TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1221-1231
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10084

Analysis of Student GitHub Repository Activity Patterns in Web Framework Programming Courses using K-Means Clustering

Yori Adi Atma*, Andrew Kurniawan Vadreas

Department of Information Technology, Politeknik Negeri Padang, Padang, Indonesia
Email: 1,*yori@pnp.ac.id, 2andrew@pnp.ac.id
Correspondence Author Email: yori@pnp.ac.id

Abstract–Instructors in project-based web framework courses frequently lack objective mechanisms to monitor individual student development progress, particularly when engagement occurs asynchronously across multiple weeks and repository activity is not systematically analyzed. This study aims to identify distinct behavioral engagement profiles among students using GitHub repository activity data, and to demonstrate the utility of unsupervised machine learning as a scalable progress monitoring tool for project-based programming courses. The K-Means Clustering algorithm was applied to analyze repository activity patterns of 73 students enrolled in a Web Framework Programming course using Laravel at a vocational higher education institution. Five behavioral features were extracted from each student's GitHub repository, namely total_commit, active_days, avg_commit_per_day, weekend_commit, and last_commit_gap. Following data normalization using StandardScaler, the optimal number of clusters was identified as k=3 using the Elbow Method. The clustering analysis revealed three distinct behavioral profiles: Cluster 0 (51 students, 69.86%) as Passive Learners characterized by low commit activity and a high last_commit_gap indicating deadline-driven development behavior; Cluster 2 (20 students, 27.40%) as Productive Learners demonstrating substantially higher commit intensity and broader repository engagement; and Cluster 1 (2 students, 2.74%) as Highly Consistent Learners exhibiting stable, multi-session repository interaction throughout the project period. As an initial validation of clustering quality, the Silhouette Score of 0.4041 confirms a moderate yet meaningful partition structure within the dataset. The primary contribution of this study lies in demonstrating that mandatory GitHub repository submissions, already required in most project-based programming courses, can be repurposed into an objective, low-cost behavioral monitoring instrument without additional data collection burden. This contributes a replicable, repository-based learning analytics framework that enables instructors to objectively classify student project engagement, supporting early instructional intervention and more process-oriented assessment strategies in software engineering education.

Keywords: GitHub Repository; K-Means Clustering; Laravel; Learning Analytics; Web Framework Programming

1. INTRODUCTION

Web framework programming has become one of the most practically oriented courses in informatics and software engineering curricula, owing to the direct relevance of web development competencies to industry demands (Sunardi & Suharjito, 2019). The Laravel PHP framework, in particular, has gained substantial adoption in Indonesian higher education settings as the primary teaching framework for backend web development courses, given its well-structured MVC architecture, expressive Eloquent ORM, and extensive ecosystem that mirrors professional development practice (Cakra et al., 2026). In courses that use Laravel as the primary teaching framework, student learning is most effectively demonstrated not through written examinations but through the development of functional web application projects, an approach that aligns fundamentally with Project-Based Learning (PBL) principles (Saad & Zainudin, 2022).

Project-Based Learning in programming education requires students to engage with a development problem over an extended period, iteratively building, testing, and refining their codebase across multiple work sessions (Alsmadi et al., 2024). GitHub, as the dominant version control and code hosting platform, is the natural infrastructure for this kind of iterative development (Rostami Mazrae et al., 2026). When instructors mandate that students maintain a GitHub repository for their project, the platform simultaneously serves as a submission medium and as a behavioral data source. Every commit a student pushes to the repository is timestamped, attributable, and auditable, creating a longitudinal trace of the student's development process that is far richer than a single submitted ZIP file would reveal. The ability to observe not just what students produced, but how and when they built it, represents a significant analytical opportunity.

Despite this opportunity, most instructors who require GitHub repository submission do not systematically analyze the behavioral data that repositories generate. Assessment typically focuses on the final state of the repository, the completeness of features, code quality, and documentation, rather than on the process by which students arrived at that final state. This leaves a critical dimension of student engagement unexamined. A student who commits code in small, regular increments across the entire project duration is engaging with the material in a fundamentally different way from a student who submits a large volume of commits in the final two days before the deadline, even if both repositories contain similar final products (Decan et al., 2023). Learning Analytics provides the methodological foundation for extracting and interpreting these behavioral signals in a principled and scalable way (Sailer et al., 2024).

Recent studies have increasingly explored the application of clustering and learning analytics techniques to understand student behavioral patterns in digital learning environments. (Cui et al., 2024) utilized a constrained K-Means algorithm combined with pre-class GitHub contribution statistics to support the formation of balanced student teams in programming education, demonstrating the potential of repository activity data as an objective indicator of collaborative readiness. (Xu, 2025) applied K-Means clustering to online learning behavior data and showed that clustering-based approaches can effectively identify distinct learner engagement patterns in technology-supported educational settings. Similarly, (Mohd Talib et al., 2023) applied K-Means clustering to identify distinct student behavioral patterns in higher education, demonstrating the utility of unsupervised learning methods for detecting latent


TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1221-1231
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10084

behavioral structures in educational datasets. Beyond K-Means approaches, (Delgado et al., 2021) analyzed student behavior in online learning environments using Self-Organizing Maps (SOM) neural networks and demonstrated that user clustering techniques can provide valuable insight into participation patterns and learning engagement dynamics. Although these studies confirmed the effectiveness of clustering-based learning analytics in educational contexts, most focused on Learning Management System (LMS) interaction logs or generalized online learning behavior. Limited attention has been given to repository-centered behavioral analytics in web framework programming courses, particularly in project-based Laravel development environments where GitHub repositories provide structured, longitudinal records of student software engineering activities.

The practical challenge motivating this study is the difficulty of individual-level project progress monitoring in web framework courses with class sizes typical of Indonesian universities. When an instructor assigns a Laravel project to 73 students and collects GitHub repository links as the submission mechanism, manually reviewing each repository's commit history to assess individual engagement is operationally infeasible. The instructor can observe the final product, whether the movie database application is complete, whether routes are correctly defined, whether migrations were implemented, whether Blade templates are well-structured, but cannot easily distinguish, at scale, between students who worked consistently throughout the semester and those who assembled their project in a rushed final effort. This distinction matters pedagogically because late-stage concentrated effort may produce a passable final product while failing to develop the iterative debugging, version management, and incremental feature development skills that the course intends to build.

Unsupervised machine learning, particularly K-Means Clustering, provides an effective and computationally efficient approach for identifying behavioral patterns within multidimensional educational datasets. Cluster analysis operates by grouping data objects based on similarity or distance measures, enabling observations with comparable characteristics to be partitioned into meaningful clusters (Oti et al., 2021). In the context of this study, each student's GitHub repository activity is represented as a five-dimensional behavioral vector consisting of commit volume, temporal distribution of work sessions, commit intensity, voluntary weekend engagement, and recency of repository activity. By applying K-Means to these behavioral features, students can be grouped into distinct engagement profiles without requiring manual inspection of individual repositories.

K-Means is particularly suitable for this task because it is one of the most widely adopted partitioning-based clustering algorithms and has demonstrated effectiveness in handling numerical datasets with relatively low computational complexity (Oti et al., 2021). The algorithm iteratively assigns each data point to the nearest centroid using Euclidean distance and continuously updates cluster centroids until convergence is achieved. This centroid-based representation allows behavioral patterns to be interpreted clearly and intuitively, making the resulting clusters pedagogically meaningful for instructors. Furthermore, the scalability and interpretability of K-Means make it highly appropriate for classroom-scale learning analytics applications in which behavioral monitoring must be performed efficiently across large groups of students (Tuyishimire et al., 2022).

The research gap addressed by this study is the absence of a repository-based behavioral clustering framework specifically designed for web framework programming courses using Laravel in the Indonesian vocational campus context. Existing learning analytics studies in programming education have focused predominantly on LMS interaction logs, general programming exercise platforms, or introductory computer science courses. The specific behavioral dynamics of a semester-long web application project, where students must integrate routing, controllers, models, migrations, and views in a coherent application structure, produce a distinct pattern of repository activity that warrants dedicated investigation. This study contributes an applied K-Means Clustering methodology that can be directly replicated by instructors of web framework courses who use GitHub as a project monitoring platform.

The primary objective of this study is to identify distinct behavioral clusters among students based on their GitHub repository activity during a Web Framework Programming course project using the Laravel framework. Specific objectives include: (1) applying K-Means Clustering to multi-dimensional repository activity data to segment students into behaviorally meaningful groups; (2) evaluating clustering quality using the Silhouette Score; (3) interpreting each cluster in terms of its pedagogical implications for project-based Laravel instruction; and (4) demonstrating the feasibility of repository-based learning analytics as a practical and scalable instructor tool for project progress monitoring at scale.

There are two things this study adds to the field of educational data mining. First, from a methodological perspective, it lays out a complete, end-to-end K-Means clustering pipeline tailored to Laravel-based web project repositories, one that other researchers can reproduce. Second, from a practical perspective, it offers instructors a three-tier behavioral taxonomy they can use directly for early-warning intervention, without needing any machine learning expertise. In addition to identifying behavioral engagement patterns, this study also aims to provide empirical insight into how repository analytics can support data-driven instructional strategies in programming education. By analyzing behavioral indicators extracted directly from version control activity, instructors may obtain a more objective understanding of student participation consistency, work distribution habits, and development discipline throughout collaborative software projects. The findings are expected to contribute to the growing body of educational data mining research by demonstrating how repository-centered analytics can complement conventional assessment approaches in modern software engineering education environments.

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1221-1231
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10084

2. RESEARCH METHODOLOGY

2.1 Research Design

This study adopts a quantitative, exploratory research design employing unsupervised machine learning to analyze behavioral patterns within educational datasets. Quantitative research designs are appropriate when the objective involves measuring observable phenomena through numerical data and identifying structural patterns through statistical or computational methods (Papatheodosiou et al., 2024). In this study, the primary data source consists of GitHub repository activity records, which are inherently numerical and timestamped, making them suitable for quantitative behavioral analysis. The research does not evaluate student academic performance or test a pedagogical hypothesis; rather, it applies a data-driven clustering methodology to behavioral data extracted from version control repositories to uncover latent structure in student project engagement patterns. The approach is situated within the field of Educational Data Mining (EDM), specifically the sub-domain of learning analytics applied to version control system data. EDM is defined as the application of data mining techniques to educational datasets for the purpose of understanding learners and the environments in which they learn, with the ultimate goal of improving educational outcomes (Romero & Ventura, 2020). By treating GitHub commit history as a behavioral dataset, this study operationalizes learning analytics within the context of a project-based web framework programming course, contributing a methodological framework that is replicable across similar instructional settings.

2.2 Course Context and Dataset

The dataset was compiled from GitHub repository activity logs of 73 students enrolled in a Web Framework Programming course using Laravel at a vocational campus during the 2024/2025 academic semester. As part of the course requirements, each student was mandated to develop a movie database web application using the Laravel PHP framework and to maintain a GitHub repository that would serve as both the development environment and the project submission mechanism. This setup allowed the instructor to monitor each student's development progress throughout the semester by observing commit history, active development periods, and the recency of contributions. Repository URLs were submitted by students at the beginning of the project period, and activity data was collected at the end of the semester using the GitHub REST API. The project task, building a functional movie database with features including movie listing, detail views, search functionality, and user authentication, was standardized across all students, ensuring that the behavioral features extracted from repositories are directly comparable.

Five behavioral features were extracted from each student's repository commit history. Table 1 presents a statistical summary of these features across the 73 student dataset.

Table 1. Statistical Summary of GitHub Repository Activity Features (n = 73 Students)

Feature	Min	Max	Mean	Std Dev	Description
total_commit	1	21	10.86	3.46	Total Git commits pushed
active_days	1	9	5.40	1.58	Unique days with ≥1 commit
avg_commit_per_day	1.00	4.25	2.02	0.55	Commits per active day
weekend_commit	0	2	0.19	0.49	Commits on Sat/Sun
last_commit_gap	75	372	337.25	47.33	Days since last commit (from reference date)

The feature total_commit captures the cumulative number of commits pushed to the repository across the entire project period, serving as a direct proxy for total development effort. In the Laravel project context, each meaningful commit typically corresponds to the completion of a discrete development task, creating a migration, implementing a controller method, writing a Blade view, or configuring a route, making commit count a reasonable indicator of project construction activity. The feature active_days records the number of distinct calendar days on which at least one commit was recorded, capturing the temporal distribution of work sessions and distinguishing students who spread their project work across many sessions from those who concentrated effort into a small number of intensive sessions. The feature avg_commit_per_day reflects the commit intensity per active session, computed as the ratio of total_commit to active_days, and indicates how productively students used the time they did allocate to the project. The feature weekend_commit records commits made on Saturdays or Sundays, providing an indicator of voluntary engagement outside of scheduled class or laboratory hours, a behavioral signal associated with intrinsic motivation in academic project work. Finally, last_commit_gap measures the number of days elapsed between the most recent commit and a fixed reference date at the end of the semester, functioning as a recency indicator that reveals whether students maintained active development throughout the project period or ceased activity well before the final submission.

2.3 Research Workflow

The research followed eight sequential phases, shown in Figure 1, moving from raw data collection through feature engineering, preprocessing, normalization, clustering, and final interpretation. Structuring the work this way keeps each analytical decision traceable and makes the process easier to reproduce (Compagnucci et al., 2025).

Phase 1 (Data Collection) pulled commit history metadata, timestamps, commit counts, and author attribution, from the GitHub REST API using the repository URLs students submitted. Phase 2 (Feature Extraction) turned that raw

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1221-1231
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10084

commit data into the five behavioral features described earlier, producing a 73×5 feature matrix. Phase 3 (Data Preprocessing) checked for missing values and confirmed all features were numeric; no imputation was needed since every repository had valid commit records. Phase 4 (Data Normalization) standardized the features with StandardScaler to remove scale bias before distance calculations. Phase 5 (Optimal k Determination) ran the Elbow Method across k=2 to k=8 to find the best cluster count. Phase 6 (K-Means Clustering) fitted the algorithm on the normalized data using that selected k. Phase 7 (Cluster Evaluation) computed the Silhouette Score to check how well-separated the resulting clusters were. Phase 8 (Interpretation and Visualization) analyzed the centroids in their original scale, labeled the behavioral profiles, and produced the scatter plots and distribution tables used to interpret the results.

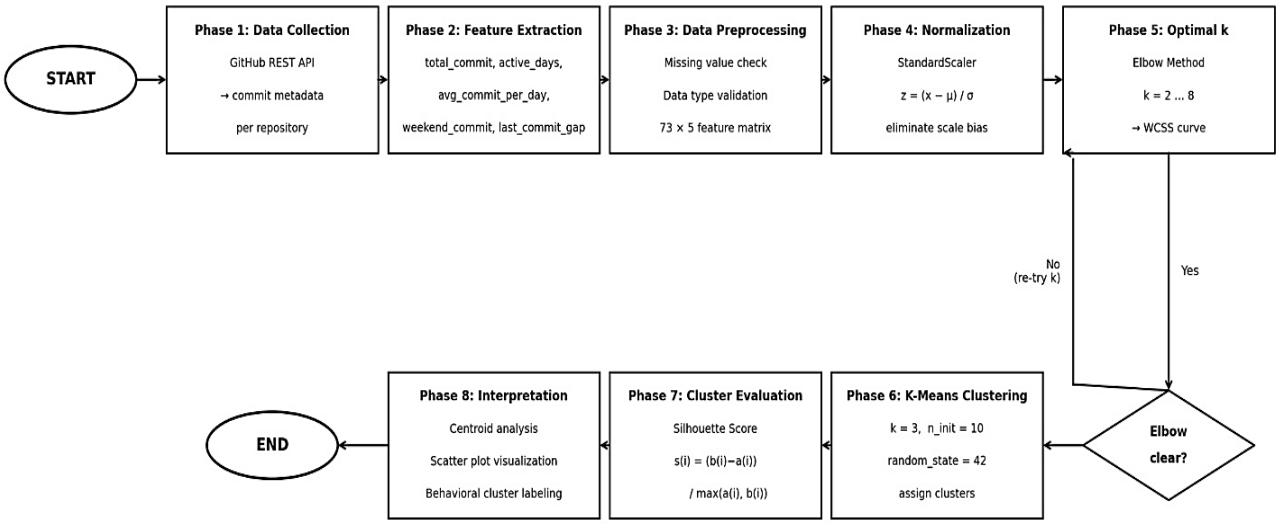

Figure 1. Research Workflow Diagram for K-Means Clustering-Based GitHub Activity Analysis

2.4 Data Normalization

Feature normalization was applied using the StandardScaler transformation, which standardizes each feature to zero mean and unit variance. For a feature value x drawn from a distribution with sample mean μ and standard deviation σ, the standardized value z is given by Equation 1:

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

Normalization is essential in this dataset because the features span dramatically different numerical ranges: last_commit_gap ranges from 75 to 372 days, while weekend_commit ranges from 0 to 2. Without standardization, the Euclidean distance computation underlying K-Means would be dominated by features with larger absolute values, rendering the clustering results misleading (Liew et al., 2024). After applying StandardScaler, each feature contributes proportionally to the distance calculation based on its relative variability rather than its absolute scale.

2.5 Elbow Method for Optimal Cluster Determination

The number of clusters k is a required hyperparameter for K-Means that must be specified prior to fitting. The Elbow Method determines k empirically by plotting the Within-Cluster Sum of Squares (WCSS), also called inertia, as a function of k (Liew et al., 2024). WCSS quantifies the total squared Euclidean distance between each data point and the centroid of its assigned cluster, as defined in Equation 2, where K is the number of clusters, Ck is the set of points in cluster k, and μk is the centroid of cluster k:

$$WCSS = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \quad (2)$$

The Euclidean distance between a data point xi and centroid μk in an n-dimensional feature space is computed per Equation 3:

$$d(x_i, \mu_k) = \sqrt{\sum_{t=1}^{n} (x_{it} - \mu_{kt})^2} \quad (3)$$

The candidate range k=2 to k=8 was evaluated, yielding WCSS values of 273.46, 207.10, 149.72, 116.40, 99.15, 82.53, and 70.80 respectively. The resulting curve is presented in Figure 2. As shown in Figure 2, the WCSS curve exhibits a transition in slope at k=3, where the rate of WCSS reduction begins to diminish relative to earlier increments. The reduction from k=2 to k=3 (approximately 66.4 units) is the largest single-step decrease after k=2, and the subsequent decrease from k=3 to k=4 (57.4 units) is followed by progressively smaller gains. In combination with the natural interpretability of three behavioral categories in an educational monitoring context, distinguishing highly engaged, moderately engaged, and low-engagement student profiles, k=3 was selected as the optimal cluster count.

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1221-1231
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10084

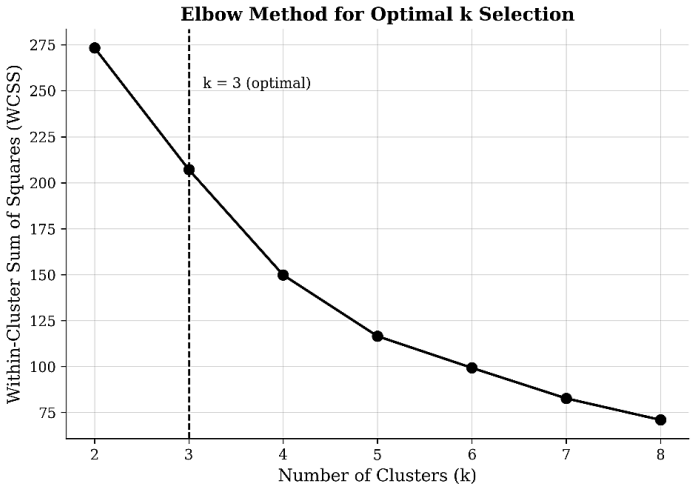

Figure 2. Elbow Method Results for K-Means Cluster Selection (k = 2 to k = 8)

2.6 K-Means Clustering Implementation

The K-Means algorithm partitions a dataset of n observations into K non-overlapping clusters by iteratively minimizing WCSS. The algorithm initializes K centroids, assigns each data point to the nearest centroid via Euclidean distance, recomputes centroids as the mean of assigned points, and repeats until convergence. In this study, the scikit-learn KMeans implementation was used with k=3, random_state=42 for reproducibility, and n_init=10 to ensure robustness against sensitivity to initial centroid placement by running ten independent initializations and retaining the solution with the lowest WCSS.

The selection of K-Means was also motivated by the relatively low dimensionality and numerical nature of the repository activity dataset. Since all extracted behavioral features were continuous numerical variables with comparable semantic meaning after normalization, centroid-based clustering was considered appropriate for identifying naturally emerging engagement patterns. Furthermore, K-Means has been widely adopted in educational data mining studies due to its low computational complexity and ease of interpretation in classroom-scale datasets (Atma & Montesori, 2022).

2.7 Silhouette Score Evaluation

Clustering quality was assessed using the Silhouette Score, an intrinsic evaluation metric that requires no ground-truth labels (Vardakas et al., 2026). For each data point i, the silhouette coefficient $s(i)$ is defined in Equation 4, where $a(i)$ is the mean intra-cluster distance (cohesion) and $b(i)$ is the mean distance to the nearest neighboring cluster (separation):

$$s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))} \quad (4)$$

The Silhouette Score ranges from -1 to +1. Values in the range 0.25-0.50 indicate weak but present cluster structure; 0.50-0.70 indicate reasonable structure; and above 0.70 indicate strong structure. The overall score is the mean of $s(i)$ across all data points (Mulyani et al., 2023).

3. RESULTS AND DISCUSSION

3.1 K-Means Clustering Process on Repository Activity Data

The K-Means algorithm was executed on the normalized 73×5 feature matrix produced from student GitHub repository activity data. Prior to the first iteration, three initial centroids were selected using the k-means++ initialization strategy, which places the first centroid randomly and subsequent centroids with probability proportional to their squared distance from the nearest already-selected centroid. This initialization approach was adopted to reduce sensitivity to random starting positions and improve convergence stability, as implemented via the n_init=10 parameter in the scikit-learn KMeans function, which ran ten independent initializations and retained the solution yielding the lowest WCSS.

In the first iteration, each of the 73 student data points was assigned to the nearest centroid by computing the Euclidean distance between the student's five-dimensional feature vector and each of the three centroids. Upon completion of the initial assignment, three preliminary clusters were formed. The centroid of each cluster was then recomputed as the arithmetic mean of all feature vectors belonging to that cluster, producing three updated centroid positions in the five-dimensional normalized feature space. In subsequent iterations, cluster assignments were recalculated based on the updated centroids, and centroids were recomputed again after each reassignment. This assignment-and-update cycle continued until no data point changed its cluster membership between consecutive iterations, indicating that the algorithm had reached convergence. With random_state=42 ensuring reproducibility, the

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1221-1231
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10084

algorithm consistently converged to the same final partition across all ten initializations, confirming the stability of the k=3 solution on this dataset.

Upon convergence, the final cluster assignments were mapped back to the original (non-normalized) feature space to produce interpretable centroid values. The three resulting centroids, presented in Table 2, represent the mean behavioral profile of each student group across all five repository activity features. The stability of convergence and the consistency of centroid positions across multiple initializations provide confidence that the identified clusters reflect genuine structure in the data rather than artifacts of a particular starting configuration.

3.2 Elbow Method Analysis

Figure 2 presents the WCSS curve across *k*=2 to *k*=8. The most substantial reduction in WCSS occurs between *k*=2 (273.46) and *k*=3 (207.10), corresponding to a decrease of 66.4 units. Subsequent reductions are smaller and progressively diminishing: 57.4 units from *k*=3 to *k*=4, 33.3 units from *k*=4 to *k*=5, and 17.2 units from *k*=5 to *k*=6. The inflection point at *k*=3 indicates that adding a fourth cluster beyond this point yields comparatively smaller compactness gains while increasing model complexity and reducing interpretability. Given that three clusters map naturally onto the behavioral categories observable in a project-based course, students who work consistently throughout the project, students who produce a high volume of work in concentrated sessions, and students who engage minimally or only at deadline, k=3 was confirmed as the operationally optimal and educationally meaningful selection.

3.3 Clustering Quality: Silhouette Score

The Silhouette Score computed for the *k*=3 solution on the normalized feature matrix was 0.4041, placing the result in the weak-but-present structure range (0.25-0.50). This score is consistent with the nature of behavioral data derived from a single cohort in one course: commit behavior within a class is inherently more homogeneous than behavior across diverse populations, and the boundaries between engagement levels are continuous rather than discrete. It is worth noting that *k*=4 produced a marginally higher Silhouette Score of 0.4422; however, the fourth cluster in that configuration split the Passive Learner group without producing a fourth behaviorally distinct and pedagogically interpretable category. The *k*=3 solution was therefore retained as the superior choice on grounds of both Elbow Method evidence and interpretive clarity. The Silhouette Score of 0.4041 confirms that the three clusters identified are meaningfully distinct, students in different clusters exhibit genuinely different repository engagement profiles, while acknowledging the natural overlap at behavioral boundaries.

3.4 Cluster Centroids and Student Profiles

Following K-Means convergence at *k*=3, the algorithm produced three cluster centroids computed in the original feature space. Table 2 presents these centroid values alongside the number of students assigned to each cluster, providing the quantitative foundation for behavioral profile interpretation.

Table 2. K-Means Cluster Centroids in Original Feature Scale and Cluster Sizes

Feature	Cluster 0 (Passive Learner)	Cluster 1 (Highly Consistent Learner)	Cluster 2 (Productive Learner)	Unit
total_commit	9.20	12.00	15.00	commits
active_days	4.98	8.00	6.20	days
avg_commit_per_day	1.89	1.51	2.55	commits/day
weekend_commit	0.04	0.00	0.60	commits
last_commit_gap	348.02	79.00	335.60	days
Cluster Size (n)	51 students	2 students	20 students	-

Cluster 0, designated as the Passive Learner group, is the largest cluster with 51 students representing 69.86% of the total cohort. The centroid values in Table 2 reveal a profile of low-to-moderate repository activity: an average of 9.20 total commits across approximately 5 active days, an average commit intensity of 1.89 commits per active day, and near-zero weekend engagement (0.04). The most diagnostically significant feature for this cluster is the last_commit_gap of 348.02 days, the highest among all three clusters. This value indicates that the most recent commit in these repositories was made very early in the project timeline, that is, students in this group ceased active development well before the submission deadline, or the bulk of their work was confined to an initial burst at project launch. In the context of the Laravel movie database project, this pattern suggests that many students may have initialized the repository, made a few early commits during the scaffolding phase (running laravel new, setting up the database configuration, and generating initial models via artisan), and then either stopped committing while continuing to work locally, a behavioral artifact sometimes arising from misunderstanding of version control workflow, or genuinely reduced their engagement with the project. Either interpretation has implications for course delivery that warrant instructor attention.

Cluster 1, designated as the Highly Consistent Learner group, contains only 2 students but represents the most educationally distinctive behavioral profile in the dataset. These students exhibit a moderate total_commit count of 12.00 but the highest active_days value of 8.00 and, most critically, the lowest last_commit_gap of only 79.00 days. The last_commit_gap value of 79 days, compared to the cohort average of 337 days, indicates that these students' most

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1221-1231
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10084

recent commits were made substantially more recently than their peers relative to the reference date, suggesting ongoing iteration and refinement of the project well beyond the initial development phase. The lower avg_commit_per_day of 1.51 combined with high active_days indicates a pattern of distributed, low-intensity daily engagement consistent with what learning science literature terms distributed practice, a strategy associated with deeper procedural skill acquisition in technical domains. In the Laravel project context, this profile corresponds to a student who regularly opens the project, makes a few targeted changes (implementing a new route, adjusting a Blade component, debugging an Eloquent query), commits those changes, and returns the next day for another session, an approach that closely mirrors professional software development practice.

Cluster 2, designated as the Productive Learner group, comprises 20 students representing 27.40% of the cohort. This group exhibits the highest total_commit average of 15.00, a moderate active_days value of 6.20, and the highest avg_commit_per_day of 2.55, indicating that when these students work, they commit code at a substantially higher rate than students in other clusters. The weekend_commit average of 0.60 is also the highest among the three clusters, suggesting a meaningful proportion of voluntary off-hours engagement. The last_commit_gap of 335.60 days, while still high, is moderately lower than that of Cluster 0, indicating somewhat more recent repository activity. In the Laravel project context, this profile corresponds to students who engage in sprint-based development: allocating focused, multi-hour work sessions on selected days (including weekends) during which they implement multiple features in sequence, for example, creating a complete CRUD module for movie entries, including the controller, model relationships, migration, and corresponding Blade views, within a single commit session. While this approach is less aligned with professional iterative development norms than the Consistent Learner profile, it does reflect genuine engagement with the project task and produces a high-volume output.

3.5 Distribution of Students Across Clusters

Table 3 summarizes the distribution of students across the three clusters, providing a quantitative overview of the cohort's behavioral composition and the dominant engagement pattern within the class.

Table 3. Distribution of Students Across K-Means Clusters and Behavioral Interpretation

Cluster	Label	N	Percentage	Key Behavioral Indicators
0	Passive Learner	51	69.86%	Moderate commits, few active days, very high last_commit_gap; likely deadline-driven or early-stop pattern
1	Highly Consistent Learner	2	2.74%	High active days, very low last_commit_gap; sustained multi-session engagement throughout project period
2	Productive Learner	20	27.40%	High total commits, high daily commit rate, moderate weekend activity; intensive sprint-based development sessions

The distribution presented in Table 3 reveals a markedly imbalanced engagement landscape within the cohort. The dominance of the Passive Learner cluster, encompassing nearly 70% of all students, indicates that the majority of the class did not engage with the GitHub repository in a manner consistent with sustained iterative development. This finding is particularly relevant in the context of a Laravel project, where the development process is inherently incremental: each feature of the movie database application for examples the movie CRUD interface, the search functionality, the authentication system, the detail view with related movies are represents a natural commit boundary that, in an ideal workflow, would be committed individually and progressively. The clustering result suggests that most students in this cohort either did not adopt this incremental workflow or did adopt it locally without committing their progress to the remote repository with sufficient frequency.

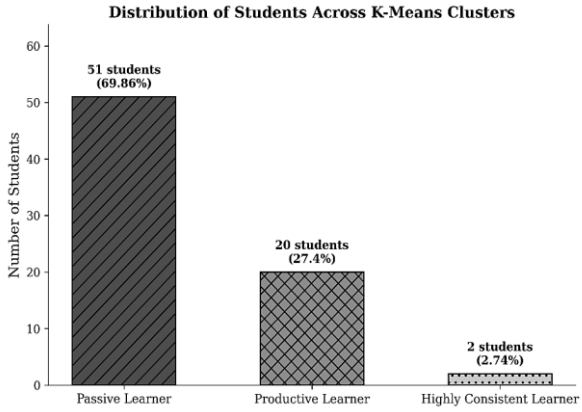

Figure 3. Distribution of Students Across Behavioral Clusters

Figure 3 illustrates the distribution of students across the three behavioral clusters. The Passive Learner cluster dominates the cohort with 51 students (69.86%), followed by Productive Learners with 20 students (27.40%), and

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1221-1231
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10084

Highly Consistent Learners with only 2 students (2.74%). This distribution indicates that sustained, iterative repository engagement remains relatively uncommon within the observed cohort.

The dominance of the Passive Learner cluster suggests that a substantial proportion of students demonstrated limited interaction with version control practices throughout the project lifecycle. This may reflect irregular development habits, low commit discipline, or a tendency to work offline before committing accumulated changes, all of which reduce the transparency of individual project progression for instructors.

In contrast, Productive Learners exhibited substantially higher commit volume and broader active-day distribution, indicating stronger engagement with the development process and a closer alignment with iterative software engineering workflows. Meanwhile, the Highly Consistent Learner cluster, despite comprising only two students, represents the most disciplined behavioral profile, characterized by regular commit distribution across active days and minimal commit gaps, reflecting strong self-regulated learning tendencies that are particularly valuable in project-based programming courses.

3.6 Scatter Plot Visualization

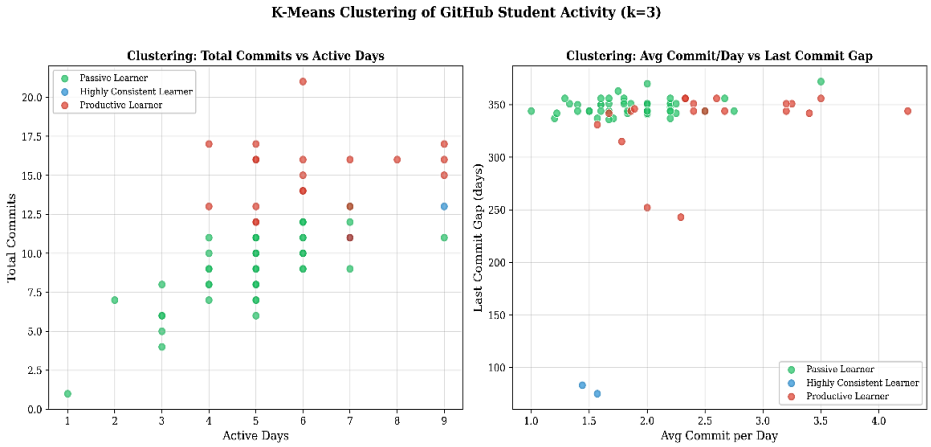

Figure 4. K-Means Clustering Scatter Plots: (Left) Total Commits vs. Active Days; (Right) Avg Commit/Day vs. Last Commit Gap

Figure 4 presents two scatter plots that visualize the clustering results across key feature pairs, with each data point colored by cluster assignment: green for Cluster 0 (Passive Learner), blue for Cluster 1 (Highly Consistent Learner), and red for Cluster 2 (Productive Learner). The left panel of Figure 4 plots total_commit on the vertical axis against active_days on the horizontal axis. The separation between Cluster 0 (green) and Cluster 2 (red) is clearly visible along the vertical axis, with Cluster 2 students consistently occupying higher positions corresponding to their elevated total commit counts (centroid: 15.00), while Cluster 0 students are concentrated in the middle portion of the plot (centroid: 9.20). Both clusters exhibit comparable active_days ranges, reflecting that the distinction between Passive and Productive Learners is primarily driven by the volume and intensity of commits rather than the number of distinct work sessions. The two Cluster 1 students (blue) are distinctly positioned at the high end of the active_days axis with moderate commit counts, separated from the main body of Cluster 0 and Cluster 2 students, which visually confirms their unique behavioral profile.

The right panel of Figure 4, plotting avg_commit_per_day against last_commit_gap, provides the most diagnostically informative visualization. The most prominent pattern is the near-complete vertical separation between Cluster 1 (blue, positioned at very low last_commit_gap values of approximately 75-83 days) and Clusters 0 and 2 (both densely concentrated at last_commit_gap values of approximately 300-372 days). This stark separation confirms that last_commit_gap is the primary feature discriminating the Highly Consistent Learner cluster from the rest of the cohort. Within the high-gap region, the horizontal separation between Cluster 2 (red, positioned further right at higher avg_commit_per_day values, centroid 2.55) and Cluster 0 (green, centroid 1.89) is visible, though with some overlap reflecting the moderate Silhouette Score of 0.4041. Collectively, the two panels in Figure 4 confirm that the K-Means algorithm has identified genuine behavioral structure in the data rather than arbitrary partitions, with each cluster occupying a distinct region of the feature space that corresponds to an interpretable educational profile.

3.7 Implications for Project-Based Laravel Course Monitoring

The cluster structure identified in this study carries direct and actionable implications for how instructors can use GitHub-based learning analytics to improve project monitoring and intervention in web framework programming courses. The most immediate practical application is the use of cluster assignments as a real-time monitoring dashboard. If repository activity data is collected at intermediate project checkpoints, for instance, at the two-week, four-week, and six-week marks of an eight-week project period, the K-Means model can be applied incrementally to identify students whose emerging behavioral profile matches the Passive Learner cluster before the project reaches its final submission stage. Students flagged at the four-week mark as low-commitment can then be targeted for instructor office hours

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1221-1231
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10084

invitations, peer study group assignments, or structured mid-project code reviews, giving the intervention time to meaningfully alter the student's development trajectory.

The finding that 69.86% of students fall into the Passive Learner cluster also carries implications for course structure. The high last_commit_gap values characteristic of this cluster suggest that a significant proportion of the cohort treats the project as a single deadline event rather than an iterative development process. This behavioral pattern can be structurally discouraged by implementing mandatory intermediate commit milestones, for example, requiring that students demonstrate a working authentication module by week three, a functional movie listing page by week five, and complete search and detail view functionality by week seven. Binding grade components to these intermediate deliverables, and using the GitHub repository as the evidence mechanism, would create accountability checkpoints that reward distributed engagement and penalize deadline-concentrated effort in a way that is directly observable from the repository data.

The Productive Learner cluster (Cluster 2), while exhibiting higher total commits and daily productivity than the Passive group, still shows a high last_commit_gap relative to the Consistent Learner ideal. This suggests that sprint-based students are effective at producing code volume during their active sessions but may not maintain ongoing engagement with the project between sprints. An instructor aware of this profile could design supplementary challenges or optional enhancement tasks, for example, adding an IMDB API integration, implementing a recommendation engine, or deploying the application to a cloud platform, that give productive students meaningful reasons to return to the project across additional sessions, thereby extending their engagement duration and reducing the last_commit_gap.

Finally, the two students in the Highly Consistent Learner cluster exemplify the behavioral ideal for a project-based Laravel course: regular, distributed development sessions spread across many active days, with the most recent commit occurring substantially more recently than their peers. Their behavioral patterns, particularly the combination of high active_days and low last_commit_gap, can serve as a concrete, data-visible standard of engagement that instructors can share with the broader cohort at the beginning of the project period. Communicating to students that repository activity patterns will be analyzed and interpreted through learning analytics tools may itself function as a behavioral nudge, encouraging students to develop more consistent commit habits from the outset of the project.

3.8 Methodological and Practical Implications

Beyond its immediate application in project monitoring, the methodology presented in this study also demonstrates the broader feasibility of repository-centered learning analytics within software engineering education environments. One of the primary advantages of GitHub-based behavioral analysis lies in its unobtrusive and continuously generated nature. Unlike traditional learning analytics approaches that depend on questionnaires, self-report surveys, or manual observation, repository activity data is automatically produced as a natural consequence of the software development workflow itself. This characteristic reduces observer bias and enables scalable monitoring across large student cohorts without increasing instructor workload.

From a methodological perspective, the use of K-Means Clustering provides an interpretable and computationally efficient approach for identifying latent behavioral structures in programming project environments. Although more complex machine learning approaches such as DBSCAN, Gaussian Mixture Models, or hierarchical clustering could potentially capture more nuanced behavioral relationships, K-Means offers an important balance between analytical simplicity and pedagogical interpretability (Lashiyanti et al., 2023). In educational contexts, interpretability is particularly critical because the resulting behavioral categories must be understandable to instructors who may not possess advanced machine learning expertise. The centroid-based representation produced by K-Means allows instructors to directly associate each cluster with observable repository behaviors such as commit intensity, work distribution, and recency of engagement.

The findings also highlight an important distinction between productivity and consistency in software development learning behavior. Students in the Productive Learner cluster demonstrated high commit volume and elevated daily commit intensity, yet their high last_commit_gap values indicate that productivity alone does not necessarily reflect sustained engagement. Conversely, the Highly Consistent Learner cluster exhibited lower daily commit intensity but substantially more distributed activity across time. This distinction aligns closely with professional software engineering practice, where maintainability, iterative refinement, and continuous integration are often more valuable than short-term bursts of coding productivity. Consequently, repository analytics should not be interpreted solely as indicators of coding quantity, but rather as behavioral signals reflecting underlying development habits and workflow discipline.

Another important implication concerns the integration of repository analytics into formative assessment strategies. In many programming courses, instructors evaluate only the final functionality of the application, resulting in limited visibility into the development process itself. By incorporating behavioral indicators such as active_days and last_commit_gap into assessment frameworks, instructors may encourage students to adopt healthier development practices, including incremental commits, regular testing, and continuous version management. Such integration would shift assessment emphasis from purely outcome-oriented evaluation toward process-oriented learning evaluation, which is more aligned with the iterative philosophy of modern software engineering education.

Despite the promising results, several limitations should also be acknowledged. First, the dataset represents a single course cohort within one institution, which may limit the generalizability of the resulting behavioral patterns to other educational settings or programming disciplines. Second, repository activity metrics do not directly measure code

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1221-1231
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10084

quality, conceptual understanding, or project complexity. A student may produce many commits with minimal functional contribution, while another may produce fewer but technically substantial commits. Third, the GitHub API captures only committed repository activity and cannot observe offline development behavior occurring outside the version control workflow. Future research should therefore integrate repository analytics with complementary indicators such as automated code quality analysis, issue tracking behavior, pull request interactions, or final project evaluation scores to obtain a more comprehensive representation of student software development behavior.

3.9 Discussion

These results connect to earlier work on clustering-based learning analytics in a few useful ways. (Cui et al., 2024) used a constrained K-Means approach on pre-class GitHub statistics to form balanced student teams, and found that raw contribution metrics carry real predictive value. The present findings point in a similar direction under different conditions: here the clustering covers a full semester of individual Laravel project work rather than a short pre-class snapshot, yet the same kind of commit-based features still produce clean separation between engagement levels. That consistency is a useful confirmation that repository activity holds up as a clustering signal beyond its original use case.

A comparable pattern shows up in (Xu, 2025), where K-Means applied to online learning behavior logs produced distinguishable engagement groups, much like the gap seen here between the Passive Learner and Highly Consistent Learner clusters. The main difference is the data source: version control history is arguably a more direct trace of actual work than LMS click-stream data, which tends to reflect platform login time more than genuine output.

The split between productivity and consistency observed in this dataset also lines up with (Delgado et al., 2021), who found using Self-Organizing Maps that engagement intensity and engagement regularity do not necessarily move together. The Productive Learner cluster here shows high commit volume paired with a large last_commit_gap, while the Highly Consistent Learner cluster shows the opposite, fewer commits overall but far more recent and evenly spread activity, a fairly direct illustration of the same separation in a software development setting.

Where this study departs from that prior work is the nature of the task itself. (Cui et al., 2024) and (Xu, 2025) both worked with generalized programming exercises or LMS interaction data, not a full framework-specific application build. The Laravel movie database project required students to work across routing, controllers, Eloquent models, and Blade views over an extended period, a more structured task than a single assignment or a stream of platform clicks. This may explain why the Passive Learner profile is not simply low activity in a generic sense, but specifically an early burst of commits followed by a long silence, consistent with a project being scaffolded and then abandoned. Whether this pattern is unique to framework-based projects is a reasonable question for future comparative work.

4. CONCLUSION

This study applied K-Means Clustering to GitHub repository activity data from 73 students in a Laravel-based Web Framework Programming course, using five behavioral features, total_commit, active_days, avg_commit_per_day, weekend_commit, and last_commit_gap, to objectively profile student project engagement at scale. The Elbow Method identified k=3 as the optimal cluster configuration, yielding a Silhouette Score of 0.4041 that confirmed the presence of meaningful partition structure. The primary contribution of this research is the demonstration that behavioral clustering of version-control activity can serve as a low-cost, scalable, and reproducible alternative to manual repository inspection for instructors managing project-based programming courses at scale. Three distinct engagement profiles emerged: Passive Learners (51 students, 69.86%) with low commit frequency and high last_commit_gap reflecting deadline-driven behavior; Productive Learners (20 students, 27.40%) with the highest commit volume and daily intensity indicating sprint-based development; and Highly Consistent Learners (2 students, 2.74%) with the most distributed active-day patterns and lowest commit gap, representing the behavioral ideal for iterative project-based development. These findings demonstrate that mandatory GitHub repository submissions can serve as an unobtrusive yet informative behavioral data source, enabling instructors to monitor individual engagement patterns without additional data collection overhead and to design targeted interventions, such as intermediate commit milestones, that promote more consistent development practices. Future work should integrate repository behavioral features with academic performance outcomes to assess predictive validity, extend the approach longitudinally across multiple cohorts, and incorporate code quality indicators to produce a more comprehensive model of student software development behavior.

REFERENCES

Alsmadi, H., Kandasamy, G., Al Kafri, A., & Zahirah, K. F. (2024). Empowering computing students through multidisciplinary project based learning (PBL): Creating meaningful differences in the real world. *Social Sciences & Humanities Open*, *10*, 101180. https://doi.org/10.1016/J.SSAHO.2024.101180

Atma, Y. A., & Montesori, S. (2022). Analisis Data Mining Untuk Menentukan Profit Perusahaan Menggunakan Metode K-Means. *Device: Journal Of Information System, Computer Science And Information Technology*, *3*(2), 29–36. https://doi.org/https://doi.org/10.46576/device.v3i2.2699


TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 1221-1231
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10084

Cakra, G., Wibowo, A., & Wardhani, J. A. (2026). Analysis of Agile Methods in Laravel-Based Information System Development : A Review. *International Journal of Informatics and Computation (IJICOM)*, *8*(1), 69–86. https://doi.org/10.35842/ijicom

Compagnucci, L., Spigarelli, F., Sernani, P., Frontoni, E., & Seri, P. (2025). A systematic literature review of business-to-business platforms for the digital transformation of the manufacturing industry: taking stock and advancing through research. *Technovation*, *148*, 103330. https://doi.org/10.1016/J.TECHNOVATION.2025.103330

Cui, J., Zhou, F., Liu, C., Jia, Q., Song, Y., & Gehringer, E. (2024). Utilizing the constrained k-means algorithm and pre-class github contribution statistics for forming student teams. *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, 569–575. https://doi.org/10.1145/3649217.3653634

Decan, A., Mens, T., & Onsori Delicheh, H. (2023). On the outdatedness of workflows in the GitHub Actions ecosystem. *Journal of Systems and Software*, *206*. https://doi.org/10.1016/j.jss.2023.111827

Delgado, S., Morán, F., San José, J. C., & Burgos, D. (2021). Analysis of students' behavior through user clustering in online learning settings, based on self organizing maps neural networks. *IEEE Access*, *9*, 132592–132608. https://doi.org/10.1109/ACCESS.2021.3115024

Lashiyanti, A. R., Rasyid Munthe, I., Nasution, F. A., & Korespondensi, E. P. (2023). Optimisasi Klasterisasi Nilai Ujian Nasional dengan Pendekatan Algoritma K-Means, Elbow, dan Silhouette. *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI*, *6*(1), 14–20. https://doi.org/10.55338/jikomsi.v6i1

Liew, Y. C., Lim, K. Y., Lim, T. Y., Tan, C. J., Chai, K. K., & Deng, X. (2024). The Effect of Data Transformation Techniques on Machine Learning Performance: A Case Study on Student Dropout Prediction. *2024 IEEE 5th International Conference on Pattern Recognition and Machine Learning (PRML)*, 1–7. https://doi.org/10.1109/PRML62565.2024.10779714

Mohd Talib, N. I., Abd Majid, N. A., & Sahran, S. (2023). Identification of Student Behavioral Patterns in Higher Education Using K-Means Clustering and Support Vector Machine. *Applied Sciences*, *13*(5). https://doi.org/10.3390/app13053267

Mulyani, H., Setiawan, R. A., & Fathi, H. (2023). Optimization of K value in clustering using silhouette score (case study: Mall customers data). *Journal of Information Technology and Its Utilization*, *6*(2), 45–50. https://doi.org/10.56873/jitu.6.2.5243

Oti, E. U., Olusola, M. O., Eze, F. C., & Enogwe, S. U. (2021). Comprehensive review of K-Means clustering algorithms. *International Journal of Advances in Scientific Research and Engineering*, *12*(08), 22–23. https://doi.org/10.31695/IJASRE.2021.34050

Papatheodosiou, P., Panagoulias, D. P., Virvou, M., Tsihrintzis, G. A., Bonakis, A., & Dikeos, D. (2024). Leveraging Artificial Intelligence for personalised insomnia-sleep calibration via the Big Five Personality Traits. *Procedia Computer Science*, *246*(C), 2539–2548. https://doi.org/10.1016/J.PROCS.2024.09.723

Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. *WIREs Data Mining and Knowledge Discovery*, *10*(3), e1355. https://doi.org/https://doi.org/10.1002/widm.1355

Rostami Mazrae, P., Decan, A., Mens, T., & Wessel, M. (2026). An empirical study of the evolution of GitHub actions workflows. *Journal of Systems and Software*, *236*, 112824. https://doi.org/10.1016/J.JSS.2026.112824

Saad, A., & Zainudin, S. (2022). A review of Project-Based Learning (PBL) and Computational Thinking (CT) in teaching and learning. *Learning and Motivation*, *78*, 101802. https://doi.org/10.1016/J.LMOT.2022.101802

Sailer, M., Ninaus, M., Huber, S. E., Bauer, E., & Greiff, S. (2024). The End is the Beginning is the End: The closed-loop learning analytics framework. *Computers in Human Behavior*, *158*, 108305. https://doi.org/10.1016/j.chb.2024.108305

Sunardi, A., & Suharjito. (2019). MVC Architecture: A Comparative Study Between Laravel Framework and Slim Framework in Freelancer Project Monitoring System Web Based. *Procedia Computer Science*, *157*, 134–141. https://doi.org/10.1016/J.PROCS.2019.08.150

Tuyishimire, E., Mabuto, W., Gatabazi, P., & Bayisingize, S. (2022). Detecting Learning Patterns in Tertiary Education Using K-Means Clustering. *Information*, *13*(2). https://doi.org/10.3390/info13020094

Vardakas, G., Papakostas, I., & Likas, A. (2026). Deep clustering using the soft silhouette score: Towards compact and well-separated clusters. *Machine Learning*, *115*(4), 81. https://doi.org/10.48550/arXiv.2402.00608

Xu, M. (2025). Cluster Analysis of Online Learning Behavior Based on K-Means. *2025 IEEE International Conference on Electronics, Energy Systems and Power Engineering (EESPE)*, 1552–1555. https://doi.org/10.1109/EESPE63401.2025.10987550