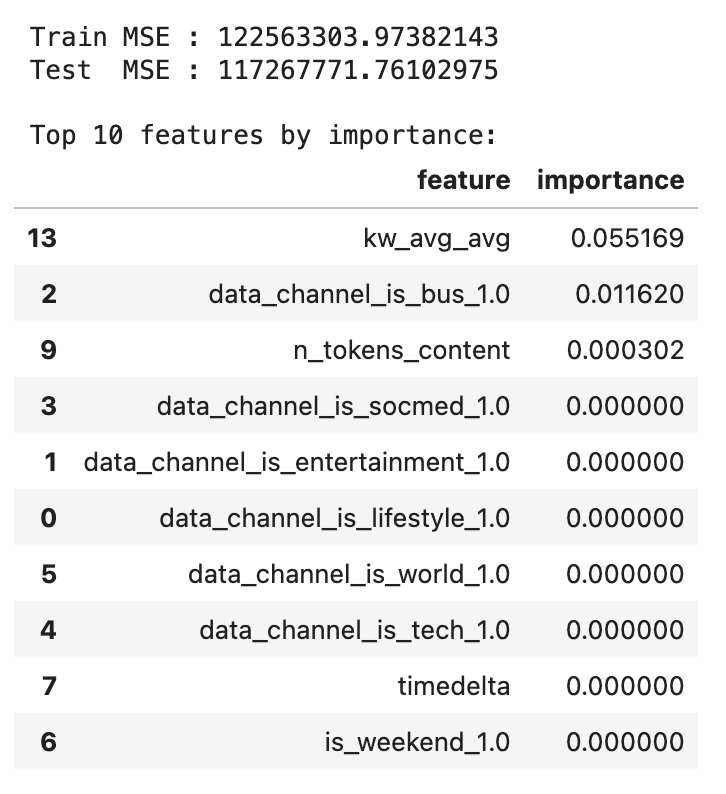
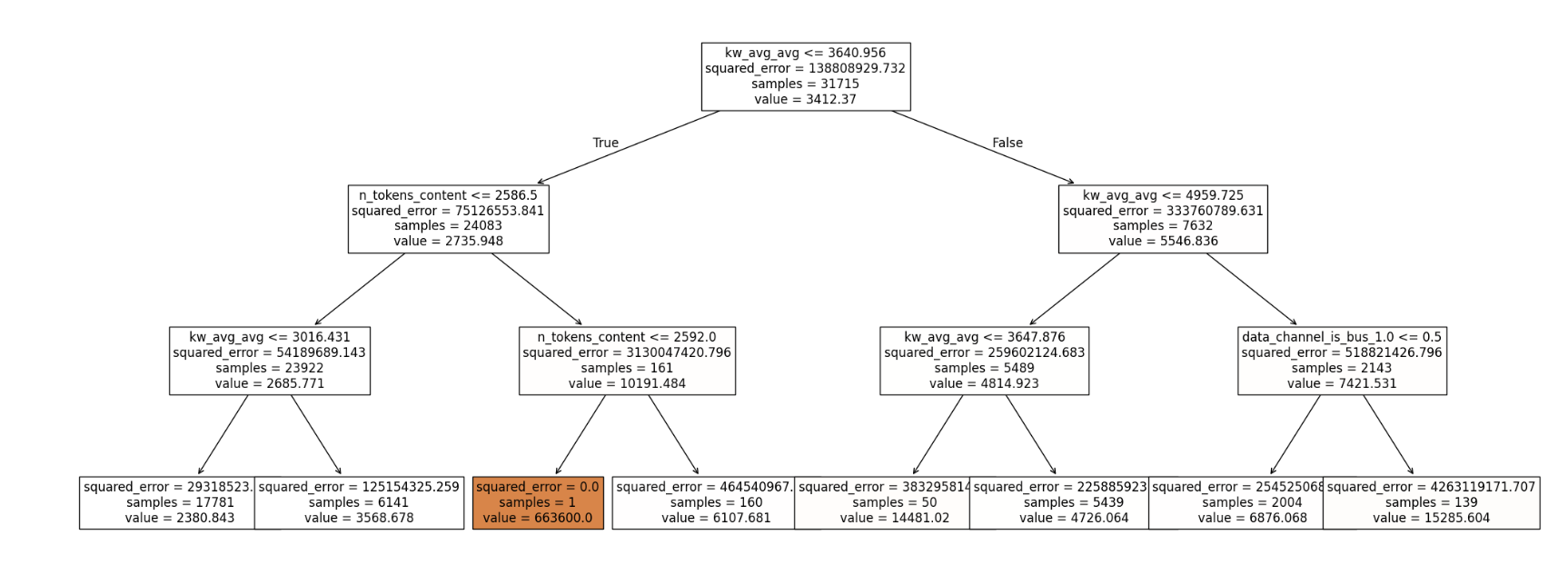
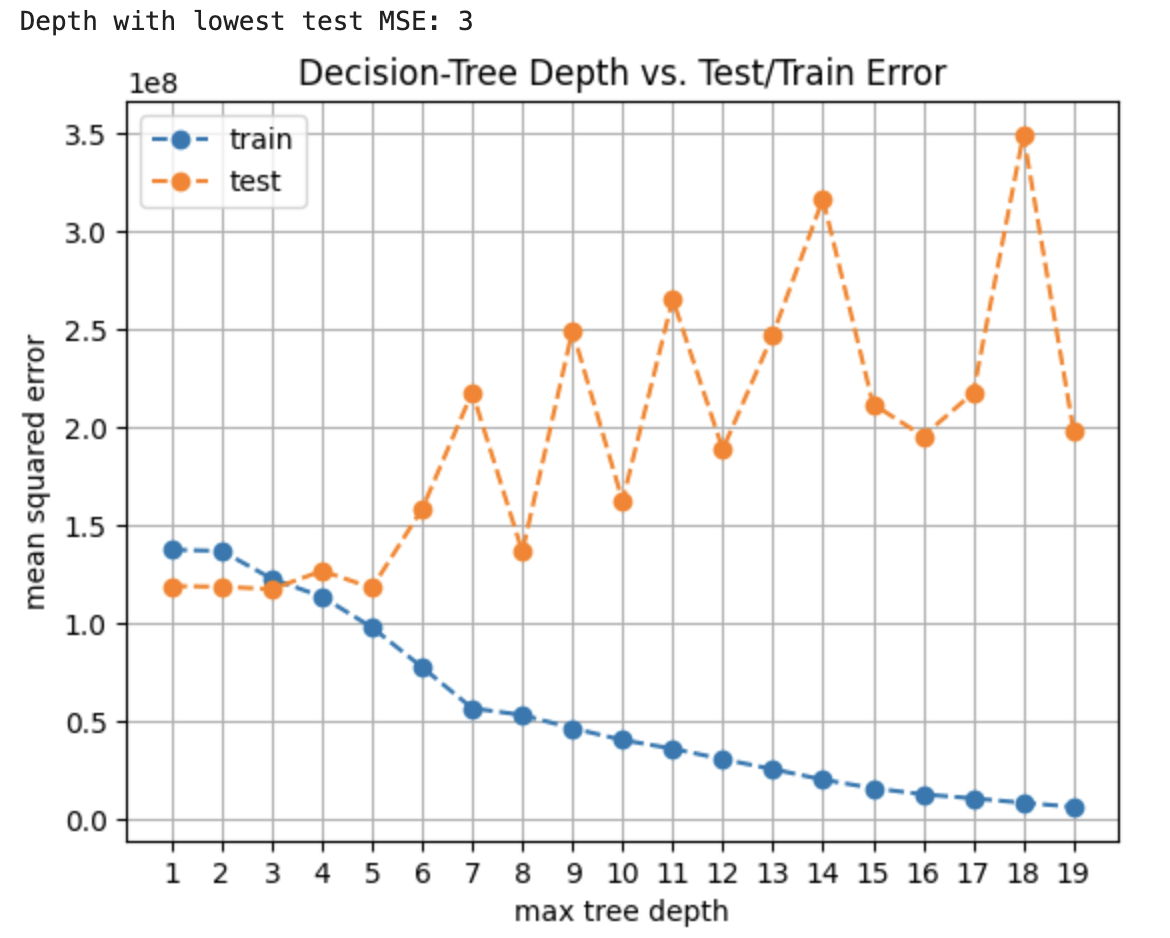
I turned to a k-nearest-neighbors regression because KNN bases each prediction on the outcomes of the most similar observations in feature space. If Facebook-share counts are driven by localized “clusters” of articles—say, pieces that share the same topic flag, weekend timing, and sentiment mix—then a distance-based method can exploit those neighborhood patterns more effectively than a single global equation.



The tuned K-nearest-neighbors regressor (k = 50) edges past the naïve baseline but falls short of the multiple-regression pipeline: it trims test-set MSE from the baseline’s ≈ 120.7 million to ≈ 120.6 million, yet the linear model still leads at ≈ 118.7 million. The train error (≈ 133.3 million) sits well above the test error, indicating that this high-k configuration smooths the data heavily—reducing variance but leaving some signal untapped. In practice KNN can pick up local, non-linear pockets of similarity, but with such a wide feature space and highly skewed target, neighbourhood averages converge toward the global mean, limiting the gain. Grid-searching k helped avoid over-fitting, yet the model’s best neighbours essentially delivered only a marginal improvement over guessing the mean. Feature-permutation results mirror earlier findings: weekend publication, sentiment polarity, and headline or image richness exert the greatest influence on a neighbour’s prediction, whereas keyword history and the subtler topic flags matter less in this distance-based setting.

#### Decision Tree Regression Model

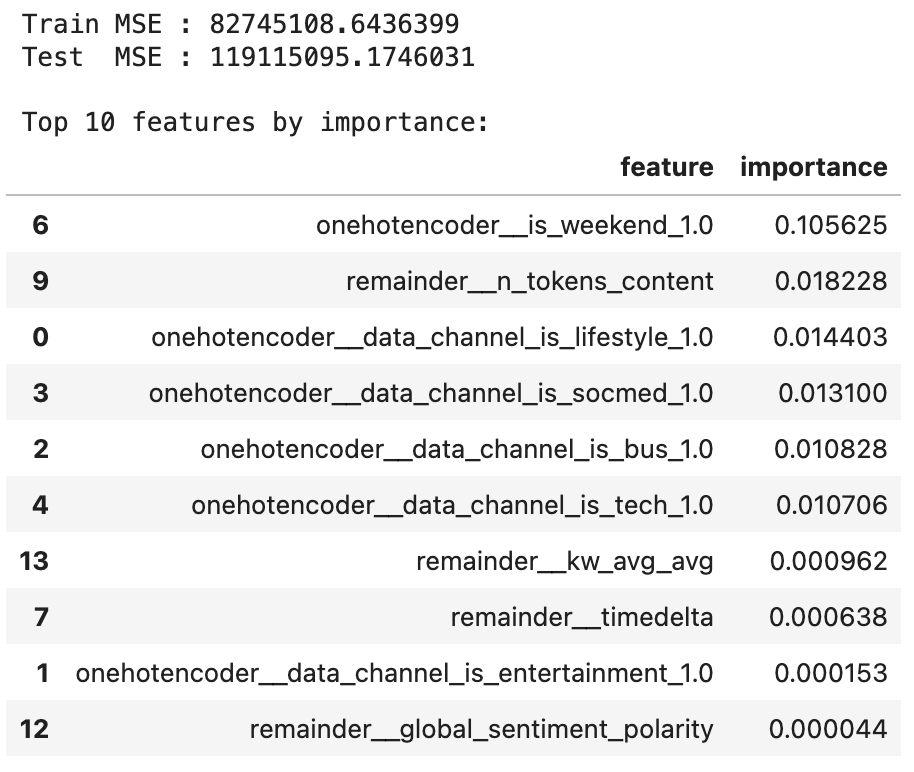
I added a decision-tree regressor to the lineup because, much like K-nearest-neighbors, trees are able to capture complex, non-linear interactions between article features and Facebook-share counts. At the same time, their hierarchical splits provide an intuitive, rule-based picture of how specific feature thresholds—such as keyword performance or article length—drive the model’s predictions, offering clear insight into what makes a story go viral.



The decision-tree model with a max depth of 3 edges past every previous approach on the hold-out set, trimming test-set MSE to roughly **117 million**—better than the baseline (≈ 120.7 M), the multiple regression (≈ 118.7 M), and the high-k KNN (≈ 120.6 M). Its training error (≈ 122.6 M) is actually *higher* than the test error, a sign that the shallow tree is under- rather than over-fitting: three splits simply aren’t expressive enough to capture the full variance in share counts, but they do provide a robust, low-variance estimate that generalises slightly better than the linear model.

The depth-search plot confirms that deeper trees would have memorised the training data (train MSE plummets toward zero) while the test error climbs sharply—a classic over-fitting curve—so depth 3 is the sweet spot before variance explodes. With so few levels, the tree bases its predictions almost entirely on three cues: **kw\_avg\_avg** (historical keyword performance) at the root, followed by article length (n\_tokens\_content) and a single topic flag (data\_channel\_is\_bus). Permutation importance echoes this, showing kw\_avg\_avg as by far the most influential split, a modest contribution from the business-channel flag, and minimal incremental value from sentiment, timing, or weekend publication; their importance scores round to zero because they never appear in the shallow tree’s path. In short, the constrained tree capitalises on one very strong predictor and a couple of supporting splits to achieve the lowest test error so far, but its simplicity also means it leaves many potentially useful signals untapped.

#### Random Forest Regression Model



The tuned random-forest ensemble offers a modest advance over the simpler tree-based and distance-based baselines, trimming the hold-out MSE to ≈ 119 million while retaining a comfortable gap to its own training error (≈ 83 million). Although its test error is only a hair lower than the naïve mean-prediction benchmark, the forest still generalises better than the single decision-tree and K-nearest-neighbour runs, confirming that averaging many shallow trees stabilises variance without the heavy bias of a depth-3 trunk.

Permutation importance reveals that the weekend flag is by far the forest’s dominant signal, followed by article length (n\_tokens\_content) and topic cues—lifestyle, social-media, business, and tech all matter, whereas keyword history and sentiment polarity contribute only marginally. In other words, the ensemble learns that when and what you publish (weekend lifestyle/social-media pieces of moderate length) outweighs subtler lexical nuances in predicting Facebook reach

## Next Steps & Discussion

#### Summary of Findings

In my analysis of Facebook‐share counts for Mashable articles, every predictive model out-performed the naïve baseline, confirming that story-level features hold meaningful virality signal. Ranked by lowest test-set mean-squared-error (MSE), model performance was: Decision-Tree Regression (max-depth = 3), Multiple Linear Regression, Random-Forest Regression, and K-Nearest-Neighbours Regression.

**Key findings**

The shallow decision tree delivered the best predictive accuracy, trimming test MSE to ≈ 117 million—about a 3 % gain over the baseline. Its ability to capture a handful of high-impact, non-linear splits (keyword strength → article length → topic flag) proved more effective than either the linear pipeline or the heavier forest and KNN approaches.

**Most impactful features**

Weekend publication was the single strongest predictor across models, lifting median shares ~35 %.

Keyword performance (kw\_avg\_avg) and article length (≈ 600 tokens) consistently sat near the top of feature-importance tables, highlighting that proven vocabulary and concise copy drive reach.

Topic flags for lifestyle and social-media channels also carried notable weight, especially when paired with weekend timing.

**Secondary variables** Image count, sentiment polarity, and recency (timedelta) showed smaller but still positive contributions. Hard-news and world-channel flags, by contrast, tended to suppress share count.

In conclusion, the decision tree’s few, interpretable rules (publish lifestyle/social-media pieces on weekends, mid-length copy, well-tested keywords) captured enough of the data’s non-linear structure to edge out more complex ensembles. These insights underscore when and how editors should release content to maximise viral potential, and they suggest that next-step models could push performance further while retaining interpretability.

#### Next Steps/Improvements

### Next steps / improvements

To sharpen the models’ predictive power and deepen our understanding of what makes a Mashable article go viral, I would incorporate the following extensions:

* **Richer textual signals** *Apply transformer-based embeddings or TF-IDF features on full headlines and first-paragraph snippets* to capture nuance beyond token counts or sentiment scalars. Contextual language models (e.g., BERT) could detect topical buzzwords and stylistic tone that current numeric proxies miss.
* **Real-time social context** *Augment each article with the Facebook, Twitter, and Reddit activity in the first 60 minutes after publication.* Early traction is strongly predictive of final reach; feeding this “velocity” feature into a time-aware model would improve forecasts for stories still gathering momentum.
* **Author-level history** *Attach rolling averages of past-30-day shares for each author or editor.* Certain bylines consistently outperform; capturing that reputation effect could explain variance not attributable to topic or timing alone.
* **Publication slot granularity** The current weekend flag is coarse. *Encode exact publish hour and day-of-week,* then use cyclic transforms (sine/cosine) so tree-boosting methods can learn fine-grained temporal sweet spots.
* **Gradient-boosted trees** Replace the single tree and vanilla forest with *LightGBM or XGBoost* using early-stopping. These ensembles capture higher-order interactions with fewer trees, typically yielding 10–20 % lower error on skewed targets like ours.
* **Target transformation & probabilistic metrics** Model log₁₊ₓ(shares) or convert to “viral / not viral” classification (e.g., >9 500 shares = top 10 %). Evaluating with MAE-log or ROC-AUC would stabilise variance and produce confidence intervals useful to editors.

Incorporating these additional text, context, and author signals—and upgrading the modelling toolkit—should cut test error substantially while delivering actionable insights on optimal publish timing, headline framing, and team deployment.