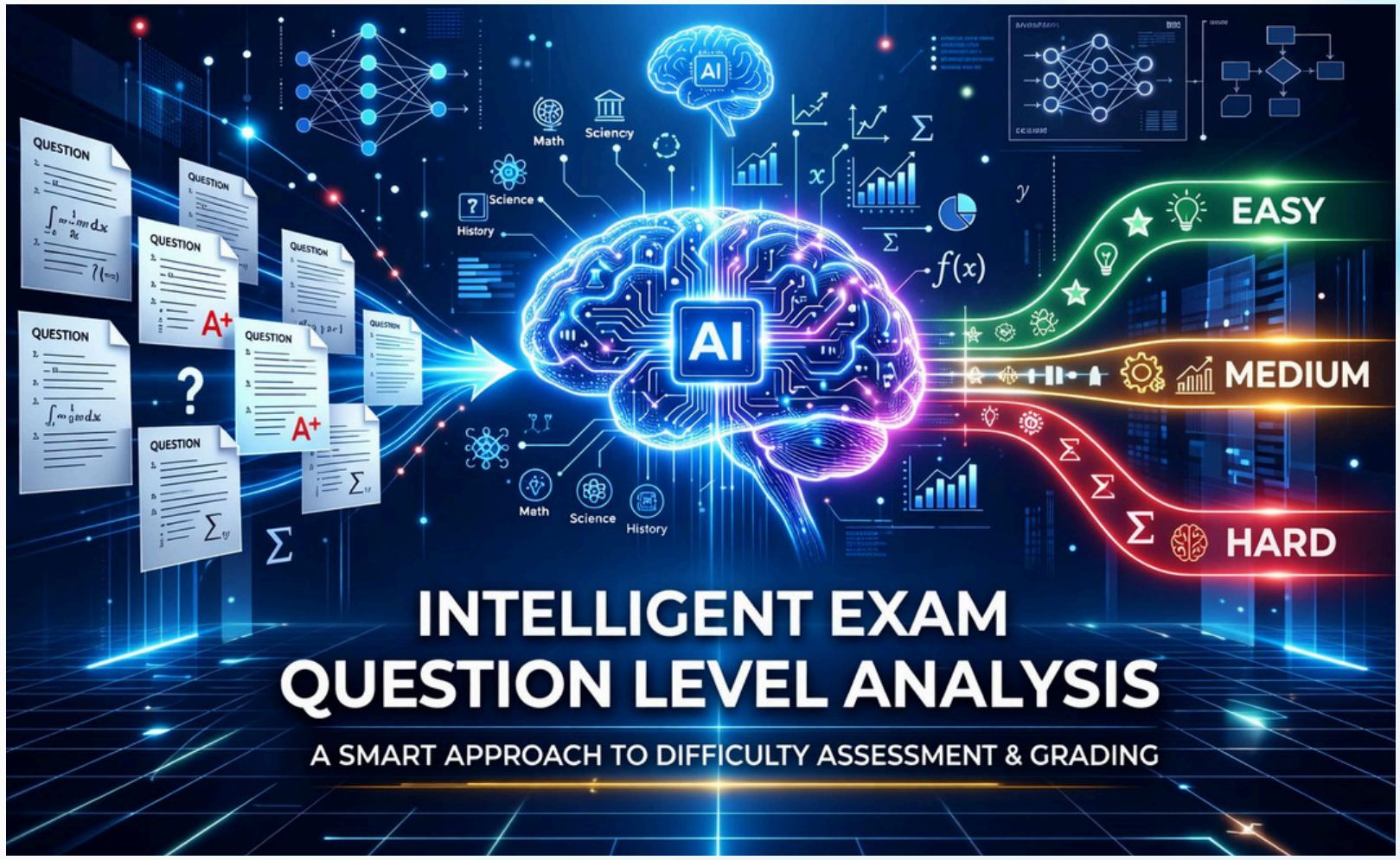


Intelligent Exam Question Level Analysis

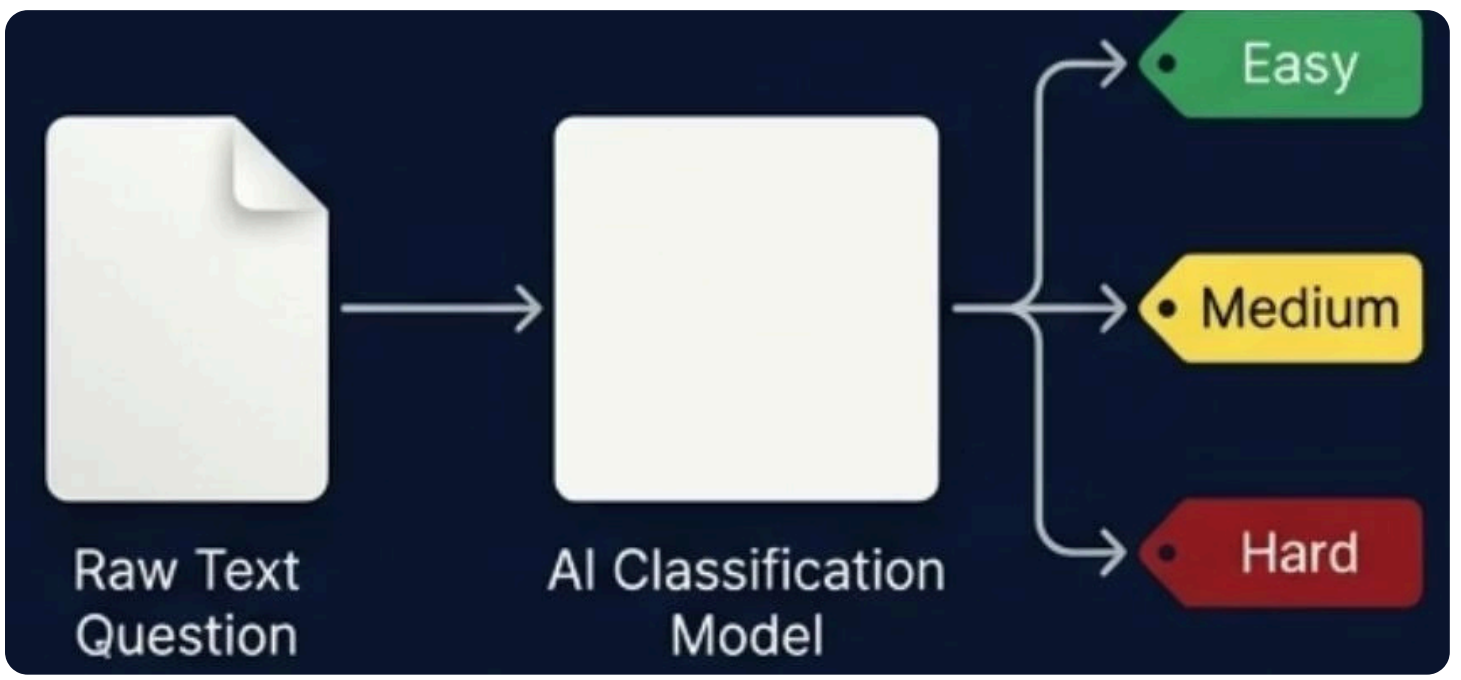
Nishant Ranjan Singh
2401010301

Atanu Adikari
2401010111

Prince Kumar Singh
2401010353

Sambhav Kumar
2401010409

Automating Cognitive Difficulty Assessment: A Machine Learning Approach



- Objective: Develop a reliable machine learning pipeline to automatically predict exam question difficulty (Easy, Medium, Hard).
- Core Challenge: Manual classification is highly subjective, lacks standardization, and struggles to scale across large question
- banks.
- Proposed Solution: An AI-driven evaluation system leveraging classical ML and NLP to align question complexity with
- Bloom's Taxonomy.
- Primary Outcome: Achieved viable automated cognitive assessment deployed via a real-time web application.

The diagram illustrates the machine learning pipeline. It starts with a 'Raw Text Question' represented by a document icon, which
is input into an 'AI Classification Model' represented by a rectangular box. The model then outputs three difficulty levels: 'Easy'
(green tag), 'Medium' (yellow tag), and 'Hard' (red tag), each represented by a colored tag with a hole. Flowchart of the AI Classification Model pipeline.

Baseline System Accuracy (Manual/Subjective)

0%

Proposed Model Accuracy (Logistic Regression)

71.00%

Dataset Characteristics and Exploratory Data Analysis (EDA)

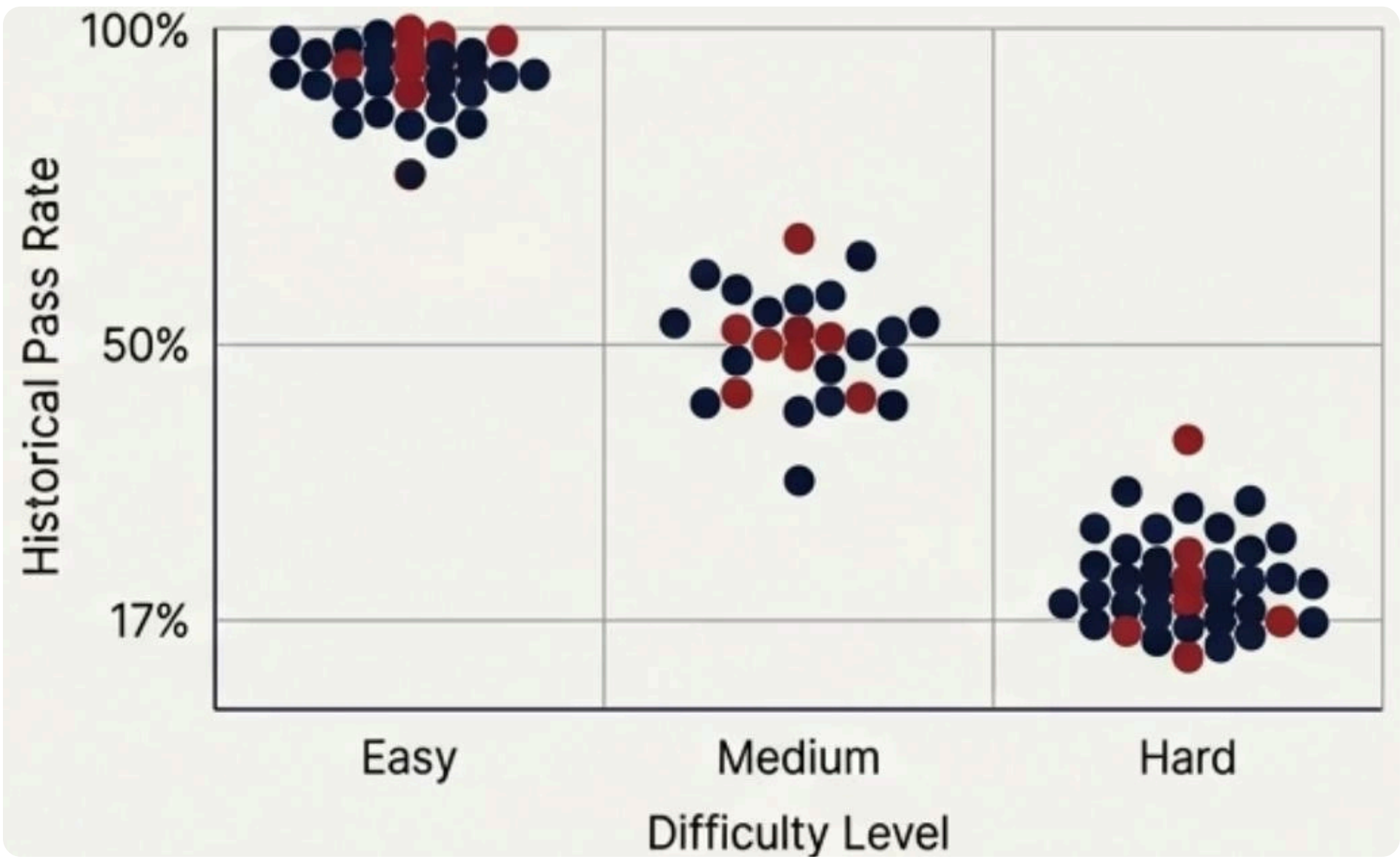
Composition: Multi-disciplinary corpus of 6,200 initial records covering Physics, ComputerScience,and Mathematics.

Feature Diversity: Integrates textual data, Bloom's Taxonomy levels, and historical student performance metrics.

Question_Text	**Bloom_Taxonomy**
Subject_Domain	**Historical_Pass_Rate**
Topic_Subdomain	**Difficulty_Level**

Key EDA Insight 1: Strong negative correlation exists between historical pass rates and 'Hard' classifications.

Key EDA Insight 2: Higher difficulty strongly correlates with increased text length and domain-specific vocabulary (e.g., 'derive', 'evaluate').

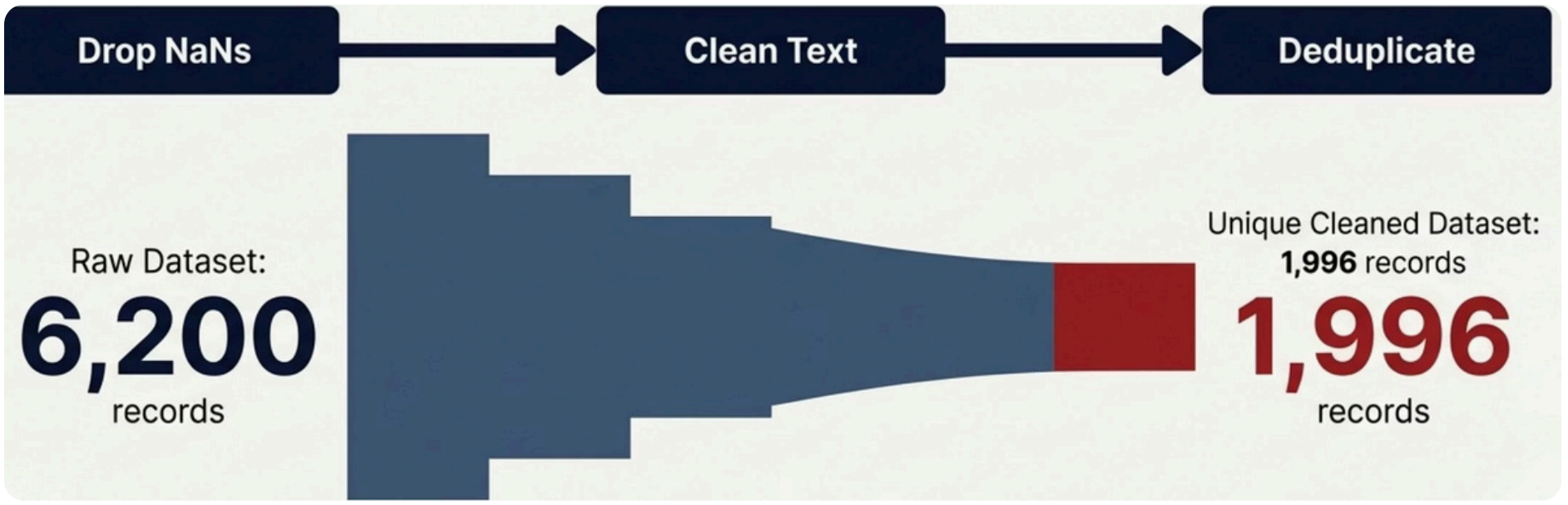


The scatter plot illustrates the relationship between the Historical Pass Rate (Y-axis, ranging from 17% to 100%) and the Difficulty Level (X-axis, categorized as Easy, Medium, and Hard). The data points, represented by blue and red circles, show a clear negative correlation. In the 'Easy' category, the pass rates are high, mostly above 70%. In the 'Medium' category, the pass rates are more varied, ranging from approximately 30% to 70%. In the 'Hard' category, the pass rates are significantly lower, mostly below 40%.

Scatter plot showing Historical Pass Rate vs Difficulty Level. The plot is divided into three vertical sections: Easy, Medium, and Hard. The y-axis represents Historical Pass Rate from 17% to 100%. Data points are colored blue and red. In the Easy section, points are clustered between 70% and 100%. In the Medium section, points are clustered between 30% and 70%. In the Hard section, points are clustered between 17% and 40%.

Sequential Data Preprocessing and Standardization

- **Missing Value Handling:** Strict removal of records lacking essential target or text data to preserve training integrity.
- **Text Normalization:** Conversion to lowercase, stripping of HTML artifacts, and punctuation removal using regex.
- **Deduplication:** Aggressive removal of duplicate questions to prevent data leakage and model overfitting.



A flowchart showing the sequential data preprocessing steps: Drop NaNs, Clean Text, and Deduplicate. Below the flowchart is a funnel-shaped diagram illustrating the reduction in dataset size from 6,200 records to 1,996 records.

Raw Dataset

6,200

records

Unique Cleaned Dataset

1,996

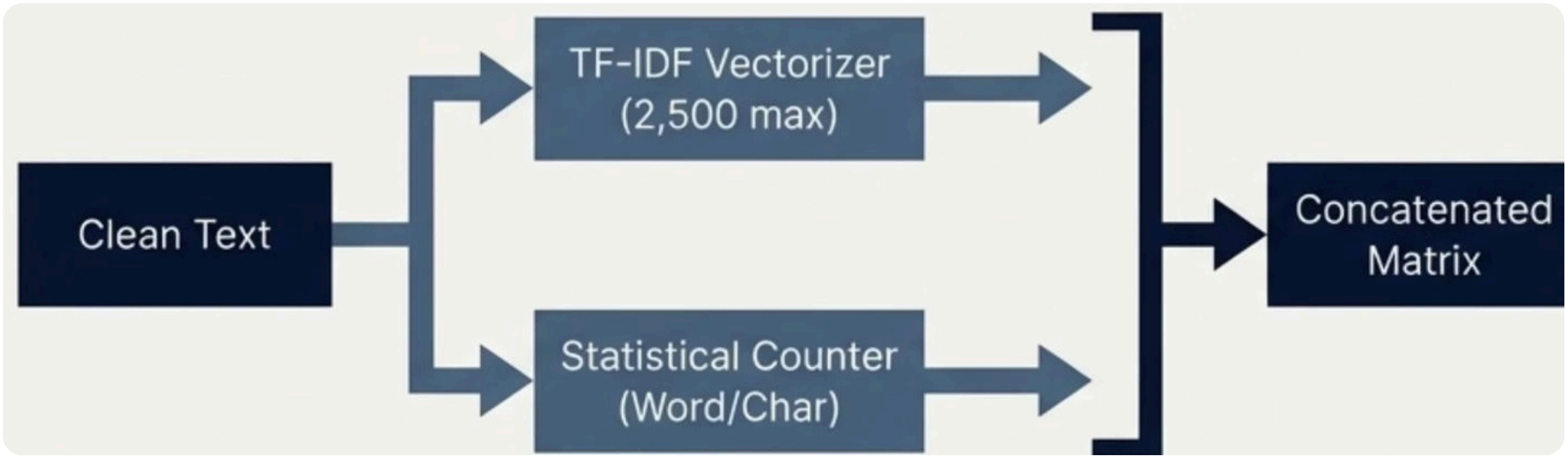
records

Step	Records
Raw Dataset	6,200
Drop NaNs	~5,500
Clean Text	~4,500
Deduplicate	1,996

Feature Engineering: Bridging Lexical and Statistical Spaces

TextVectorization: TF-IDFimplemented with a 2,500 maximum feature limit and unigram/bigram(1,2)extraction.

Semantic Filtering: Standard English stop words removed to isolate discriminative academic vocabulary. **Statistical Metrics:** Word count and character length extracted to capture surface-level complexity. **Dimensionality Alignment:** Statistical features normalized via MaxAbsScaler to maintain sparsity compatibility with TF-IDF
arrays.



The diagram illustrates the feature engineering process. It starts with a 'Clean Text' block, which branches into two parallel processing paths. The top path uses a 'TF-IDF Vectorizer (2,500 max)' to process the text. The bottom path uses a 'Statistical Counter (Word/Char)' to extract word and character-level metrics. The outputs of these two paths are combined into a 'Concatenated Matrix'.

Flowchart of feature engineering process

Final Feature Matrix

(Shape: 1996 x 539)

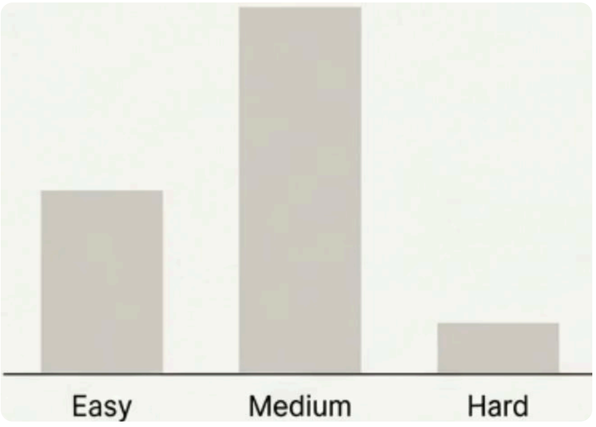
A vertical bar representing the Final Feature Matrix. The bar is divided into two sections: a small red section at the top labeled 'Statistical Features: 2' and a large blue section below it labeled 'TF-IDF Features: 537'. The total height of the bar represents the shape 1996 x 539.

Vertical bar representing the Final Feature Matrix

Target Encoding and Synthetic Minority Over-sampling

- **Target Transformation:** Categorical labels encoded to integers (Easy: 0, Hard: 1, Medium: 2). **The Imbalance**
- **Challenge:** Educational datasets inherently skew toward specific difficulty distributions, risking majority-class bias.
- **SMOTE Implementation:** Generated synthetic examples for minority classes by interpolating between existing high-dimensional samples.
- **Training Integrity:** Stratified 80/20 train-test split applied prior to SMOTE to ensure the test set represents real-world
- distributions.

Pre-SMOTE Class Distribution

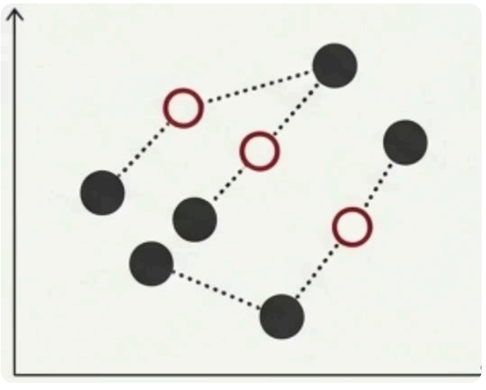


A bar chart with three categories on the x-axis: Easy, Medium, and Hard. The bars are light gray. The Medium bar is significantly taller than the Easy and Hard bars, indicating a majority class bias.

Category	Count
Easy	Medium
Medium	High
Hard	Low

Bar chart showing the pre-SMOTE class distribution for Easy, Medium, and Hard categories.

SMOTE Interpolation



A 2D scatter plot showing the SMOTE interpolation process. It features several solid black circles representing existing data points and several hollow red circles representing synthetic data points. Dotted lines connect the black circles to the red circles, illustrating the interpolation between minority class samples.

Diagram illustrating SMOTE interpolation between existing data points to generate synthetic ones.

Post-SMOTE Training Distribution

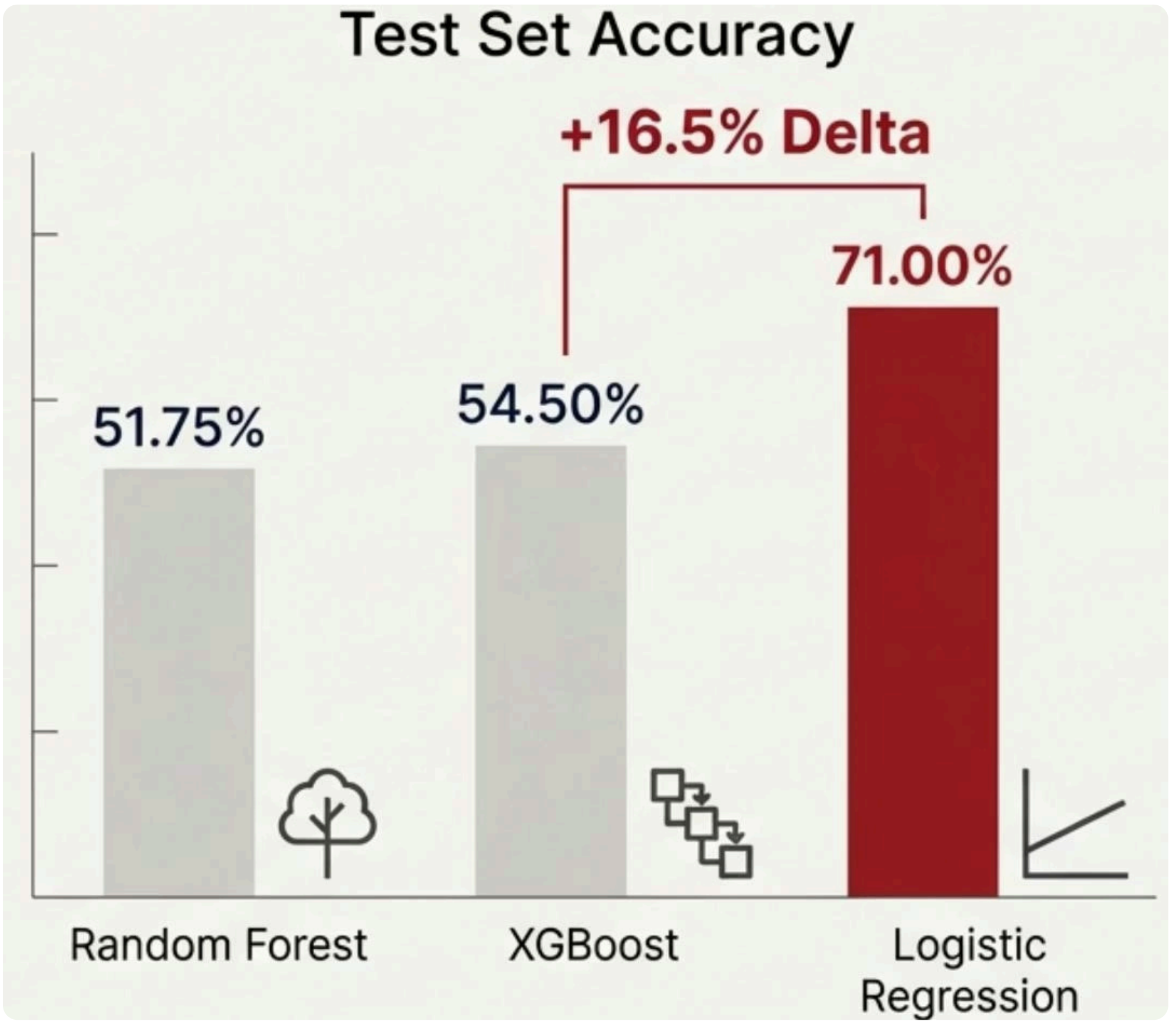
A bar chart with three categories on the x-axis: Easy, Medium, and Hard. The bars are dark blue and all three bars are of equal height, indicating that the SMOTE process has balanced the class distribution.

Category	Count
Easy	High
Medium	High
Hard	High

Bar chart showing the post-SMOTE training distribution for Easy, Medium, and Hard categories.

Algorithm Evaluation and Model Selection

- **Logistic Regression:** Evaluated for its maximum entropy principle in multi-class text classification.
- **Random Forest:** Tested as a bagging ensemble approach to handle non-linear feature interactions.
- **XGBoost:** Deployed as a gradient boosting framework for sequential weak learner optimization.
- **Selection Criteria:** Automated objective selection based strictly on maximum generalization accuracy on the holdout test set.



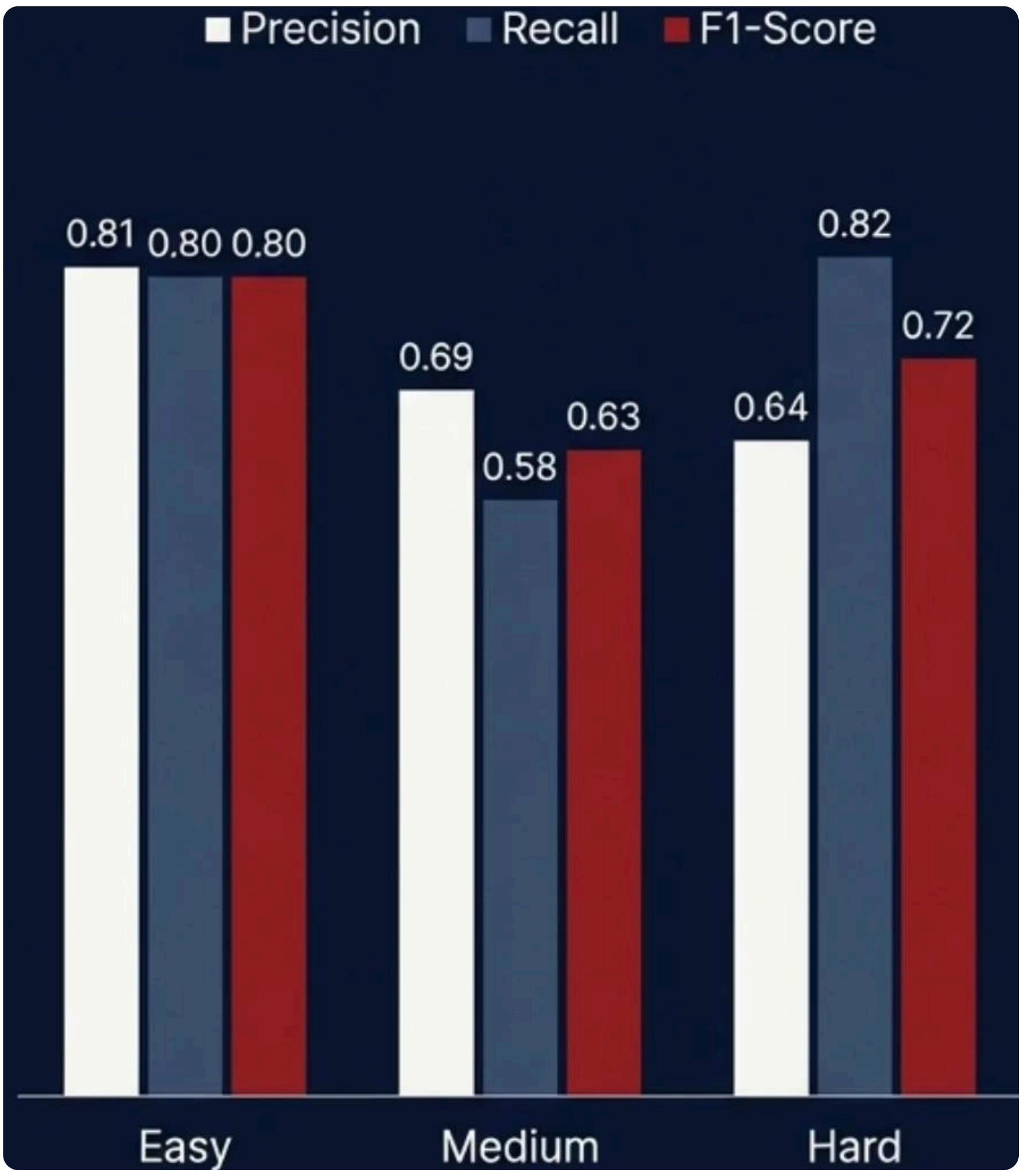
Test Set Accuracy

A bar chart titled 'Test Set Accuracy' comparing the performance of three models. The y-axis represents accuracy percentage. The x-axis lists the models: Random Forest, XGBoost, and Logistic Regression. Each bar is accompanied by an icon: a tree for Random Forest, a staircase for XGBoost, and a line graph for Logistic Regression. The accuracy values are 51.75% for Random Forest, 54.50% for XGBoost, and 71.00% for Logistic Regression. A red bracket connects the top of the XGBoost bar to the top of the Logistic Regression bar, labeled '+16.5% Delta'.

Model	Test Set Accuracy
Random Forest	51.75%
XGBoost	54.50%
Logistic Regression	71.00%

Bar chart showing Test Set Accuracy for three models: Random Forest (51.75%), XGBoost (54.50%), and Logistic Regression (71.00%). A red bracket indicates a +16.5% Delta between XGBoost and Logistic Regression.

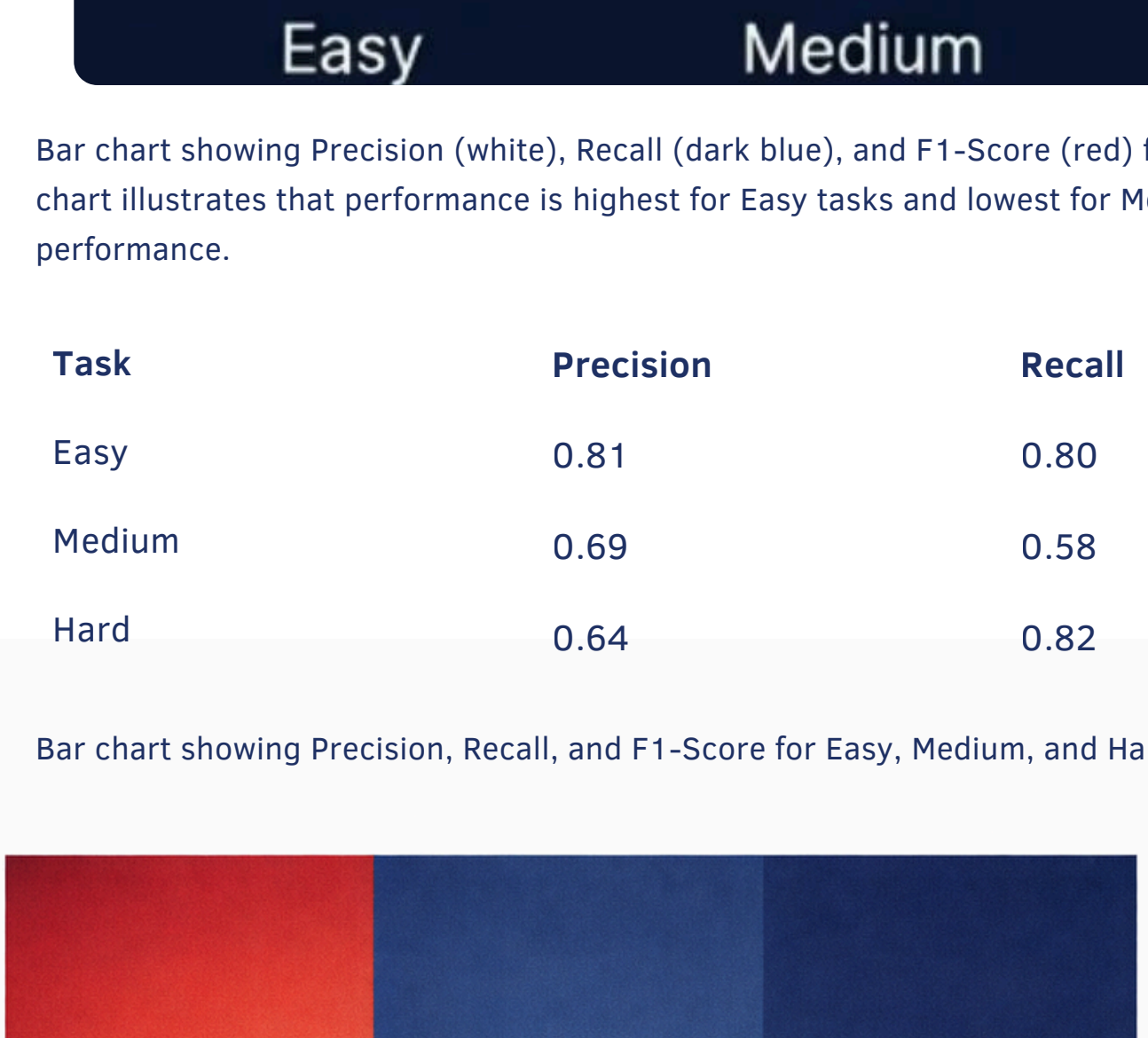
Champion Model Deep-Dive: Logistic Regression Performance



Bar chart showing Precision (white), Recall (dark blue), and F1-Score (red) for Easy, Medium, and Hard classification tasks. The chart illustrates that performance is highest for Easy tasks and lowest for Medium tasks, with Hard tasks showing intermediate performance.

Task	Precision	Recall	F1 Score
Easy	0.81	0.80	0.80
Medium	0.69	0.58	0.63
Hard	0.64	0.82	0.72

Bar chart showing Precision, Recall, and F1-Score for Easy, Medium, and Hard classification tasks.



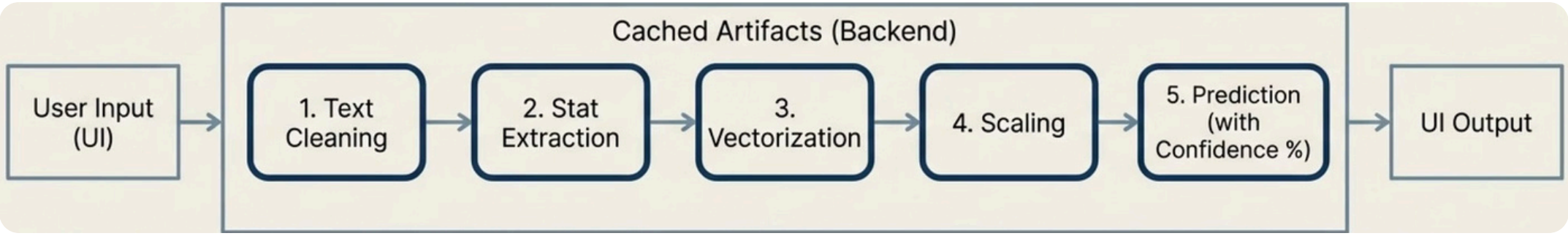
Confusion matrix heatmap showing classification performance across three tasks (Easy, Medium, Hard). The diagonal elements represent correct classifications, while off-diagonal elements represent misclassifications. The color scale indicates the magnitude of the values, with darker red representing higher values and darker blue representing lower values.

Confusion matrix heatmap showing classification performance across three tasks.

- **Easy Classification:** Excellent performance (F1: 0.80); distinct lexical patterns for basic recall are easily identified.
- **Hard Classification:** High recall (0.82) ensures genuinely difficult questions are caught, despite some false positives.
- **The Medium Boundary:** Lowest performance (F1: 0.63) reflects the inherent cognitive ambiguity between 'Apply' and 'Evaluate' thresholds.
- **Inherent Subjectivity:** 71% accuracy likely approaches the human inter-rater reliability ceiling for subjective difficulty as s es sment.

Deployment: Real-Time Web Application Architecture

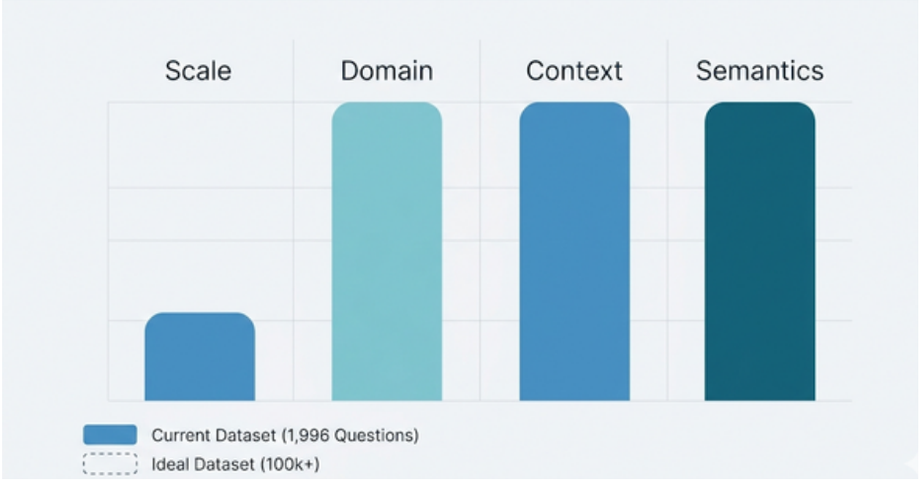
- **Model Artifact Caching:**Pre-trained vectorizer,scaler,and best model loaded via `@st.cache_resource` for low-latency inference.
- **1-to-1 Pipeline Alignment:** Inference strictly mirrors training.
- **Confidence Scoring:** Provides maximum probability distribution scores alongside difficulty predictions for user transparency.
- **Accessibility:** Deployed via Streamlit, requiring minimal compute overhead for institutional adoption.



A flowchart illustrating the real-time web application architecture. It starts with 'User Input (UI)' on the left, which feeds into a large box labeled 'Cached Artifacts (Backend)'. Inside this box, there is a sequence of five steps: '1. Text Cleaning', '2. Stat Extraction', '3. Vectorization', '4. Scaling', and '5. Prediction (with Confidence %)'. Arrows indicate the flow from one step to the next. The final step outputs to 'UI Output' on the right.

Methodological Constraints and Dataset Limitations

- **Dataset Scale:** 1,996 unique questions may not fully encapsulate cross-domain linguistic diversity.
- **Domain Specificity:** Current models are constrained to STEM (Physics, CS, Math); generalizability to humanities is untested.
- **Context Independence:** Fails to account for external difficulty variables like syllabus placement or prerequisite knowledge.
- **Semantic Shortcomings:** TF-IDF captures lexical frequency but misses deep syntactic complexity and logical structure.

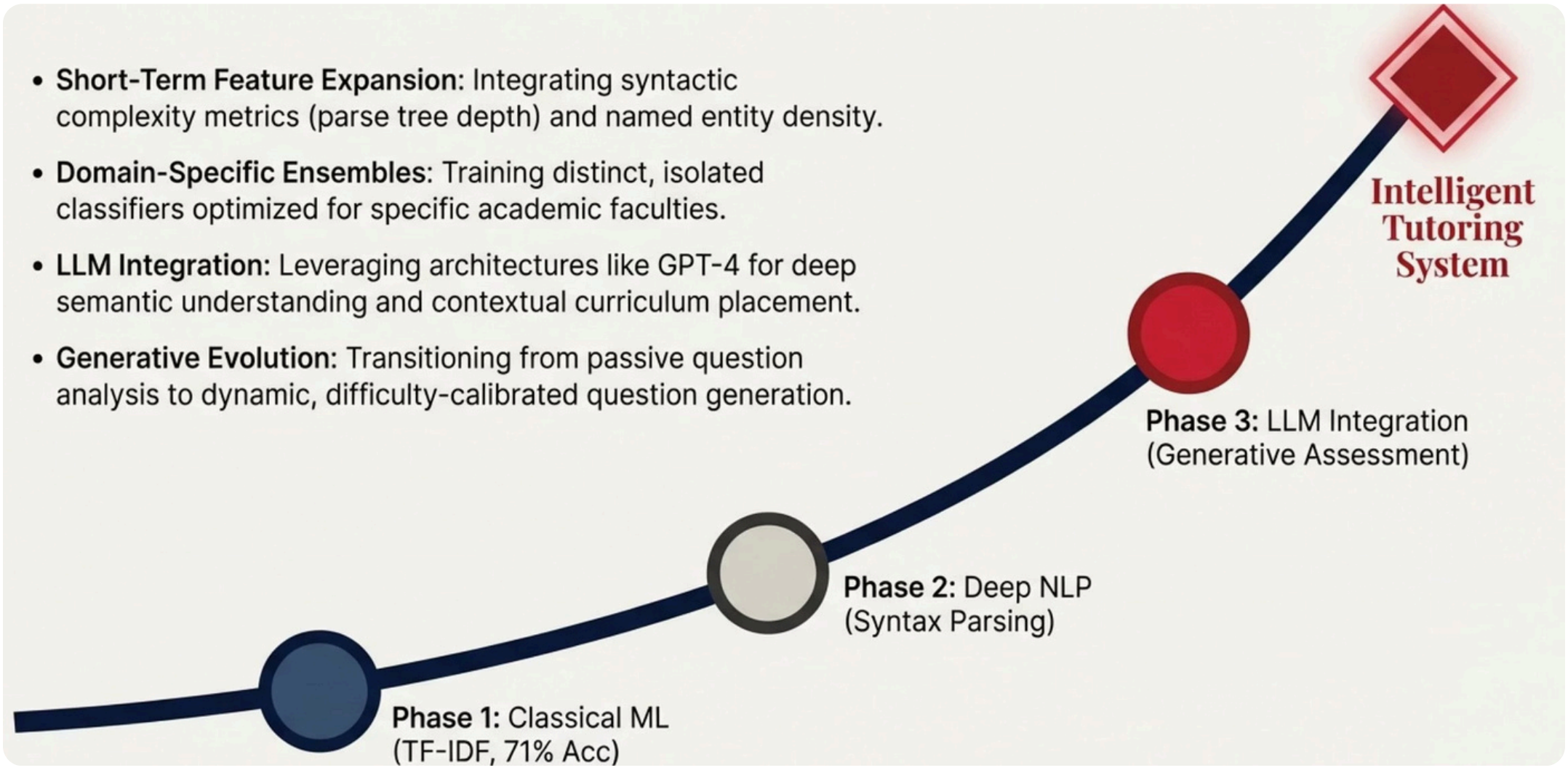


The figure consists of four vertical bars of equal height, each labeled at the top. The labels are **Scale**, **Domain**, **Context**, and **Semantics**. The **Scale** bar is the shortest and contains a legend at its base. The legend shows a small blue bar labeled 'Current: 1,996' and a longer blue bar labeled 'Ideal Dataset (100k+)'. The other three bars (**Domain**, **Context**, and **Semantics**) are all the same height and are significantly taller than the **Scale** bar.

A diagram showing four vertical bars representing constraints: Scale, Domain, Context, and Semantics. The Scale bar is the shortest and includes a legend with 'Current: 1,996' and 'Ideal Dataset (100k+)`.

Roadmap: From Classical ML to Generative AI Integration

- **Short-TermFeatureExpansion:** Integrating syntactic complexity metrics(parse treedepth) and named entity density.
- **Domain-Specific Ensembles:** Training distinct, isolated classifiers optimized for specific academic faculties.
- **LLM Integration:** Leveraging architectures like GPT-4 for deep semantic understanding and contextual curriculum placement.
- **Generative Evolution:** Transitioning from passive question analysis to dynamic, difficulty-calibrated question generation.



The diagram illustrates a progression from Classical ML to an Intelligent Tutoring System through three phases:

- **Phase 1: Classical ML (TF-IDF, 71% Acc)**
- **Phase 2: Deep NLP (Syntax Parsing)**
- **Phase 3: LLM Integration (Generative Assessment)** The final destination is the **Intelligent Tutoring System**. A roadmap diagram showing the progression from Classical ML to an Intelligent Tutoring System through three phases: Classical

  ML, Deep NLP, and LLM Integration.