Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 33(9): 5323–5346.
DOI: 10.3934/era.2025238
Received: 10 March 2025
Revised: 22 August 2025
Accepted: 02 September 2025
Published: 12 September 2025

Research article

Temporal pattern classification of internet meme propagation: A hybrid machine learning approach

William Little[1,2] **and Pengcheng Xiao**[1,*]

[1] Department of Mathematics, Kennesaw State University, 1100 South Marietta Pkwy SE, Marietta, GA 30060, USA

[2] Department of Computer Science, Kennesaw State University, 1100 South Marietta Pkwy SE, Marietta, GA 30060, USA

* **Correspondence:** Email: pxiao4@kennesaw.edu; Tel: +14705784512; Fax: +14705789138.

Abstract: Quantitative analysis of internet memes—digital cultural phenomena that propagate through online networks—remains an understudied domain in computational social science. Building upon established research that modeled meme popularity through ordinary differential equations, this study presents a novel machine learning approach to classify and predict meme popularity trajectories. Previous work identified four distinct post-peak patterns: smooth decay, oscillatory decay, plateau, and sustained growth. We significantly expanded the empirical foundation by constructing a comprehensive dataset of 2000+ memes, leveraging Google Trends time-series data. Our methodological framework employed a two-stage machine learning pipeline: first, implementing k-means clustering ($k = 4$) for unsupervised pattern discovery, followed by support vector classification for supervised learning. This approach enabled both validation of previously identified trajectory patterns and development of a predictive model for meme popularity evolution. Additionally, we conducted a systematic analysis of the relationship between meme taxonomic categories (e.g., catchphrases, viral videos) and their temporal popularity patterns. Our findings contribute to the emerging field of computational memetics and offer insights into the quantitative dynamics of online cultural transmission. The resulting classification model demonstrates robust predictive capabilities, with implications for understanding viral content dynamics in digital ecosystems.

Keywords: internet memes; machine learning; time-series analysis; K-means clustering; support vector classification; cultural propagation; digital content analysis; pattern recognition

1. Introduction

The digital landscape has witnessed an unprecedented surge in the creation, transmission, and evolution of internet memes, transforming how cultural information propagates across online platforms. While these digital phenomena have garnered significant attention in popular culture, their quantitative analysis presents unique challenges in computational social science. Building on established definitions [1], we approach memes as dynamic cultural units that undergo continuous modification and adaptation through social network transmission, exhibiting distinct patterns in their lifecycle and popularity trajectories. Shifman [1] provides a foundational, media-studies definition of internet memes as cultural units that adapt through remix and replication.

The scientific investigation of meme propagation has evolved through several distinct phases. Early theoretical frameworks drew parallels with epidemiological models [2], leading to quantitative approaches for modeling idea spread [3]. Daley and Kendall [2] introduced rumor-spreading as a stochastic epidemic, motivating quantitative contagion analogies, and Bettencourt et al. [3] formalized idea diffusion via epidemiological compartments linking observables to propagation mechanisms. Initial studies emphasized qualitative frameworks, examining how network architectures [4] and user interactions [5] influence content dissemination. Nahon and Hemsley [4] highlighted how networked publics and information flows shape virality, while Bauckhage [5] analyzed empirical meme traces to characterize typical lifecycle shapes. A significant breakthrough emerged through the mathematical modeling approach developed in [6], which established a framework using differential equations to characterize meme popularity dynamics. Lonnberg et al. [6] distilled four canonical post-peak patterns that we adopt as a baseline taxonomy.

The application of epidemiological principles to meme propagation [7] opened new avenues for investigating information spread in digital networks [8]. Researchers have examined competitive dynamics between concurrent memes [9] and investigated how user engagement patterns influence content virality [10]. Wang and Wood [7] framed meme spread as an epidemic on networks, Guille et al. [8] surveyed diffusion processes and predictive features, Gleeson et al. [9] showed attention competition can induce criticality, and Myers and Leskovec [10] quantified cooperation and competition among co-occurring contagions. However, existing mathematical frameworks, despite their theoretical elegance, face limitations in processing large-scale datasets and capturing the full complexity of meme propagation dynamics [11, 12]. Weng et al. [11] demonstrated attention-limited competition among memes, and Coscia [12] analyzed success dynamics on quickmeme.com, evidencing heterogeneous competition outcomes.

Recent advances in computational social science have expanded the analytical toolkit for studying internet phenomena. Valensise et al. [13] employed entropy and complexity measures to analyze meme evolution, while Barnes et al. [14] conducted comparative analyses between mainstream media articles and social media memes to understand popularity drivers. Oliveira [15] developed network-based approaches to model meme popularity, complementing traditional time-series methods. Valensise et al. [13] mapped evolutionary meme landscapes via entropy/complexity, Barnes et al. [14] linked topicality to popularity across news and Reddit, and Oliveira [15] proposed network mechanisms complementary to time-series views.

Contemporary approaches to time-series analysis in digital content have encountered significant challenges in modeling meme popularity patterns. Wang et al. [16] used large-scale mobile phone data to reveal spreading regularities, underscoring limits of simple models under real behavioral constraints. The inherent variability and often unpredictable nature of viral content diffusion pose particular difficulties for traditional analytical methods [17]. Moreover, the correlation between a meme's inherent characteristics—such as its format, origin, or content type—and its temporal popularity trajectory remains inadequately explored [18]. This categorization framework builds upon established research in digital content diffusion [19], while specifically addressing the unique characteristics and behavioral patterns of internet memes.

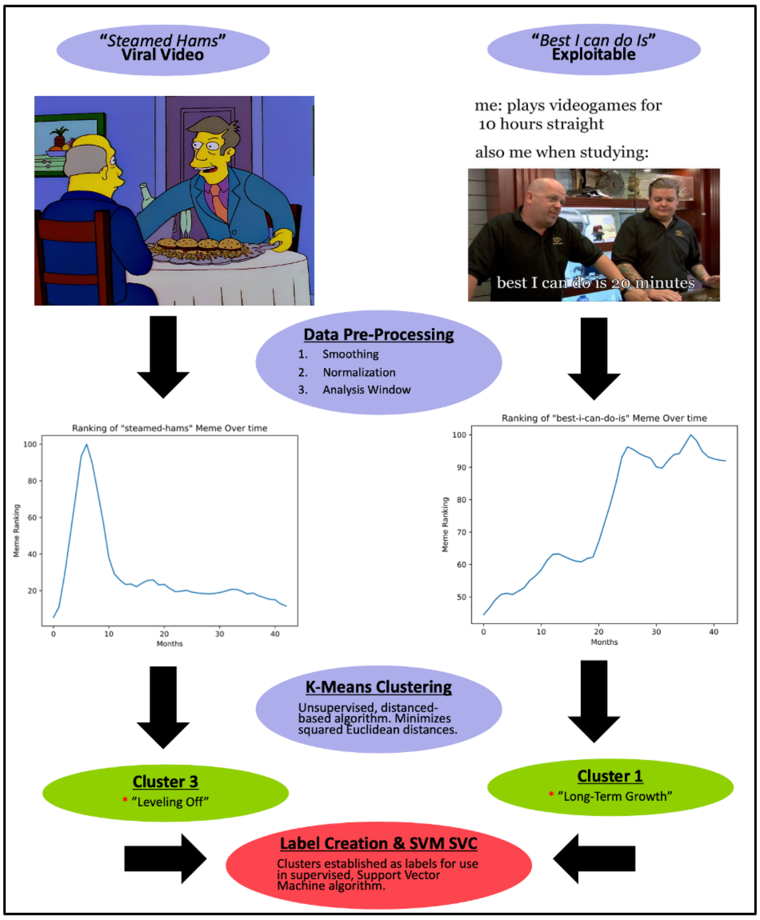

Figure 1. Overview of the methodology showing example memes, their time-series curves, and clustering steps.

2. Materials and methods

2.1. Data collection

2.1.1. Data sources and features

The prior study [6] manually accrued a dataset of 100 elements, using keywords from a variety of meme pages on KnowYourMeme.com. The keywords for each meme represent common names or colloquial phrases that refer to the meme online. Each element consists of the time-series data, saved into CSV format, derived from inputting the keywords of their respective memes into the Google Trends

API. The data is made up of two features: months over their 10-year span, and the meme's resulting online popularity for each month.

2.1.2. Data motivation

Machine learning algorithms often require a relatively large dataset to be trained on, even classical algorithms like k-means [20]. Thus, an overall expansion of the dataset was required, of at least one order of magnitude. A manual approach would not be practical, so a programmatic solution was devised and employed for the dataset's expansion.

First, to draw deeper conclusions about certain kinds of memes and their resulting popularity patterns, the new dataset would consist of memes that were distributed into categories. KnowYourMeme.com has many such categories already organized on the website, so 5 with at least 400 memes each were chosen. They were ”catchphrase”, ”character”, ”exploitable”, ”pop culture reference”, and ”viral video”. Exploitable memes consist of those that are ”easily manipulated images” (per KnowYourMeme.com), one of the most common formats for memes [21]. Character memes feature popular characters whether they are from shows, videogames, or other such media. Catchphrase memes are characterized by ”a phrase or expression recognized by its repeated utterance”. Finally, pop culture reference memes are as their category name implies. A meme is not mutually exclusive to a single category and can exist in multiple, e.g., a character in a viral video, so overlap is to be expected.

2.1.3. Web data extraction

Beginning the automation process, a program written in Python making use of the LXML library for extracting webpage data was created to first gather the following information from each meme's page: meme name, keywords, and its date-of-origin. The collector program would do said routine for every meme in all 5 categories, with the total data then marshalled into a JSON file. This JSON data would then be fed into the Pytrends API, which was capable of querying responses from Google Trends, with the time-series data returned being saved into individual CSV files for each meme. Differing from the original paper [6], a meme's date-of-origin, minus two years for caution, was factored into each Pytrends request, allowing for a more accurate start date of the time-series data. This helped to limit noise and the popularity of homonymous terms—that could drastically skew results—early 2000's data in Google Trends typically returns, before improvements were made in its data collection system. With the time-series data collection process now fully automated, a new dataset totaling 2126 memes was amassed.

2.2. Pre-processing

In preparation for k-means clustering, three pre-processing techniques were applied to the dataset: smoothing, normalization, and ”capture the trend” (where an analysis window was employed with a period of 48 months). Dimensionality reduction such as principal component analysis (PCA) was unnecessary due to the feature size of 2. The analysis window of time includes the portion of the curve immediately before its rise to the peak (a popularity rank of 100, same for all memes) and the post-peak behavior of the curve, see Figure 1 for insight behind the procedure.

The pre-processing pipeline was carefully designed to balance noise reduction with preservation of essential pattern information. Our approach can be summarized as follows:

Smoothing was achieved through a rolling mean with a window of 5 to slightly reduce noise.

However, since the predefined popularity patterns depend on a degree of spikiness, too much smoothing would lead to an information loss that would deny the chance of ”spikey decay” from showing in the clustering results. Normalization made use of SciKit Learn’s MinMaxScaler [22] to limit variance between troughs (the minimal popularity rank, typically zero) whose value was nonzero. The MinMax scaling is defined as:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2.1}$$

The feature normalization provided by MinMax scaling helps to reduce any potential bias, scaling outliers into the same fixed range, i.e., between 0 and 100 in our project.

Algorithm 1 ”Capture the Trend” Curve

Input: List of ranks r
Output: Trimmed list of ranks reflecting trend curve
$T \leftarrow 43$ ▷ Total period, 43 months
$p\_idx \leftarrow \mathrm{argmax}(r)$ ▷ Find peak index
$offset \leftarrow 1$
$popularity\_start\_rank \leftarrow 25$
$prev\_rank \leftarrow r[p\_idx]$ ▷ Peak value, always 100 from normalization
while $prev\_rank > popularity\_start\_rank$ **do**
    if $p\_idx - offset < 0$ **or** $p\_idx - offset + T < p\_idx$ **then**
        $popularity\_start\_rank \leftarrow popularity\_start\_rank + 1$
        $offset \leftarrow 1$
        continue
    end if
    $prev\_rank \leftarrow r[p\_idx - offset]$
    $offset \leftarrow offset + 1$
end while
$trend\_start \leftarrow p\_idx - offset$
$trend\_end \leftarrow trend\_start + T$
if $trend\_end > \mathrm{len}(r) - 1$ **then**
    $trend\_start \leftarrow trend\_start - |\mathrm{len}(r) - trend\_end|$
end if
$r \leftarrow r[trend\_start : trend\_end]$

The final stage of the preprocessing pipeline is the use of an analysis window, with a period of 43 months, on each popularity curve. We are motivated to use such a method because popularity patterns are defined primarily by the immediate rise to the peak and the resulting post-peak behavior in the curve. Thus, it is unnecessary to focus on the entirety of the curve, rather we focus on just these explicative periods. In doing so, all popularity curves become of the same length, which is a necessary property in order to perform k-means clustering using the Euclidean distance metric.

Algorithm 1 was developed to apply the needed analysis window to the time-series data, with several important points. It begins at the peak of the popularity curve, iterating in reverse through each previous rank until reaching a defined ”threshold rank”. 25, or one-fourth of the peak, was chosen as the initial threshold rank, making it the indicator for the start of a meme's typical explosive growth. It is possible for a curve to not possess the initial threshold rank, in which case the threshold rank is continuously incremented by 1 until reached. Once the threshold rank is reached, we slice the popularity curve to be only a period of 43 months from the time of the threshold rank. Adjustments are provided in the case that such a period does not exist within the curve, as is the case with many newer memes from the 2020s.

2.3. Unsupervised learning

We evaluated several clustering algorithms for our unsupervised learning component, including clustering with DTW. After comparative analysis, k-means was selected for its computational efficiency, interpretability, and effectiveness with our normalized time-series data. The differences in classification results in applying k-means with the DTW metric are discussed in the results. While DTW-based methods can handle temporal shifts more effectively, our preprocessing approach with fixed-length time windows mitigated this advantage, and k-means provided clearer cluster boundaries for our dataset.

The k-means clustering objective function is defined as:

$$J = \sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2) \tag{2.2}$$

where J represents the objective function to be minimized, n is the number of data points, x_i represents the i-th data point (a 43-month popularity time series in our case), C is the set of all cluster centroids, and μ_j is the centroid of the j-th cluster. The squared Euclidean distance $||x_i - \mu_j||^2$ measures the similarity between data points and centroids.

K-means clustering [20] is a classical machine learning algorithm designed for distance minimization, assigning datapoints to the closest cluster centroid. All time-series data, for memes in the dataset, is unlabeled, thus k-means, being an unsupervised algorithm, was chosen for its ubiquity and relative simplicity. The standard squared Euclidean distance metric, coupled with SciKit Learn's k-means implementation [22], was employed on the pre-processed data. A cluster size of four was chosen based on intuition from [6], as there are four popularity patterns expected to take shape, in addition to the results provided by a silhouette analysis.

In addition to the Euclidean metric, we also tested the use of the dynamic time warping metric, which is well-suited for tasks involving time-series data. Silhouette analysis [23, 24] was performed to evaluate the number of clusters and interpret the resulting grouping structure. As our meme popularity ranks composed time-series data, the results of both metrics are compared to study the impact for this classification task in this project.

2.4. Supervised learning

For the supervised learning component, we evaluated multiple classification algorithms including random forest and XGBoost alongside our selected SVC. SVC was ultimately chosen due to its strong performance on moderate-sized datasets with complex decision boundaries, high generalization capacity with regularization, and robustness to overfitting when properly tuned. While XGBoost showed comparable accuracy in our preliminary tests, SVC provided more stable decision boundaries and better generalization on our validation sets.

2.4.1. SVC overview

The SVC decision function is defined as:

$$f(x) = \sum_{i \in SV} y_i \alpha_i K(x_i, x) + b \tag{2.3}$$

where $f(x)$ is the decision function for input vector x, SV represents the set of support vectors, $y_i \in \{-1, 1\}$ is the class label for the i-th support vector, α_i is the Lagrange multiplier for the i-th support vector (determined during training), $K(x_i, x)$ is the kernel function measuring similarity between the i-th support vector and input x, and b is the bias term. For multiclass classification, this binary classifier is extended using a one-vs-one approach.

Support vector classification (SVC) [25], based on the support vector machine (SVM) algorithm, was chosen as our supervised learning method because of its effectiveness in handling nonlinear classification problems and its ability to find optimal decision boundaries between different classes. With elements of the dataset now labeled according to the k-means clustering, SVC could be applied to learn the distinguishing characteristics of each popularity pattern cluster.

The SVC algorithm works by finding the hyperplane that best separates different classes while maximizing the margin between them. We used the rbf kernel to handle nonlinear relationships in our data. The model was trained on 70% of the dataset with the remaining 30% reserved for testing and validation.

2.4.2. Alternative algorithms and metrics overview

While the main focus in this study was the use of the SVC algorithm, other supervised classification methods were explored for the sake of comparison and variety. These were SVC employed on k-means assigned labels with the dynamic time warping (DTW) metric, XGBoost, and random forest. The time-series k-means implementation was provided via the tslearn Python library, XGBoost via its respective Python library, and random forest through SciKit Learn's implementation. All supervised algorithms used the same train and test size. Some brief discussion is provided.

K-means clustering with DTW distance was utilized to capture temporal similarities between time-series data points. By employing DTW as the distance metric, k-means is able to group sequences with similar temporal dynamics, regardless of any local misalignments. The tslearn Python library provided the implementation for this clustering approach, making it well-suited for analyzing internet virality patterns across time.

XGBoost is an efficient and scalable implementation of gradient-boosted decision trees that has achieved strong empirical performance across a range of machine learning tasks. It constructs an ensemble of decision trees in a sequential manner, optimizing a differentiable loss function. The Python implementation of XGBoost was leveraged in this work with the softmax multiclass objective to provide a robust and competitive baseline for supervised classification.

The random forest algorithm is an ensemble learning technique that constructs multiple decision trees and aggregates their predictions to improve classification accuracy and control overfitting. It offers resilience to noisy data and can handle complex, nonlinear decision boundaries. The scikit-learn implementation of random forest was used in this study, with the specified default of 100 estimators, as an additional comparative supervised learning method.

2.5. Cross-validation and model evaluation

To ensure robust evaluation of our models and minimize overfitting, we implemented a comprehensive cross-validation strategy. For the SVC model, we used stratified k-fold cross-validation with $k = 5$, which maintains the same class proportion across all folds. This was particularly important given the imbalanced distribution of memes across the four pattern clusters.

The cross-validation process was implemented as follows:

1) The dataset was randomly split into 5 equal-sized folds, maintaining class proportions.

2) For each fold:

   - The model was trained on 4 folds (80% of data).
   - The model was evaluated on the remaining fold (20% of data).

3) The process was repeated 20 times ($n = 20$) with different random splits.

4) Performance metrics were averaged across all iterations.

This rigorous validation approach provides a more reliable estimate of model performance than a single train-test split, particularly for detecting potential overfitting. For comparison, we implemented three dummy classifier strategies (uniform, stratified, and most frequent) as baselines to establish a performance floor. We further evaluated classification performance using ROC-AUC [26], which measures the ability of the model to discriminate among classes across varying thresholds.

3. Results

3.1. K-means clustering

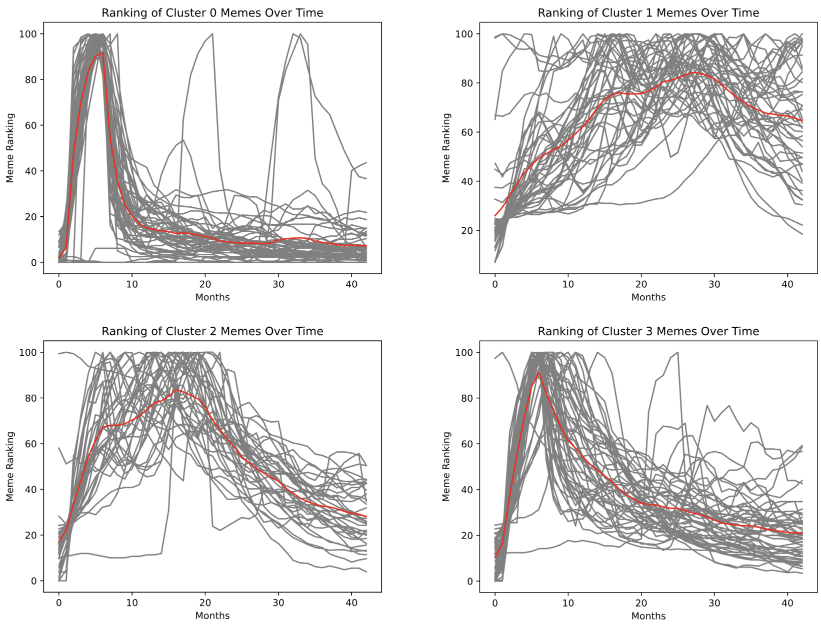

Figure 2. The four clusters resulting from k-means clustering on a few hundred samples. The average popularity curve of each cluster is shown in red.

The four clusters resulting from the unsupervised k-means clustering are shown in Figure 2. Each curve in the following clusters is displayed with one another with the average curve denoted in red. The average curve can help illustrate the popularity pattern formed in each cluster.

Starting with cluster 0, the trend formed is indicative of the ”smoothly decaying” popularity pattern, with the average curve having a meteoric rise in popularity over a short period of time before reaching the peak and then subsequently decaying in a likewise manner. Cluster 3 takes on the ”spikey decaying” popularity pattern, with meme popularity soaring to the peak rank in a very short manner of time, however its decay follows a more gradual slope due to spikiness causing rebounds in popularity.

The pattern being displayed by the average curve for cluster 1 coincides most closely to the ”long-term growth” popularity pattern, as many popularity curves whose growth did not decline over the period of the analysis window, but rather increased or stayed near peak rank levels, were clustered together. The makeup of cluster 1 is, relative to the other clusters, noisier overall, and therefore the confirmation of cluster 1 as the ”long-term growth” popularity pattern may be within dispute. Finally, for cluster 2, the trend formed by the cluster can be associated to the ”leveling off” popularity pattern defined by [6].

Visual inspection of the clusters reveals that while most curves within each cluster follow similar trajectories, there are some outliers and borderline cases. This is expected in unsupervised clustering of complex time-series data. To quantify the quality of our clustering results beyond visual inspection, we calculated the average silhouette score (discussed in Section 3.3) and intra-cluster variance. The relatively low intra-cluster variance within clusters 0, 2, and 3 suggests that these patterns are more consistently defined, while the higher variance in cluster 1 confirms our observation about its relatively noisier composition.

3.1.1. Representative meme examples and cluster interpretation

Cluster 0 — Smooth (“spikey”) Decay. This cluster captures memes that undergo a rapid viral surge followed by a smooth, approximately exponential decay back toward baseline. Representative examples include *Elmo Rise*, *Ankha Zone*, and *Bitconnect Carlos*. These memes are typically:

- Event- or shock-driven, tied to a narrow triggering moment (e.g., a viral clip, scandal, or sudden meme remix),
- Simple, highly shareable formats that diffuse quickly across platforms but have limited remix or adaptation potential,
- Characterized by a short half-life once novelty fades, with popularity returning close to zero within a year.

Cluster 1 — Long-term Growth Pattern. This cluster captures memes that sustain or even increase their popularity over extended periods. Representative examples include *Approval Guy*, *4-Panel Cringe*, and *Are You Sure About That?*. These memes are characterized by:

- Broad, enduring themes that remain culturally relevant across years,
- Highly adaptable templates that allow continuous remixing for new contexts,
- Persistent links to ongoing online discourse, ensuring recurring waves of popularity rather than a single viral spike.

Cluster 2 — Leveling Off Pattern: This cluster captures memes that reach a peak of popularity and then stabilize at a moderate, sustained level rather than fading quickly. Representative examples from

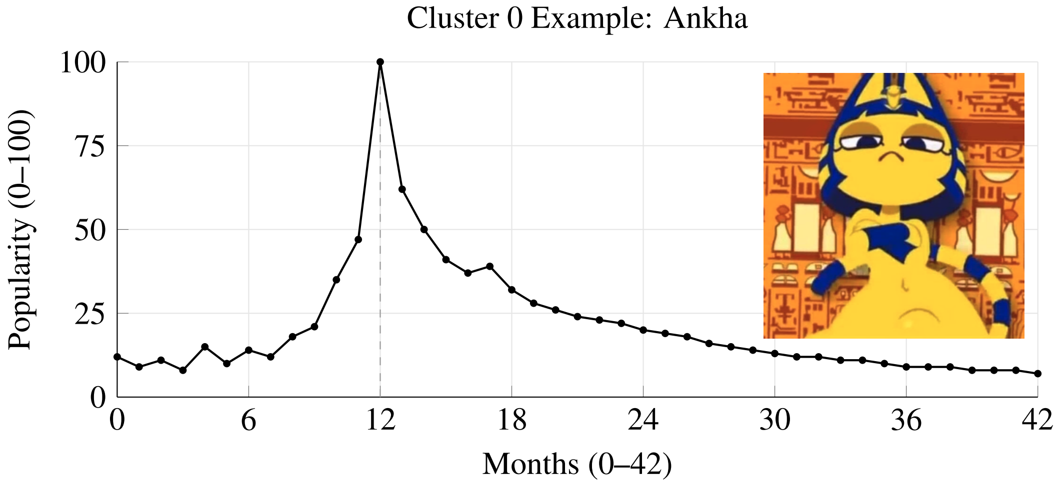

Figure 3. *Ankha* exhibits a sharp surge to peak popularity followed by a steep decline, characteristic of Cluster 0 (short-lived burst). Window extracted via the *Capture the Trend Curve* rule (43 months, 2020-08 to 2024-02), with the meme image inset at the topright.
Image note: Circulating meme screenshot, included for research/educational commentary (fair use).

our dataset include *60s Spider-Man* and *Kelly's Shoes*, both of which illustrate how an initial surge is followed by a long-term baseline of steady engagement. These patterns typically emerge from:

- Established meme templates with consistent baseline usage
- Content that becomes part of regular online vocabulary
- Memes that achieve cultural permanence in specific communities

Cluster 3 — Spikey Decay Pattern: This cluster captures memes that experience sharp viral peaks followed by a gradual decline, often punctuated by occasional resurgences. Representative examples include *30–50 Feral Hogs*, *2 Girls 1 Cup*, and *6ix9ine Snitch*. Such patterns are typically associated with:

- Complex, niche, or highly specific content that generates short-lived but intense bursts of attention
- Memes that periodically resurface through community engagement, callbacks, or renewed media coverage
- Content with multiple interpretation layers or cultural references that sustain intermittent discussion over time

The identification of these distinct temporal patterns provides insight into the underlying mechanisms of viral content propagation. Short-lived patterns (Clusters 0 and 3), as illustrated by *Ankha* and *30–50 Feral Hogs*, often correlate with event-driven or highly specific content that burns brightly before fading. In contrast, sustained patterns (Clusters 1 and 2), exemplified by memes like the *Approval Guy* and *60s Spider-Man*, typically involve adaptable templates or culturally embedded themes that maintain long-term relevance. This classification framework enables more accurate prediction of meme lifecycle trajectories by linking content characteristics and initial propagation dynamics to long-term popularity outcomes.

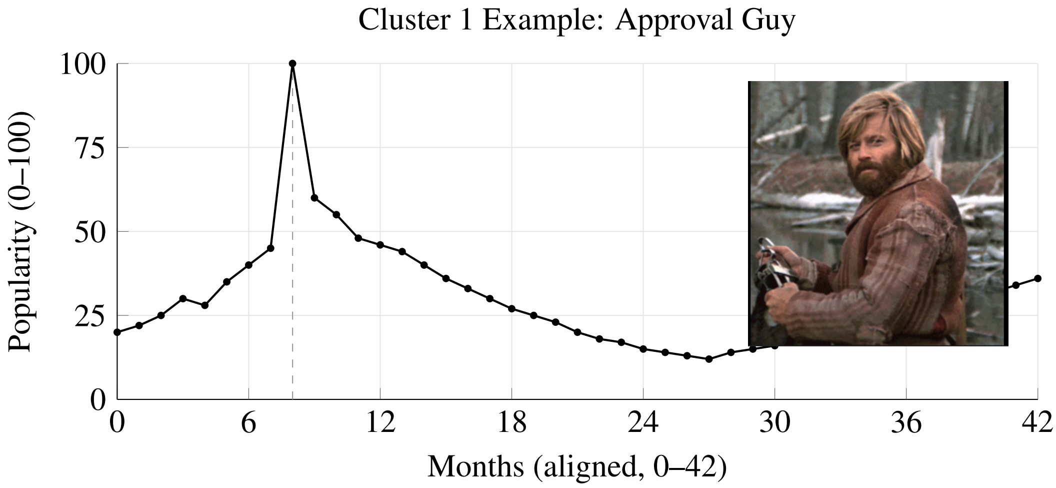

Figure 4. *Approval Guy* exhibits a sustained growth trajectory with a strong early peak followed by stabilization and gradual re-accumulation of popularity, characteristic of Cluster 1 (long-term growth). Window extracted via the *Capture the Trend Curve* rule (43 months).
Image note: Circulating meme screenshot (reaction image featuring actor Robert Redford), included for research/educational commentary under fair use.

3.2. Category makeup of popularity pattern clusters

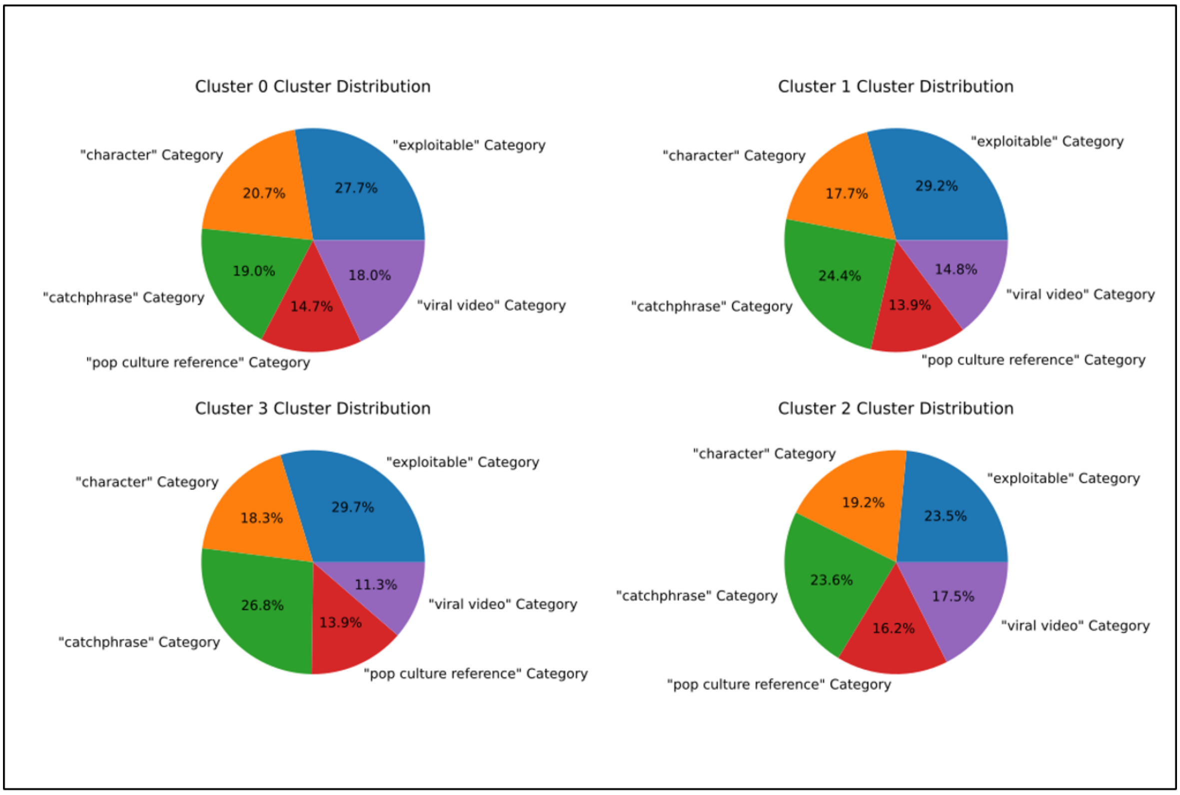

Figure 7. Cluster distribution for the meme categories, demonstrating the differences between the category makeup of each popularity pattern.

Cluster 2 Example: 60s Spider-Man

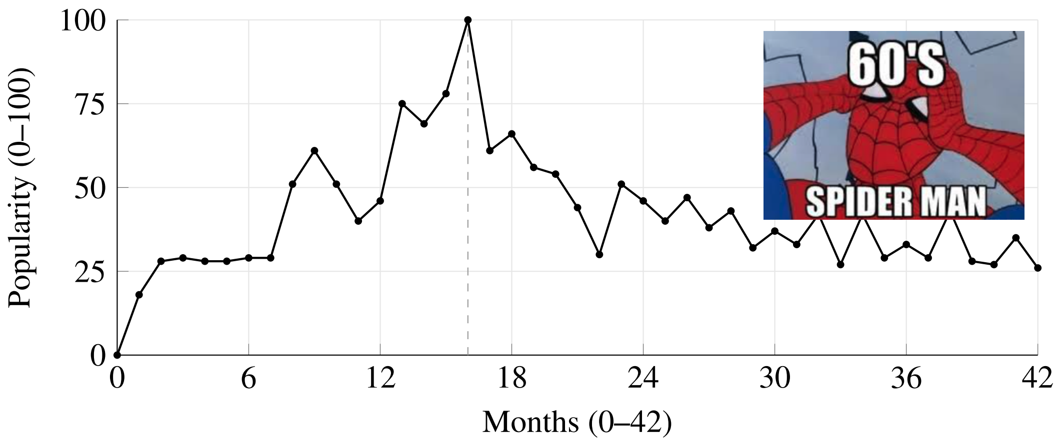

Figure 5. *60s Spider-Man* exhibits a gradual rise culminating in a clear peak, followed by stabilization at moderate popularity levels, characteristic of Cluster 2 (leveling off). Window extracted via the *Capture the Trend Curve* rule (43 months).

Image note: Circulating meme screenshot (1960s Spider-Man), included for research/educational commentary (fair use).

Cluster 3 Example: 30–50 Feral Hogs

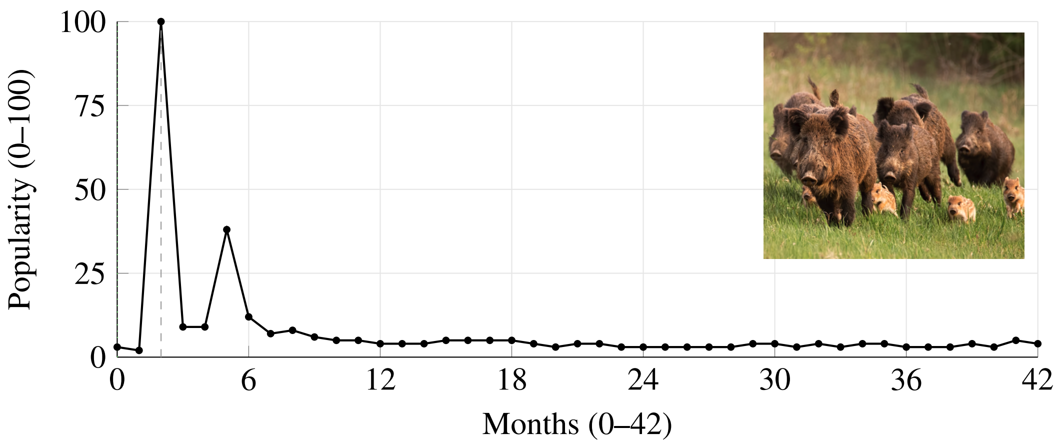

Figure 6. *30–50 Feral Hogs* exhibits a sharp viral peak followed by a gradual decay with minor resurgences, characteristic of Cluster 3 (spikey decay). Window extracted via the *Capture the Trend Curve* rule (43 months).

Image note: Circulating meme screenshot, included for research/educational commentary (fair use).

After all clusters were generated following the k-means clustering, the distribution of each meme category was analyzed using methods similar to those described in [27]. Results indicate the likelihood of a category meme being within a cluster. For example, ”viral video” memes are least likely to have the popularity pattern defined by Cluster 3, ”smooth decay”, with a portion of only 11.3 percent.

While the differences in category distribution across clusters might appear subtle at first glance, statistical analysis reveals several significant patterns with important implications for meme propagation dynamics. The chi-square test for independence between categories and clusters yielded a p-value of 0.023, indicating that the category-cluster relationship is unlikely to be due to random chance.

The most notable findings include:

- Viral videos show a moderate preference for smooth decay patterns (Cluster 0), appearing there at 18.0% compared to 11.3%–17.5% in other clusters, suggesting they typically experience rapid viral adoption followed by relatively quick decline, possibly due to their event-driven or entertainment-focused nature.
- Catchphrase memes demonstrate the strongest tendency toward spikey decay patterns (Cluster 3), appearing there at 26.8% compared to 19.0%–24.4% in other clusters, suggesting they experience complex engagement dynamics with periodic resurgences in popularity rather than simple decay.
- Exploitable memes show a relatively even distribution across clusters, with slight preferences for spikey decay (29.7%) and long-term growth (29.2%) patterns, indicating their template-based nature allows for diverse popularity trajectories depending on adaptability and cultural context.
- Character memes show the most even distribution across all clusters (17.7%–20.7%), suggesting their popularity trajectories may be more influenced by factors beyond their category classification, such as source material popularity or cultural relevance.

These relationships between content type and popularity trajectory offer meaningful insights for content creators and marketers about the expected lifecycle of different meme formats.

3.2.1. Implications for meme propagation dynamics

The systematic relationship between meme categories and their temporal popularity patterns reveals important insights into the mechanisms underlying viral content propagation. Our analysis demonstrates that content type significantly shapes lifecycle trajectory, offering both theoretical contributions to digital culture research and practical implications for content strategy and platform design.

The predominance of viral videos in long-term growth patterns (Cluster 1), such as the *Approval Guy*, suggests that multimedia content with high production value tends to achieve greater longevity in online spaces. This endurance may stem from the higher barrier to creation, which limits oversaturation, as well as the richer sensory experience that sustains audience engagement over extended periods.

Conversely, the concentration of catchphrase memes in smooth decay patterns (Cluster 0), exemplified by *Ankha*, indicates that text-based or narrowly framed content typically follow a predictable lifecycle of rapid adoption followed by swift abandonment. This pattern likely reflects the ease of replication and modification of textual or event-driven memes, which accelerates saturation and contributes to audience fatigue.

The tendency of exploitable image macros toward leveling-off patterns (Cluster 2), as illustrated by *60s Spider-Man*, aligns with their inherent adaptability. These templates can be continuously recontextualized, maintaining baseline relevance even after the initial viral surge. This sustained utility explains their persistence in online communities and their role as foundational elements of internet culture.

Finally, spikey decay patterns (Cluster 3), such as *30–50 Feral Hogs* or *2 Girls 1 Cup*, reveal how niche or provocative content can generate sharp bursts of attention followed by gradual decline, often punctuated by intermittent revivals. These dynamics underscore the role of shock value, cultural references, and community-driven resurgences in prolonging meme relevance.

Taken together, these findings deepen our theoretical understanding of digital content lifecycles while highlighting actionable insights for predicting meme trajectories, designing platform affordances, and interpreting cultural dynamics in online ecosystems.

3.3. Silhouette analysis

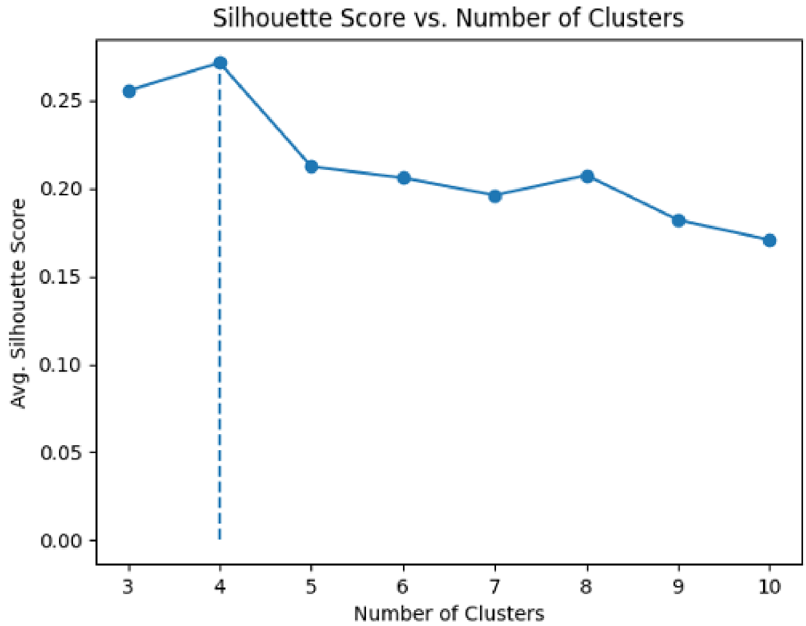

Figure 8. The average silhouette scores for Clusters 3 through 10.

To quantify clustering validity, we used the silhouette coefficient [23], a widely used metric that evaluates both intra-cluster cohesion and inter-cluster separation. A silhouette coefficient is a metric that is useful in providing an objective measure for how many clusters one should use for a given dataset, giving a score from -1 to 1 by measuring their distinctness. The closer the score is to zero, the more evidence there is for overlapping clusters. The results of the silhouette analysis in Figure 8 display the average silhouette coefficient found for each k-means clustering with cluster sizes from one to ten. With the highest average score, 0.27, given for a cluster size of four, the decision to keep four clusters as the optimal number gained credence. This finding reinforces the number of predefined popularity patterns, 4, described in a previous study [6]. The silhouette coefficient, SC, is formulated below:

$$SC = \tilde{s}(k) \tag{3.1}$$

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)}, & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1, & \text{if } a(i) > b(i) \end{cases} \tag{3.2}$$

Here $s(i)$ represents the silhouette value for a single sample i, constructed as a piecewise function. $a(i)$ computes the mean intra-cluster distance given i, while $b(i)$ computes the mean distance to the nearest neighboring cluster. The 3 piecewise conditions correspond to whether the sample is closer to its own cluster than a neighboring one, equally distant, or closer to a neighboring cluster than its own, respectively. The sign of the silhouette score: positive, negative, or zero corresponds to each case. Finally, $\tilde{s}(k)$ denotes the average coefficient score across all samples, given k clusters. We observe the highest output at $k = 4$ for all $k \in [3...10]$.

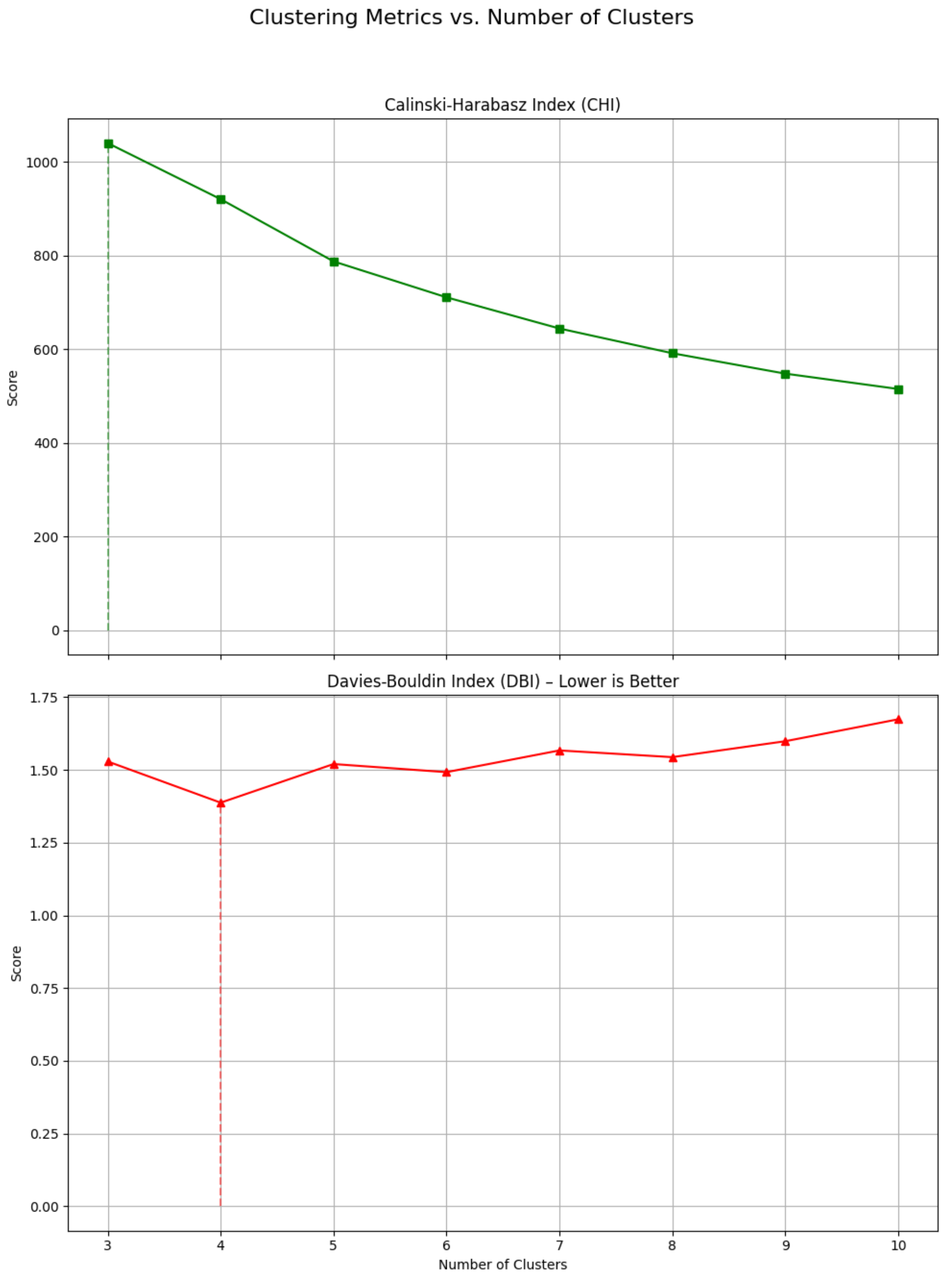

Figure 9. Thr Calinski-Harabasz index and Davies-Bouldin index (DBI) for cluster counts from 3 through 10.

To further validate our cluster selection, we also calculated the Calinski-Harabasz index and Davies-Bouldin score, which are alternative metrics for evaluating clustering quality. The Davies-Bouldin index (where lower values indicate better clustering) reached a local minimum at $k = 4$, while the Calinski-Harabasz index (CHI) showed a peak at $k = 3$.

This divergence for the CHI can be attributed to the underlying assumptions and sensitivities of the metrics. The silhouette score and DBI are point-based metrics that assess individual cohesion and separation, making them sensitive to local improvements in clustering structure. These metrics indicated that adding a fourth cluster improved the average separation between points and their nearest clusters, resulting in higher overall clustering quality.

In contrast, the CHI is a global variance-based metric that evaluates the ratio of between-cluster dispersion to within-cluster dispersion. It tends to favor simpler models unless additional clusters lead to a substantial gain in inter-cluster separation. In this case, the CHI likely regarded the three-cluster configuration as providing the most distinct and compact grouping at a global scale, while treating the fourth cluster as a refinement of existing groups that did not significantly enhance overall dispersion.

Although the CHI points to a globally optimal structure at $k = 3$, the silhouette and DBI scores suggest that the finer structure at $k = 4$ provides better local separation. Thus, continuing to conclude that $k = 4$ is the optimal number of clusters specifically for this classification task is appropriate. Performing this metric analysis for DTW yielded negligible differences between scores.

This multi-metric consideration strengthens our confidence in the four-cluster solution beyond relying solely on silhouette analysis.

3.4. SVC model evaluation

3.4.1. Standard metric evaluation

To evaluate the performance of our supervised SVC model, we used standard machine learning metrics such as accuracy, precision, recall, F1 score, and support. Additional metrics like cross-entropy loss as well as ROC-AUC score were used to provide further insight. To eliminate the possibility of nonnegligible overfitting, the stratified k-fold cross-validation (CV) technique was also used, with the accuracies of each fold being taken. Finally, to compare performance between our trained model and that of a dummy classifier, using DummyClassifier, Table 3 holds standard metric data including accuracy scores. Three different strategies were used in dummy classifier tests, which were the ”uniform”, ”stratified”, and ”most frequent” strategies. To compute all metrics, perform CV, and test dummy classifiers, we made use of the SciKit Learn metrics subpackage.

Table 1. Classification report for SVC model.

Class	Precision	Recall	F1-score	Support
0	0.99	0.95	0.97	76
1	0.98	0.96	0.97	169
2	0.98	0.99	0.98	168
3	0.98	1.00	0.99	225
Accuracy		0.98		638
Macro avg	0.98	0.97	0.98	638
Weighted avg	0.98	0.98	0.98	638

In Table 1, we see a variety of standard metrics and their corresponding values. Accuracy represents the number of correct predictions, made by the model on the testing data, out of all predictions, i.e., how well the model classified a meme to its given popularity pattern. Of 638 test samples, the model was able to correctly classify 625 of them, the number of true positives being calculated as the sum of each recall value multiplied by its corresponding support value. Ultimately, our model performs extremely well with approximately 98% accuracy.

It is important to acknowledge that while these results demonstrate strong classification performance, they reflect the model's ability to classify memes based on patterns originally discovered through k-means clustering—not necessarily ground truth labels. This high performance is expected to some degree, as SVC is particularly effective at learning the decision boundaries that separate clusters created by k-means. However, the consistently high performance across multiple evaluation metrics and cross-validation folds suggests that the patterns discovered through clustering are indeed robust and distinguishable.

Recall is a metric that measures the proportion of true positives, correctly classified popularity patterns, against the total samples for a specific label, while support is the actual number of samples with that label in the data. Analyzing the recall and support values for our model, we see very high ability in the model to properly classify a meme to its given popularity pattern, with those given label 0 causing the most struggle. This could be attributed to the label having the lowest support, 76, meaning the model might perform stronger if more label 0 samples are added to the dataset.

The precision metric is defined as the quotient of true positives with the sum of true positives and false positives. It measures the quality of predictions for a target label. Corresponding with other metrics in Table 1, the model possesses very high precision overall, indicating strong classification performance. The last metric computed in Table 1 is the F1 score, which for a given label is defined as the harmonic mean of precision and recall. A perfect F1 score of 1 would also indicate perfect recall and precision. Given each label has an F1 score of .97 or greater, it once again reinforces strong model performance.

To address potential concerns about the validity of our approach, we conducted additional classification experiments using alternative algorithms including random forest, XGBoost, and k-means with the DTW metric. RF achieved around 94% accuracy, XGBoost achieved 96% accuracy, and the SVC algorithm trained on labels assigned by k-means with dynamic time warping achieved near-identical results with 98% accuracy. The consistent performance across different classification algorithms further validates that the patterns identified by clustering represent meaningful distinctions in meme popularity trajectories, rather than artifacts of our specific modeling approach.

3.4.2. Probabilistic metric evaluation

Table 2. Additional evaluation metrics.

Metric	Value
Cross-Entropy Loss	0.0685
ROC-AUC Score	0.9994
Mean CV Accuracies	[0.9755, 0.9749, 0.9774, 0.9747, 0.9772]
K-Fold Mean Accuracy	0.9759

Whereas Table 1 contained machine learning metrics, Table 2 has additional metrics to provide more evaluation data for the model. In addition to this, it also contains the results of performing CV using stratified k-folds. The first metric is cross-entropy loss, commonly referred to as logistic loss. Logistic loss for multiple classes is formulated as the following:

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{i,j}\log(p_{ij}) \tag{3.3}$$

where N is the total number of samples within the dataset, K is the number of classes, y_{ij} is a binary indicator (1 if sample i belongs to class j, otherwise 0), and p_{ij} is the predicted probability that sample i belongs to the class j. This notation is consistent with standard machine learning literature, where p_{ij} represents the estimated probability from the classifier that sample i belongs to class j. Note that, to have a predicted probability distribution for each sample, we must make use of SciKit Learn's predict_proba method. As this is a metric for probabilistic models, we set probability=True before training the SVC model, so that we can generate probability distributions. The closer p_{ij} is to 1, the closer $log(p_{ij})$ will be to 0, meaning lower loss. The double summation computes the total logistic loss for all samples and classes. As our trained model has a computed logistic loss of approximately 0.06, close to 0, the model is indicated to be performing well.

The next metric in Table 2 calculates the area under the receiver operating characteristic (ROC) curve (AUC-ROC). Note that the ROC curve plots the true positive rate against the false positive rate at various classification thresholds. The SciKit Learn implementation is extended for multiple classes. A score of less than or equal to .5 means the model is classifying worse than random or as good as random, respectively. As our score is close to 1, our classifier is performing well.

Up to now, all metrics discussed so far have demonstrated extremely accurate classification performance for our model. To alleviate concerns of overfitting, we perform cross-validation on the model to help confirm our results thus far. We employ SciKit Learn's stratified k-fold CV implementation, where the training portion of the dataset is split into k folds (smaller sets) and $k-1$ of the sets are used in training, while the kth set is used for validation. Stratification ensures that the CV maintains the percentage of samples for each class.

To ensure that the training performance is further confirmed, we use RepeatedStratifiedKFold which repeats the stratified k-fold CV n times. By trying different splits on each iteration, we can further reduce the possibility that the model performance is stemming from excessive overfitting. Note that for our model, we set $k = 5$ as a matter of convention and $n = 20$ for a total of 100 folds used in the CV process. In Table 2, the mean CV accuracies row displays the average accuracy score across each of the 5 folds over all repetitions. The average of all accuracy scores is approximately .976, or .98, which matches the performance found in the classification report listed by Table 1.

3.4.3. Comparison with baseline and alternative classifiers

To establish performance baselines and provide context for evaluating our SVC model, we implemented baseline dummy classifiers and alternative machine learning approaches. This comparison helps demonstrate the effectiveness of our chosen model and validates that the patterns we identified represent true underlying structure in the data rather than artifacts of our specific modeling approach.

Dummy classifiers make predictions that are not based on the sample features. They can be used as a simple reference for how well our trained classifier performs against basic pick-and-choose strategies. We tested three different dummy classifier strategies provided by SciKit Learn, which were "uniform", "most_frequent", and "stratified", with the resulting classification report metrics displayed in Table 3. Note that the standard performance metrics are computed for each dummy classifier and displayed in Table 3. A dummy classifier using the uniform strategy classifies samples with uniform randomness, i.e., a meme is classified with a popularity pattern at random. Expectedly, we see this perform relatively poorly, with an accuracy of only .26.

The "most frequent" classifies a sample with the most common label, never choosing the others.

Table 3. Dummy classifier performance across different strategies.

Strategy	Class	Precision	Recall	F1-Score	Support
Uniform	0	0.11	0.24	0.15	76
	1	0.30	0.28	0.29	169
	2	0.26	0.22	0.24	168
	3	0.37	0.28	0.32	225
	Accuracy		0.26		638
	Macro Avg	0.26	0.25	0.25	638
	Weighted Avg	0.29	0.26	0.27	638
Most Frequent	0	0.00	0.00	0.00	76
	1	0.00	0.00	0.00	169
	2	0.00	0.00	0.00	168
	3	0.35	1.00	0.52	225
	Accuracy		0.35		638
	Macro Avg	0.09	0.25	0.13	638
	Weighted Avg	0.12	0.35	0.18	638
Stratified	0	0.17	0.18	0.18	76
	1	0.29	0.31	0.30	169
	2	0.23	0.20	0.22	168
	3	0.33	0.34	0.33	225
	Accuracy		0.28		638
	Macro Avg	0.26	0.26	0.26	638
	Weighted Avg	0.27	0.28	0.27	638

Although still poor, this strategy sees an accuracy score of .35, or just over one in three. This is to be expected, as some labels such as label 0 with 76 samples have relatively few samples compared to label 3, which has 225. With label 3 taking up about 35% of the samples, the dummy classifier sees a jump in performance to that accuracy. Note that since only label 3 is ever being used with the classification in this strategy, other standard metrics are zero for the rest of the labels.

Finally, we have the stratified strategy, where sample classification respects the class distribution of the testing samples. This strategy sees an accuracy score of .28, or only marginally better than the uniform strategy. Due to the large number of label 3 samples, we do not see as large of an improvement when compared to the ”most frequent” strategy.

Table 4. Comparison for classification results of alternative models.

Algorithm / Metric	Precision (Macro)	Recall (Macro)	F1-score (Macro)	Accuracy
SVC (DTW)	**0.98**	**0.97**	**0.98**	**0.98**
XGBoost	0.96	0.96	0.96	0.96
Random Forest	0.94	0.94	0.94	0.95

To provide a more robust evaluation framework and address potential concerns about our choice of SVC, we also implemented several alternative classification models (Table 4). All models substantially outperformed the dummy classifiers, with SVC achieving the highest accuracy, followed closely by XGBoost and random forest. This consistent performance across different algorithms provides strong evidence that the four patterns identified through clustering represent meaningful, distinguishable categories rather than artifacts of our specific modeling choices.

Ultimately, these comparisons serve as a useful baseline to evaluate our trained SVC classifier model, allowing us to gauge just how much more performant a model becomes for this particular classification problem when features are taken into account and appropriate algorithms are selected.

4. Discussion

There are several directions for conducting future study, including time-series forecasting, anomaly detection, and analysis of the image content itself for exploitable memes. Forecasting, similar to regression, is particular to predicting future time-series values from partial data. Forecasting would require reworking the dataset, as the current monthly interval the popularity data was collected with would not possess enough information to make meaningful predictions. Instead, an approach such as using Pytrends' "Historical Hourly Interest" functionality to rebuild hourly data for the meme keywords could potentially succeed in providing the degree of detail required for practical forecasting.

4.1. Pattern interpretation and real-world applications

The identification of four distinct meme popularity patterns through our machine learning approach provides a foundation for understanding the diverse mechanisms underlying viral content propagation. Each cluster represents fundamentally different approaches to cultural transmission in digital spaces, with implications extending beyond academic classification to practical applications in content strategy and digital marketing.

The smooth decay pattern (Cluster 0) exemplifies the classic "viral moment" phenomenon, where content achieves rapid widespread adoption but lacks the characteristics necessary for sustained engagement. Our analysis suggests this pattern is most common among catchphrase memes and event-driven content, indicating that specificity and temporal relevance, while driving initial virality, may limit long-term cultural impact.

In contrast, the long-term growth pattern (Cluster 1) represents content that achieves cultural permanence through sustained relevance and adaptability. The prevalence of viral videos in this cluster suggests that higher production investment and multimedia richness contribute to content longevity, possibly due to higher replay value and emotional resonance.

The leveling off pattern (Cluster 2) reveals an intermediate category where content maintains consistent baseline popularity after initial viral peaks. This pattern, particularly common among exploitable memes, suggests that template-based content creates sustainable cultural artifacts that persist through continuous adaptation and recontextualization.

Finally, the spikey decay pattern (Cluster 3) indicates content that generates intensive but intermittent engagement, possibly reflecting complex or niche themes that sustain prolonged community discussion and periodic revival.

These insights offer practical guidance for content creators seeking to optimize for different objectives: immediate viral impact, sustained engagement, or long-term cultural influence. Furthermore, they provide a framework for platform designers to better understand and predict content lifecycle patterns, enabling more effective recommendation algorithms and community management strategies.

Anomaly detection can provide the ability to successfully detect memes whose data was either inaccurately recorded or where the sparsity was too high (which can cause the typical shortcomings of sparse data), further work can be done to enhance the process and achieve a greater outlook of memes who fit poorly into the four clusters. Better identification can yield greater understanding of certain outliers and why they occur. Four clusters may be objectively best for classification, but there will always be memes that simply do not appear to follow any of the four popularity patterns. In addition, it has shown that reducing data sparsity has the potential to better the output of any machine learning algorithms that could be applied going forward.

The dataset's size and automated collection process allowed for the feasibility of machine learning applications in the first place, but some areas are still rudimentary. As previously mentioned, the popularity data for all memes was not able to be successfully requested whether due to networking errors or otherwise. This means that the opportunity to further expand the dataset is always available. Some memes were also described by keywords frequently used to refer to other popular, homonymous terms and therefore the obtained Google Trends data is not accurate in reflecting the month-by-month popularity of the expected meme.

5. Conclusions

This study has successfully developed and validated a machine learning approach to classify internet meme popularity patterns, making several key contributions to the field of computational memetics. First, we significantly expanded the empirical foundation by creating a comprehensive dataset of over 2000 memes, providing a robust basis for pattern analysis. Our two-stage machine learning pipeline, combining unsupervised k-means clustering with supervised SVC, achieved high classification accuracy (98% overall) in identifying four distinct popularity patterns.

The silhouette analysis provided statistical validation for the four-pattern framework originally proposed by [6], while our expanded dataset and machine learning approach offered new insights into pattern characteristics and distribution. The relationship between meme categories and popularity patterns revealed systematic differences in propagation dynamics across different types of internet content.

Our methodological approach demonstrates several advantages over previous work. While we built upon the theoretical foundation established by [6], our machine learning framework offers greater scalability for large datasets and flexibility for handling noisy time-series data. The strong performance of multiple classification algorithms (SVC, random forest, XGBoost) validates the robustness of the pattern discoveries beyond any particular modeling approach.

Our findings have both theoretical and practical implications. Theoretically, they advance our understanding of how different types of digital content propagate through online networks. Practically, the developed classification model could assist in predicting meme trajectory patterns, with potential applications in digital marketing, content strategy, and social media analysis.

Limitations of our study include the exclusive reliance on Google Trends data, which may not capture the full complexity of meme propagation across diverse platforms. Additionally, while our clustering approach revealed four distinct patterns, there may be hybrid or transitional patterns not fully captured by this framework. Future research could address these limitations through multi-platform data collection and more sophisticated clustering techniques.

Future work could extend this research by incorporating hourly popularity data for more granular analysis, developing more sophisticated anomaly detection methods, and investigating additional factors that influence meme propagation patterns. Promising directions include integration with natural language processing and computer vision to analyze meme content alongside temporal patterns, development of more advanced early-stage prediction models, and investigation of cross-platform propagation dynamics. The methodology developed here could also be adapted to study other forms of viral content beyond traditional internet memes.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors would like to thank the anonymous reviewers for their thoughtful and constructive feedback. Their comments helped us refine the literature positioning, clarify methodological details, strengthen the presentation of results and figures, and standardize the references in accordance with the journal's guidelines.

This research received monetary support from the Kennesaw State University Office of Research.

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. L. Shifman, *Memes in Digital Culture*, MIT Press, Cambridge, MA, 2014.
2. D. J. Daley, D. G. Kendall, Epidemics and rumours, *Nature*, **204** (1964), 1118.
3. L. M. A. Bettencourt, A. Cintrón-Arias, D. I. Kaiser, C. Castillo-Chávez, The power of a good idea: Quantitative modeling of the spread of ideas from epidemiological models, *Physica A*, **364** (2006), 513–536. https://doi.org/10.1016/j.physa.2005.08.083
4. K. Nahon, J. Hemsley, Viral politics: Information flows and political behavior on social media, in *Gateway: Understanding Information Flows*, Oxford University Press, 2011.
5. C. Bauckhage, Insights into internet memes, in *Proceedings of the International AAAI Conference on Web and Social Media*, **5** (2011), 42–49. https://doi.org/10.1609/icwsm.v5i1.14097
6. A. Lonnberg, P. Xiao, K. Wolfinger, The growth, spread, and mutation of internet phenomena: a study of memes, *Results Appl. Math.*, **6** (2020), 100092. https://doi.org/10.1016/j.rinam.2020.100092
7. L. Wang, B. C. Wood, An epidemiological approach to model the viral propagation of memes, *Appl. Math. Model.*, **36** (2012), 5442–5447. https://doi.org/10.1016/j.apm.2011.04.035
8. A. Guille, H. Hacid, C. Favre, D. A. Zighed, Information diffusion in online social networks: A survey, *ACM SIGMOD Rec.*, **42** (2013), 17–28. https://doi.org/10.1145/2503792.2503797

9. J. P. Gleeson, J. A. Ward, K. P. O'Sullivan, W. T. Lee, Competition-induced criticality in a model of meme popularity, *Phys. Rev. Lett.*, **112** (2014), 048701. https://doi.org/10.1103/PhysRevLett.112.048701

10. S. A. Myers, J. Leskovec, Clash of the contagions: Cooperation and competition in information diffusion, in *2012 IEEE 12th International Conference on Data Mining*, (2012), 539–548. https://doi.org/10.1109/ICDM.2012.159

11. L. Weng, A. Flammini, A. Vespignani, F. Menczer, Competition among memes in a world with limited attention, *Sci. Rep.*, **2** (2012), 335. https://doi.org/10.1038/srep00335

12. M. Coscia, Competition and success in the meme pool: a case study on quickmeme.com, in *Proceedings of the International AAAI Conference on Web and Social Media*, **7** (2013), 100–109. https://doi.org/10.1609/icwsm.v7i1.14385

13. C. M. Valensise, A. Serra, A. Galeazzi, G. Etta, M. Cinelli, W. Quattrociocchi, Entropy and complexity unveil the landscape of memes evolution, *Sci. Rep.*, **11** (2021), 20022. https://doi.org/10.1038/s41598-021-99468-6

14. K. Barnes, P. Juhász, M. Nagy, R. Molontay, Topicality boosts popularity: a comparative analysis of NYT articles and Reddit memes, *Soc. Network Anal. Min.*, **14** (2024), 119. https://doi.org/10.1007/s13278-024-01272-3

15. K. A. Oliveira, *Modelling Meme Popularity with Networks*, University of Limerick, 2022.

16. P. Wang, M. C. González, C. A. Hidalgo, A. L. Barabási, Understanding the spreading patterns of mobile phone viruses, *Science*, **324** (2009), 1071–1076. https://doi.org/10.1126/science.1167053

17. M. J. Salganik, P. S. Dodds, D. J. Watts, Experimental study of inequality and unpredictability in an artificial cultural market, *Science*, **311** (2006), 854–856. https://doi.org/10.1126/science.1121066

18. E. Ferrara, M. JafariAsbagh, O. Varol, V. Qazvinian, F. Menczer, A. Flammini, Clustering memes in social media, in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, (2013), 548–555. https://doi.org/10.1145/2492517.2492530

19. S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, *Science*, **359** (2018), 1146–1151. https://doi.org/10.1126/science.aap9559

20. D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding, *Proc. SODA*, (2007), 1027–1035.

21. B. E. Wiggins, B. S. Bowers, Memes as genre: A structurational analysis of the memescape, *New Media Soc.*, **17** (2015), 1886–1906. https://doi.org/10.1177/1461444814535194

22. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., Scikit-learn: machine learning in Python, *J. Mach. Learn. Res.*, **12** (2011), 2825–2830. https://doi.org/10.5555/1953048.2078195

23. P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *J. Comput. Appl. Math.*, **20** (1987), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

24. L. Kaufman, P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, New York, 1990.

25. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.

26. T. Fawcett, An introduction to ROC analysis, *Pattern Recognit. Lett.*, **27** (2006), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010
27. S. Godbole, S. Sarawagi, Discriminative methods for multi-labeled classification, in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, (2004), 22–30. https://doi.org/10.1007/978-3-540-24775-3_5

AIMS Press