

# Overview & Guide to an Automated Grading System (AGS)

## 1. Overview of an Automated Grading System (AGS)

An Automated Grading System (AGS) is a software solution that uses artificial intelligence (AI), machine learning (ML), and natural language processing (NLP) to automatically grade student assignments, essays, and tests.

## 2. Benefits and Features

### Benefits:

- Time-saving: Automates grading to reduce manual effort.
- Consistency: Provides standardized evaluations, reducing human bias.
- Instant Feedback: Offers immediate responses to help students learn from mistakes.
- Scalability: Handles large volumes of submissions, ideal for large classes or online courses.

### Key Features:

- Assignment Upload: Secure portals for student submissions.
- Grading Algorithms: Predefined criteria and rubrics drive the grading logic.
- Rubric-based Scoring: Allows partial credit and nuanced evaluation.
- Feedback Generation: Detailed feedback on strengths and weaknesses.
- Grade Tracking: Maintains records of student performance over time.

## 3. Techniques and Technologies

### Machine Learning & AI:

- Transformer Models: BERT, GPT, MathBERT, etc.
- Deep Learning Frameworks: TensorFlow and PyTorch.
- Natural Language Processing (NLP): Processes and understands text.
- Computer Vision & OCR: Converts scanned or handwritten work to text.

### Data Augmentation & In-Context Learning:

- Pseudo-labeling: Generates additional training data.
- In-context Learning: Provides graded examples and detailed prompts.

## 4. Challenges and Limitations

### Common Limitations:

- Contextual Understanding: AI may misinterpret nuanced or creative responses.
- Multiple Correct Approaches: Difficulty recognizing all valid solution paths.
- Handwritten Input: OCR errors can affect accuracy.
- Unit and Formatting Variability: Minor deviations may lead to misgrading.
- Bias and Data Quality: Model accuracy relies heavily on high-quality, diverse data.
- Overfitting: Model might perform well on training data but poorly on new assignments.

## 5. Steps to Build and Train an AGS

### Step 1: Data Collection & Preprocessing

- Data Sourcing: Collect graded assignments from LMSs, public datasets (e.g., ASAP, NAEP), MOOCs, or open-source educational repositories.
- Annotation & Rubric Development: Develop a clear grading rubric with criteria and partial credit guidelines.
- Preprocessing: Clean data (tokenize text, convert handwritten answers using OCR), and split into training and testing sets.

Estimated Time: ~2–3 months

### Step 2: Model Architecture Selection

- Research transformer models (BERT, GPT, MathBERT) and domain-specific variants.