Detection of AI-Generated Arabic Text

A Data Mining Approach

MSIS-822 – Advanced Data Analytic Techniques

Ahmad Silmi Alofi
Student Number : 4714243

Final Course Graduation Project

December 14, 2025

Abstract

This project addresses the task of detecting AI-generated Arabic text using a complete data mining pipeline. The study uses the *KFUPM-JRCAI/arabic-generated-abstracts* dataset and follows a structured workflow: data acquisition from Hugging Face, Arabic-specific preprocessing and exploratory analysis, feature engineering using TF–IDF combined with assigned stylometric features, and supervised model training with rigorous evaluation. Traditional machine learning models (Naïve Bayes, Logistic Regression, and Linear SVM) are compared using metrics suitable for imbalanced data, including balanced accuracy and ROC–AUC. Results show that linear models augmented with stylometric features achieve very high performance on the held-out test set.

1. Introduction

The recent progress in large language models has increased the volume of automatically generated text, creating practical needs for detection systems in areas such as academic integrity, authorship verification, and misinformation mitigation. Although modern generators can produce fluent Arabic, generation often leaves measurable stylistic traces. This project builds and evaluates a supervised binary classifier to distinguish between **human-written** and **AI-generated** Arabic abstracts, and provides an analysis of which features contribute most to the decision.

2. Dataset Description

Experiments are conducted using the Hugging Face dataset *KFUPM-JRCAI/arabic-generated-abstracts*, which provides human Arabic abstracts and multiple AI-generated variants. The dataset is organized into three generation subsets: *by_polishing*, *from_title*, and *from_title_and_content*. Each record includes an `original_abstract` (human) and model-specific generated fields (e.g., `openai_generated_abstract`). For classification, data were consolidated into a single table with columns `text`, `label`, and `source`, where label 0 represents **HUMAN** and label 1 represents **AI**. Because the dataset does not provide predefined train/validation/test splits, a stratified split of 70/15/15 was created *before* preprocessing to avoid data leakage.

3. Methodology

3.1 Phase 1: Setup and Data Acquisition

The dataset was loaded directly using the `datasets` library from Hugging Face. Each sample was expanded into one human instance and multiple AI instances (one per generator), producing a binary classification dataset with the target label.

3.2 Phase 2: Arabic Preprocessing and EDA

Two preprocessing versions were maintained:

- **Light normalization for EDA** (`text_eda`): keeps readability while removing noise.
- **Modeling-ready cleaning** (`text_clean`): includes stemming to improve lexical generalization.

EDA computed descriptive statistics such as average word length, average sentence length (in words), and type-token ratio (TTR), and inspected class differences using TF–IDF n-grams and punctuation counts.

3.3 Phase 3: Feature Engineering

The final representation combined:

- **TF–IDF** (unigrams and bigrams) fitted on the training set only (`max_features`=60,000).
- **Assigned stylometric features** computed on the original text (not stemmed), then standardized using a scaler fitted on training data only.

The assigned stylometric features implemented were:

1. $f1$: Total number of characters (C)
2. $f24$: Number of different punctuation signs divided by C
3. $f47$: Pronoun count
4. $f70$: First-person grammatical features count
5. $f93$: Emotional arousal score (mean)

3.4 Phase 4: Modeling and Evaluation

Three models were trained:

- **Multinomial Naïve Bayes** baseline (TF–IDF only)
- **Logistic Regression** (TF–IDF + stylometry)
- **Linear SVM** with probability calibration (TF–IDF + stylometry)

Given class imbalance, evaluation included **balanced accuracy** and ROC–AUC in addition to accuracy, precision, recall, and F1-score. For Logistic Regression and Linear SVM, the classification threshold was tuned on the validation set to maximize balanced accuracy.

4. Results and Analysis

Table 1 summarizes the final test performance reported by the implemented pipeline.

Table 1: Test-set performance summary.

Model	Accuracy	Balanced Acc.	Precision	Recall	ROC–AUC
Naïve Bayes (TF–IDF)	0.9374	0.8638	0.9384	0.9865	0.9838
Logistic Regression (TF–IDF + Stylo)	0.9844	0.9786	0.9922	0.9883	0.9963
Linear SVM (TF–IDF + Stylo)	0.9868	0.9846	0.9952	0.9883	0.9966
Linear SVM (Tuned Threshold)	0.9828	0.9863	0.9980	0.9805	0.9966

4.1 Confusion Matrix and ROC Curve

Figures 1 and 2 present the test-set confusion matrix and ROC curve produced for the best model by validation ROC–AUC (Linear SVM). The confusion matrix shows strong performance on both classes, while the ROC curve indicates near-perfect separability.

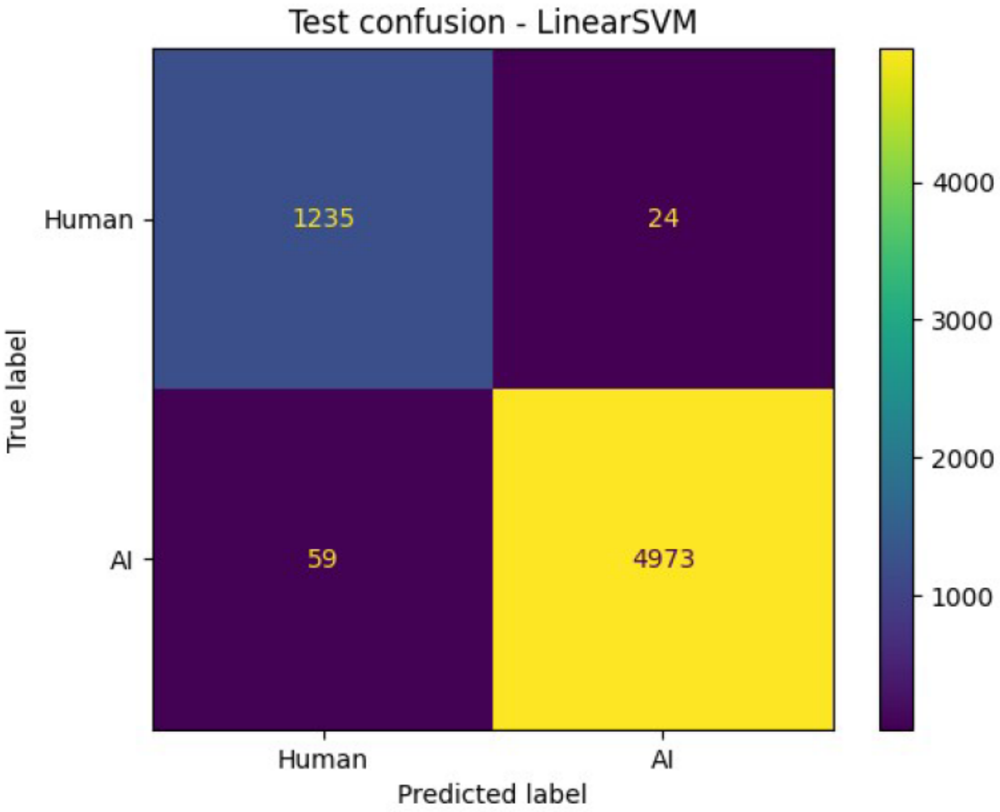

Figure 1: Test confusion matrix for Linear SVM (as generated by the project pipeline).

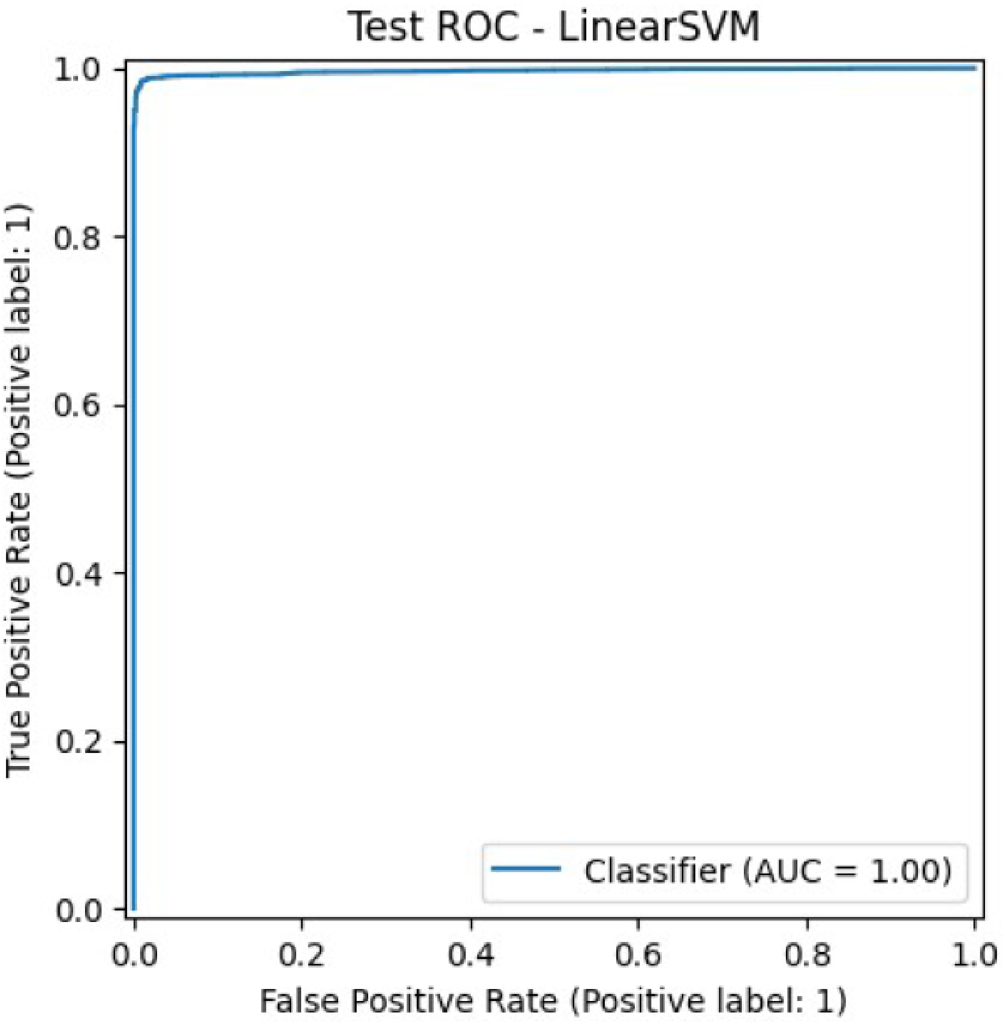

Figure 2: Test ROC curve for Linear SVM (as generated by the project pipeline).

4.2 Error Analysis

The pipeline printed the most confident misclassifications for inspection. A common pattern in the error examples is that some AI-generated texts are highly formal and coherent, resembling academic human writing, while a small portion of human abstracts can exhibit formulaic phrasing that overlaps with AI stylistic tendencies. This suggests that detection performance may degrade when generation models are prompted to mimic highly structured academic Arabic.

5. Conclusion and Future Work

This project demonstrates that AI-generated Arabic text can be detected with very high performance using traditional machine learning when combining TF–IDF lexical signals with targeted stylometric features. Linear SVM achieved the strongest overall performance, and threshold tuning improved balanced accuracy under class imbalance. Future work can extend the study by (1) evaluating cross-domain generalization, (2) testing robustness against newer generation models, and (3) adding a deeper neural approach (e.g., a feedforward network over Arabic BERT embeddings) for comparison.