# Anomaly Detection with Isolation Forest and Segmentation with K-means Clustering

A Brief Introduction to Unsupervised Learning in Student Data Analysis

Saugat The Great

December 1, 2025

# Presentation Outline

1. Outline
2. Abstract and Introduction

   Project Overview
3. Data Handling and Cleaning

   Data Overview and Filtering
4. Feature Engineering

   Feature Creation and Standardization
5. Isolation Forest for Anomaly Detection

   Theory, Math, and Implementation
6. K-means Clustering

   Algorithm and Evaluation
7. Conclusion

   Summary and Future Work

# Abstract

The presentation explores the application of unsupervised machine learning techniques to analyze student interaction data.

- The workflow focused on robust data preprocessing (handling, cleaning, and feature extraction) to prepare the dataset.
- Anomaly Detection was performed using the highly effective Isolation Forest algorithm to identify irregular student behaviors.
- Segmentation was achieved with K-means Clustering to group students into $\mathbf{K = 4}$ distinct behavioral clusters for targeted intervention and analysis.

# Project Focus and Methodology

Project Focus: This project details the comprehensive process of data ingestion, cleaning, transformation, and analysis. The core objective is the implementation of Anomaly Detection and data segmentation.

Methodology: The workflow emphasizes robust data preprocessing (handling, cleaning, and manipulation) to prepare the dataset before applying the machine learning models.

# Key Technologies and Libraries Used (Part 1/2)

Key Technologies and Libraries:

1. NumPy: Essential for high-performance numerical computation and serving as the foundational internal data structure for array manipulation.
2. Pandas: Used for efficient data loading, handling, and analysis via DataFrames.
3. Matplotlib: Primary library for generating static and interactive visualizations.

# Key Technologies and Libraries Used (Part 2/2)

Key Technologies and Libraries:

4. Seaborn: Statistical visualization library; provides a high-level interface for attractive statistical graphics.
5. Scikit-learn (sklearn): Comprehensive library for implementing the machine learning algorithms, including Isolation Forest and K-means.

# Data Handling and Cleaning

# Data Overview and Initial Preparation

Source of the Data

- The data is collected from UCI Machine Learning Repository.
- Initial total number of individual interactions: **9546**.
- The dataset contains **8** columns, capturing student country, question metadata (ID, level, topic, subtopic, keywords), and interaction outcome (`is_correct`).

Data Information:

- Handling Missing Values: A full check revealed no missing values, thus no imputation techniques were necessary.
- Column Renaming: Columns were renamed for enhanced readability and ease of programming, e.g., 'Student ID' $\rightarrow$ 'student_id'.

# Duplicate Data Filtering

Handling Duplicate Values

- **2083** rows were found to be identical instances of data.
- Rationale: Given the nature of a single student interaction being logged, identical rows likely represent data collection errors rather than simultaneous events. These duplicates were removed.
- Final Interaction Count $= 9546 - 2083 =$ **7463** interactions remaining.

# Low-Activity Student Filtering

Filtering Low-Activity Students

- Students with fewer than a minimum number of interactions (**MIN_ATTEMPTS = 5**) were removed from the analysis.
- Rationale: Students with very few attempts do not provide enough data to form reliable behavioral features, and their inclusion could skew the clustering and anomaly detection processes.

This ensures the final dataset for feature engineering represents only students with meaningful activity.

# Feature Engineering

# Feature Engineering: Step 1 - Aggregation

The raw interaction data (long format) was aggregated to create a single feature vector (wide format) for each unique student ($\sim 372$ students), forming the basis for analysis.

Key Features Extracted (**FEATURE_COLS**):

- Overall Correct Ratio (**CR**): Measures general student performance.

$$\mathbf{CR} = \frac{\sum \text{Correct Attempts}}{\sum \text{Total Attempts}}$$

- Total Attempts: Student's overall volume of engagement.

This process transforms the focus from individual attempts to student-level behavior.

# Feature Engineering: Step 2 - Diversity Metrics

Key Features Extracted (Continued):

- Unique Diversity Metrics: Count of unique Question IDs, Topics, Subtopics, and Keywords attempted. Measures the breadth of student engagement.

Rationale for Feature Engineering:

- The clustering and anomaly detection algorithms require a single, rich profile for each student, not a long list of individual attempts.
- These aggregated features (Correct Ratio, Attempts, Diversity) capture the fundamental dimensions of student behavior: performance, volume, and scope.

# Standardization for Model Readiness

Machine learning algorithms like K-means and Isolation Forest are highly sensitive to the scale of features. Features with a larger magnitude (e.g., Total Attempts) would unfairly dominate the distance calculations.

Technique: Standard Scaling (Z-score Normalization)

- The features were standardized to have a mean of **0** and a standard deviation of **1**.
- For a feature value $x$, the standardized value $z$ is:

$$z = \frac{x - \mu}{\sigma}$$

  Where $\mu$ is the mean and $\sigma$ is the standard deviation of the feature across all students.

This ensures all features contribute equally to the distance metrics used in the subsequent algorithms.

# Isolation Forest for Anomaly Detection

# Isolation Forest: Detecting Anomalous Student Behavior

Goal: Identify students whose feature vectors ($\mathbf{x} \in \mathbb{R}^d$) are significantly different from the majority.

Concept: Unlike distance-based methods that try to define a 'normal' region, Isolation Forest (iForest) works by explicitly isolating anomalies.

Mechanism: iForest is an ensemble of Isolation Trees (iTrees).

- It randomly selects a feature and a split value, recursively partitioning the data.
- Anomalies are few and far from the bulk of the data, meaning they are easier to isolate and require fewer random partitions (splits) to reach a terminal node.
- Normal points are numerous and require more splits to be isolated.

# iForest: Path Length and Anomaly Score Formula

The measure of abnormality is based on the Path Length $\mathbf{h}(\mathbf{x})$ from the root of an iTree to the terminal node containing the instance $\mathbf{x}$.

1. Average Path Length ($\mathbf{E}[\mathbf{h}(\mathbf{x})]$):

   - The path length is averaged across all iTrees in the forest.
   - A shorter average path length indicates an anomaly (easier to isolate).

2. Anomaly Score Formula ($\mathbf{s}(\mathbf{x})$): The raw path length is converted into an anomaly score $s(x) \in [0, 1]$:

$$s(x) = 2^{-\frac{E[h(x)]}{c(n)}}$$

Where $\mathbf{c}(\mathbf{n})$ is a normalization factor related to the sample size $\mathbf{n}$.

# Interpreting the Anomaly Score

Interpretation of Score ($\mathbf{s}(\mathbf{x})$):

- $s(x) \approx 1$: Indicates an anomaly (very short average path length).
- $s(x) < 0.5$: Indicates a normal instance (longer average path length).
- $s(x) \approx 0.5$: Indicates no distinct anomaly.

Detection Strategy: Instances with scores above a chosen threshold are flagged as anomalies.

Implementation Detail (Contamination): The model was run with a predefined **contamination** rate of **0.03** (**3%**). This parameter implicitly sets the threshold to classify the top 3% highest-scoring instances as anomalies.

# K-means Clustering

# K-means Clustering: Segmenting Student Behavior

Goal: Partition the student feature vectors into **K** groups (clusters), where each student belongs to the cluster with the nearest mean (centroid).

The K-means Algorithm Steps:

1. Initialization: Select **K** initial cluster centroids (randomly or using **k++** initialization).
2. Assignment (E-Step): Assign each student vector **x** to the cluster $\mathbf{S_j}$ whose centroid $\mu_\mathbf{j}$ is the closest (using $\mathbf{L_2}$ Euclidean distance).
3. Update (M-Step): Recalculate the centroid $\mu_\mathbf{j}$ for each cluster $S_j$ as the mean of all points assigned to it.
4. Convergence: Repeat steps 2 and 3 until the centroids no longer change or a maximum number of iterations is reached.

# Mathematical Objective: Within-Cluster Sum of Squares (WCSS)

K-means minimizes the Within-Cluster Sum of Squares (WCSS), also known as Inertia. This ensures clusters are compact.

$$J = \sum_{j=1}^{K} \sum_{\mathbf{x} \in S_j} \|\mathbf{x} - \mu_{\mathbf{j}}\|^2$$

Where:

- $J$ is the Inertia (WCSS).
- $K$ is the number of clusters.
- $\mathbf{x}$ is a data point (student feature vector).
- $\mu_{\mathbf{j}}$ is the centroid of cluster $S_j$.
- $\| \dots \|^2$ is the squared Euclidean distance.

# Selecting K and Evaluating Cluster Quality

Cluster Number Selection (**K**):

- The Elbow Method (plotting WCSS vs. **K**) is used to find the "bend" point where diminishing returns are observed.
- 
- For this analysis, $\mathbf{K = 4}$ was chosen based on combining the Elbow Method result with the highest Silhouette Score for a robust and interpretable set of segments.

Evaluation Metric: Silhouette Score (**s**) The score measures how similar an object is to its own cluster compared to other clusters.

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

Where **a** is the mean intra-cluster distance and **b** is the mean nearest-cluster distance. Scores closer to $+\mathbf{1}$ indicate dense, well-separated clusters.

# Interpreting the K = 4 Student Clusters

The final step is to interpret the meaning of the clusters by analyzing their Centroids (mean feature values) for the **K = 4** segments.

Example Cluster Characteristics (based on centroid analysis):

- Cluster 0: The Engaged High Achievers (High **CR**, very high Diversity)
- Cluster 1: The Struggling but Persistent (Low **CR**, high Total Attempts)
- Cluster 2: The Focused Low Engagers (Average **CR**, low Diversity and Total Attempts)
- Cluster 3: The Efficient High Achievers (Very high **CR**, moderate Attempts)

# Conclusion

# Summary of Findings

- The data was successfully preprocessed, resulting in a clean dataset of **7463** interactions from robust students.
- Isolation Forest effectively flagged a small fraction (**3%**) of students as anomalous, suggesting they exhibit highly irregular behavior compared to the norm.
- K-means Clustering successfully segmented the students into **4** distinct behavioral groups, which can be used to tailor educational strategies (e.g., targeted assistance for struggling clusters).

Future Work:

- Explore DBSCAN or OPTICS clustering for natural density-based groups.
- Incorporate temporal features (e.g., time between attempts) into the feature set.