CS 6220: PROJECT REPORT

Team Members

- Harshit Kumar Taneja
- Rohith Kumar Senthil Kumar
- Siddarth Srinath

GitHub Repo and Dataset Links

GitHub Repo: https://github.com/2025-F-CS6220/project-project-aidatahunter

Dataset: https://northeastern-my.sharepoint.com/:f:/g/personal/taneja_h_northeastern_edu/EtEBHpYIGpZGj2pj3p6OsC0BfH9eJTKgOPBUsw04JgOqiA?e=h1q6fh

Project Overview

This project focuses on the timely and increasingly important task of distinguishing AI-generated text from human-written text. Using a large dataset of 500,000 labeled texts from Kaggle, MAGE set, Mix Set, RAID, we set out to build an effective binary classifier capable of identifying whether a given piece of writing was produced by a human or an AI system. Our goal is to train accurate models by designing a clean, scalable, and reproducible machine-learning pipeline that could serve as a foundation for future extensions such as multimodal detection.

To achieve this, we developed a complete end-to-end workflow that included text cleaning, lemmatization, validation of labels, duplicate removal, feature engineering, TF-IDF and Bag-of-Words vectorization, and careful handling of class imbalance through class weights. We trained and compared multiple models, including Logistic Regression, Random Forest, XGBoost, and a BERT-based deep learning approach, evaluating using stratified train, validation, and test splits and metrics such as accuracy, Precision, Recall, and F1 score. Across models, performance remained consistently strong, with our Random Forest classifier achieving 91% accuracy with a similar F1 score, demonstrating that traditional ML models, when paired with thoughtful preprocessing and feature engineering, can perform competitively on large-scale text classification tasks.

Overall, this project showcases our ability to build a high-performing ML classifier, analyze complex text datasets at scale, and construct a well-designed pipeline that is both robust and extensible. The results highlight solid modeling choices, reliable performance across algorithms, and a foundation that can be expanded to more advanced forms of AI-generated content detection in the future.

Input Data

For our project, we have combined the following four datasets:

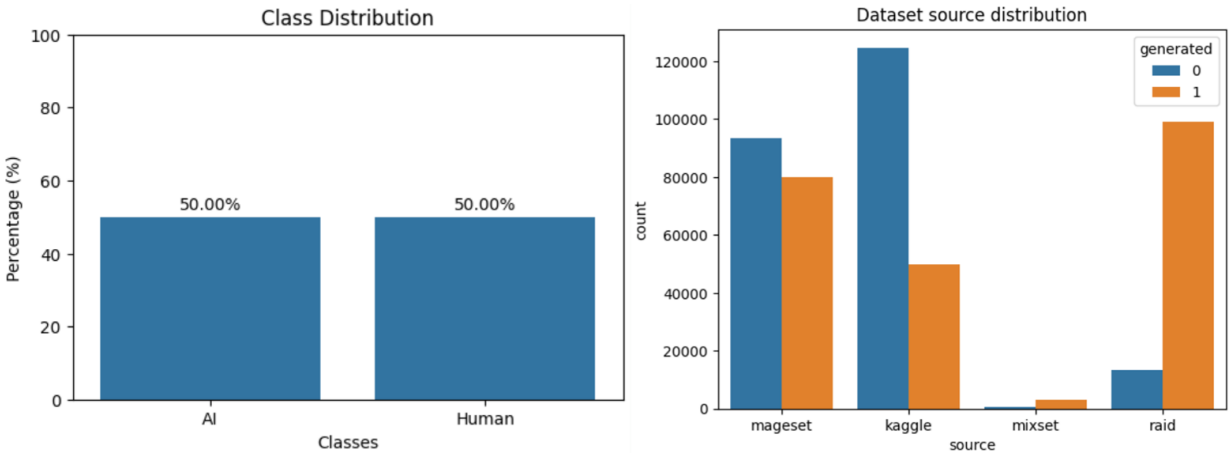
Datasets	Description
AI vs Human Text – Kaggle https://www.kaggle.com/datasets/shanegerami/ai-vs-human-text/data	Contains: Plain Human and AI generated text (no information on which model was used for AI text generation)
	Human Text Sample: Close your eyes and imagine you are Living a c...
	AI Text Sample: The Benefits of Limiting Car Usage \n\nReducin...
MAGE set Github: https://github.com/yafuly/MAGE Paper: https://arxiv.org/abs/2305.13242	Contains: Human-written texts from 10 datasets covering a wide range of writing tasks, e.g., news article writing, story generation, scientific writing. Machine-generated texts from 27 mainstream LLMs including OpenAI, LLaMA, and EleutherAI etc;
	Human Text Sample: Zero-shot learning (ZSL) can be formulated as a cross…
	AI Text Sample: The accident, which involved a white Vauxhall Astra…
MixSet Github: https://github.com/Dongping-Chen/MixSet Paper: https://arxiv.org/abs/2401.05952	Contains: Pure human written text and AI generated text which was then edited by humans to add human noise.
	Human Text Sample: There is currently no widely recognized popular drink…
	AI Text Sample: I initially purchased the game during its early…
RAID Github: https://github.com/liamdugan/raid Paper: https://arxiv.org/abs/2405.07940	Contains: Human and AI generated text that includes 11 different adversarial attacks that could fool AI detectors by transforming a text into patterns that the AI detectors have never seen during training.
	Human Text Sample: We quantify the amount of information filtered by different hierarchical\nclustering methods on…
	AI Text Sample: (Attack used: perplexity_misspelling) Lily\'s close freind, Sam, has a crush on her, and he becomes worried

To begin with, we first performed certain preprocessing to our data. The preprocessing pipeline begins by loading the dataset, cleaning text, validating labels, and removing duplicates. Text is standardized using basic_clean, which lowercases text, normalizes whitespace, and expands contractions (e.g., can't → can not) through the expand_contractions helper. We also compute simple structural features such as character counts, word counts, sentence estimates, punctuation frequencies, lexical diversity, and average word length using text_basic_features. These features capture stylistic patterns that may help distinguish AI-generated writing from human writing.

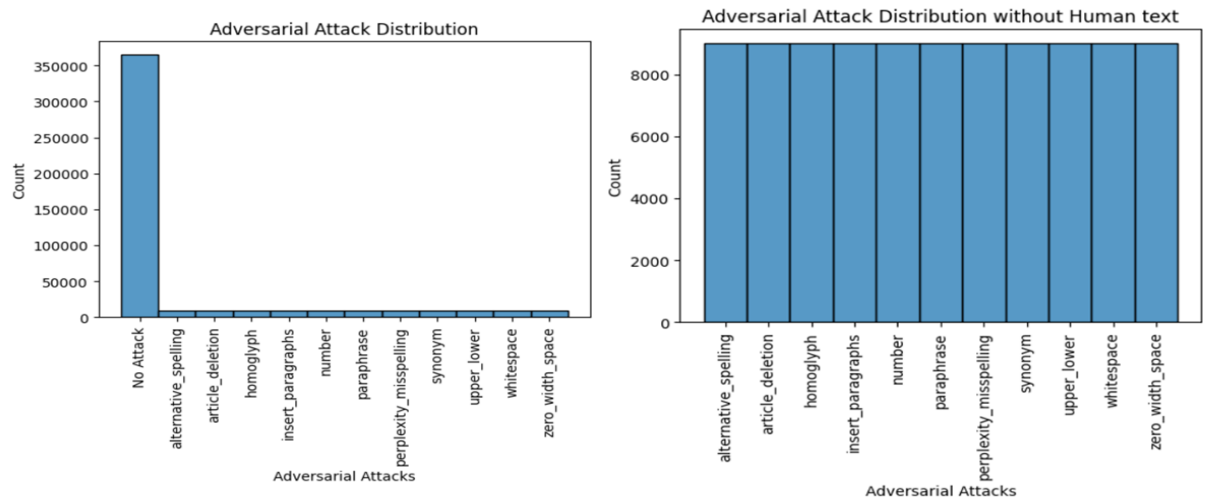
After cleaning, we perform a stratified train/validation/test split and apply two vectorization methods—TF-IDF and CountVectorizer—to convert text into sparse numerical representations. The numeric text features are scaled with StandardScaler, and class weights are computed to account for any imbalance in the dataset. The function returns all processed artifacts, including the split data, vectorizers, feature matrices, and scaling objects, enabling consistent and reproducible model training. After preprocessing, we analyzed our dataset to identify patterns in our data.

Exploratory Data Analysis

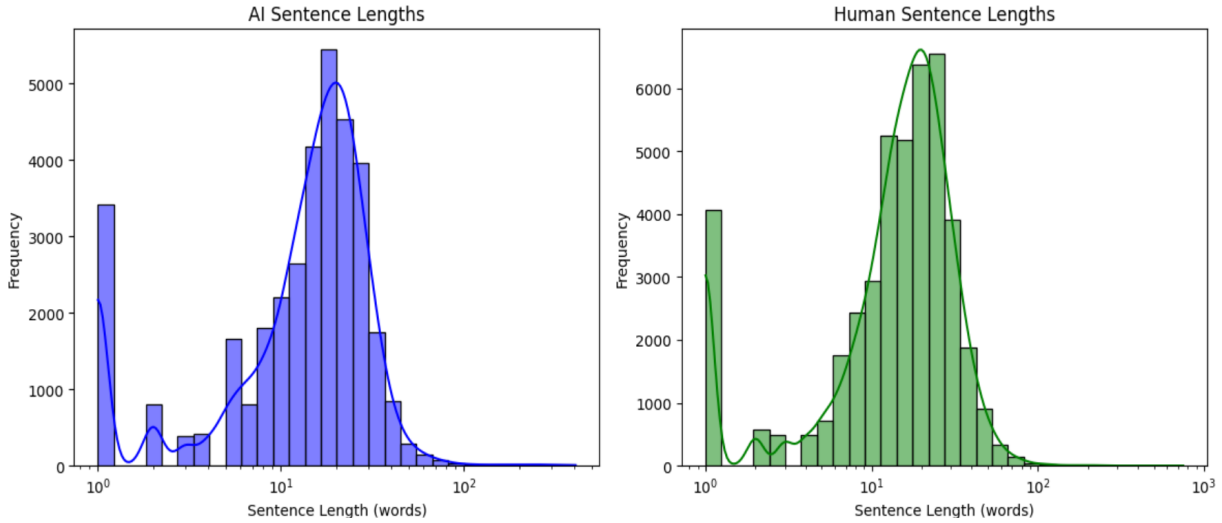
Some of the analysis that we performed are as below:



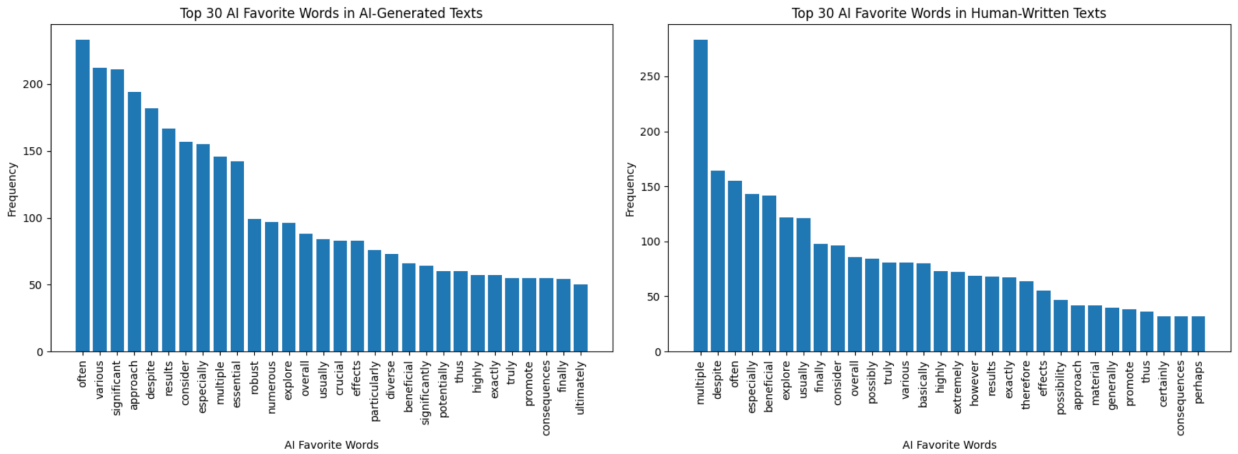
From the above two plots, we can say that the classes are well balanced overall, internally Kaggle dataset dominates for human writing and RAID dataset dominates for AI generated text.



The above two plots depict the distribution of various adversarial attacks, from the RAID set.



From the above plots, we can infer that sentence length alone is not a strong discriminative feature, and AI and human text vary in sentence length in very similar ways.



The above plot reveals that humans tend to have more lexical diversity and favor less towards AI favorite words.

Problem

The problem we aimed to solve was determining whether a given piece of text was written by a human or generated by an AI model. With the rapid rise of large language models and the increasing difficulty of distinguishing AI generated writing from authentic human writing, this task has become both relevant and challenging. We approached this as a binary classification problem, using large, labeled datasets sourced from Kaggle, RaidSet, MageSet, and MixSet. These datasets contained diverse writing samples, each clearly labeled as either human-written (labelled as 0) or AI generated (labelled as 1), allowing us to train models to recognize subtle linguistic differences between the two categories.

The core objective was to learn that could reliably indicate the source of the writing. Beyond simply assigning a label, we sought to understand how different machine learning models could make this distinction, identifying edge cases where classification becomes difficult, and exploring what types of features or model architectures are most effective. By framing the task this way, we sought to build a robust system capable of detecting AI-generated text while gaining insight into the broader landscape of human vs. AI writing.

Evidence of Success

For AI text detection we prioritize the following metrics:

Metric	Why it matters AI Text detection
Accuracy	Measure of overall correctness, a good metric as the classes is well balanced.
Human Recall	Measure to ensure Human written text is caught. False positives in AI generated text detection are bad since humans will get penalized for using AI which causes more harm than missing AI-generated text.
AI Recall	Measure to ensure AI generated content is caught.
Human Precision	Measures how reliable the model is when it predicts text as human written. Low human precision means AI-generated text slips through undetected.
F1-Score	Measure the tradeoff between precision and recall.

For logistic regression, we measured success using accuracy, F1-score, and recall, since correctly identifying both human-written and AI-generated text is important. Before hyperparameter tuning, logistic regression with default parameters achieved a validation accuracy of 83.95%, F1-score of 0.838, and recall of 0.800 for Class 0 and 0.884 for Class 1. After tuning the regularization strength (C) to 0.5 and increasing max_iter to 500, logistic regression improved to 88.85% validation accuracy, F1-score of 0.883, and recall of 0.882 for Class 0 and 0.897 for Class 1, showing stronger performance across both classes.

Next, we experimented with a Random Forest classifier, which initially reached an accuracy of around 85% using the base configuration. To enhance its performance, we carried out targeted hyperparameter tuning, after which the model improved to 91% accuracy with a similarly strong F1 score. This clear gain showed that the model responded well to structured exploration and that meaningful improvements were achievable beyond the default settings.

We began with a stable baseline configuration and then applied HalvingRandomSearchCV to efficiently search across a wide parameter space. The tuning process explored ranges such as n_estimators between 200 and 600, max_depthvalues including None and 10 to 90 in steps of 10, min_samples_split between 2 and 20, min_samples_leaf between 1 and 10, and different max_features options including "sqrt", "log2", and None. This search setup allowed us to systematically evaluate combinations and identify those that improved generalization. The results confirmed that fine-tuning these parameters had a measurable impact on both stability and predictive reliability, resulting in a stronger, more well-calibrated Random Forest model.

The fine-tuned BERT model delivered strong overall performance, achieving an accuracy of 93.3% across the evaluation set. It demonstrated particularly high reliability in identifying AI-generated text, reaching a 98% recall, meaning it correctly recognized almost all AI-written samples. Human text detection was slightly more challenging, with the model achieving an 89% recall, but still reflecting solid sensitivity to human-authored writing. Combined, these results yielded an overall F1-score of 0.93, indicating a well-balanced classifier capable of robustly distinguishing between human and AI-generated content.

Also, we performed hyper parameter tuning on the BERT model. The result of which are as follows:

Fine tune steps	Frozen layers	Learnable layers	Learning rates	Human Precision
1	Encoder layers 0-7	Encoder layer 8-9	3e-6	97%
		Encoder layer 10	5e-6	
		Encoder layer 11	1e-5	
		Encoder 11 Output layer and Pooler	2e-5	
		Classifier	4e-5	
2	Encoder layers 0-6	Encoder layers 7-9	0.5*3e-6	98%
		Encoder layer 10	0.5*5e-6	
		Encoder layer 11	0.5*1e-5	
		Encoder 11 Output layer and Pooler	0.5*2e-5	
		Classifier	0.5*4e-5	

Evidence of Meaningfulness

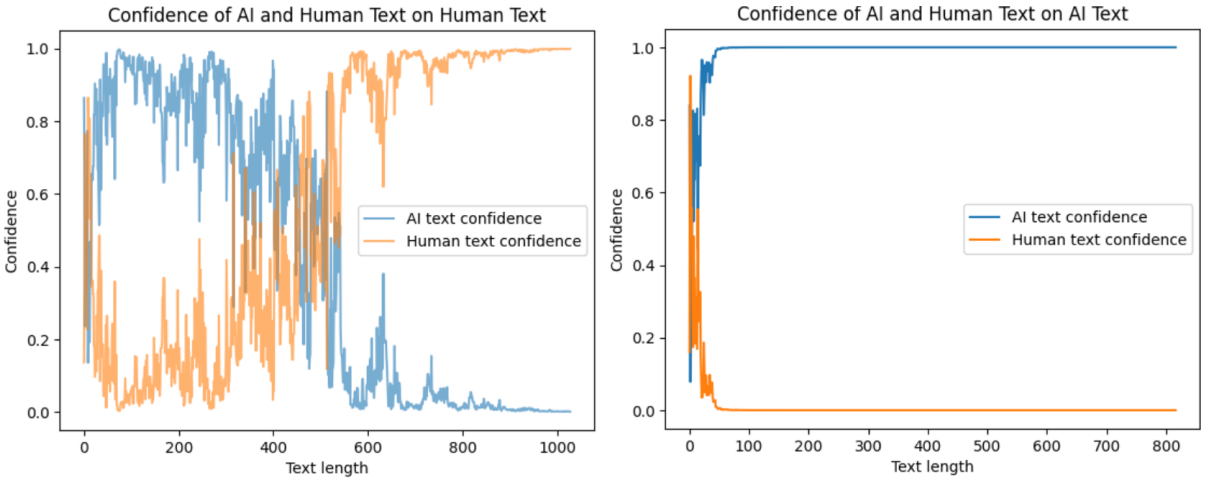
We tested the best-performing model across various scenarios:

a. Test on Uncertainty

We tested combinations of human and AI text sandwiched within each other, human and AI text alternating every sentence, and text where human and AI writing were blended. We discovered that the model is quite strict and predicts text as AI-generated as soon as it detects sufficient AI patterns within it. Consequently, all mixed samples were classified as AI-generated by the model.

b. Test Based on Input Length

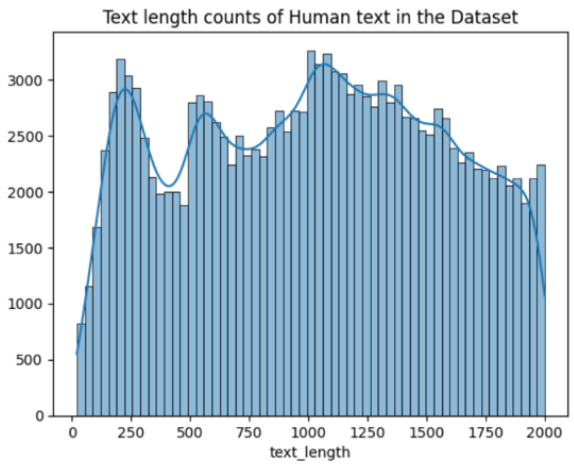
We tested the model on varying text lengths to observe how it would respond. From the results, we found that the model performs well at identifying AI patterns when the given text is genuinely AI-generated. However, for human text, the model showed uncertainty in its predictions when the text length was less than or around 500 characters. The plots below reveal this trend.

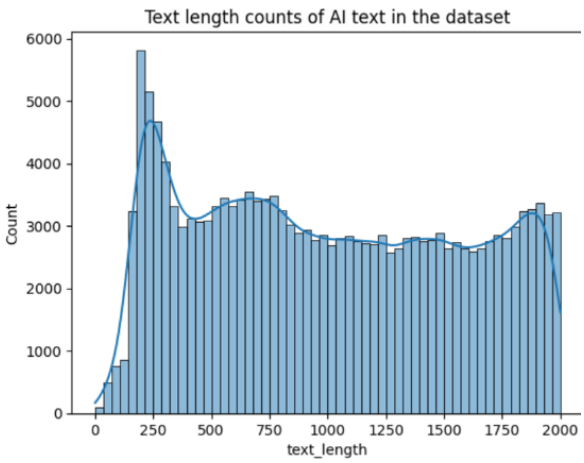


The first plot reveals the model's uncertainty in predicting human text when the text length is between 200-500 characters. Past that range, the model appears to gain more signal and becomes increasingly confident that the given input is human-written. As for the second plot, the model performs well and is highly certain that the given text is AI-generated, irrespective of text length (except when it is very short). This demonstrates that the model has captured genuine AI patterns and is not solely dependent on text length.

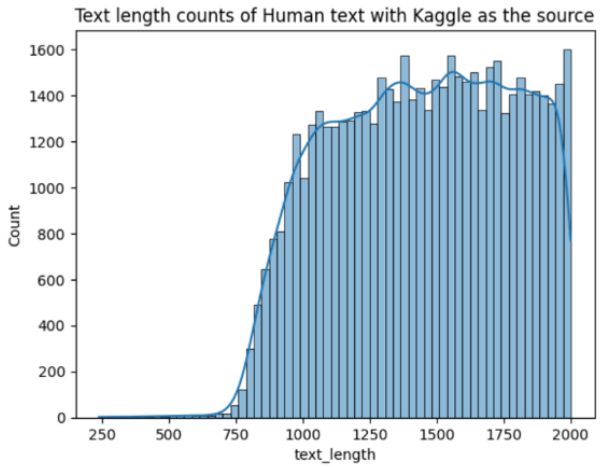
From the above plots, we hypothesize that there aren't enough human records in the 200-500 character range, and because of this, the model hasn't seen enough variance at that range to confidently predict human text.

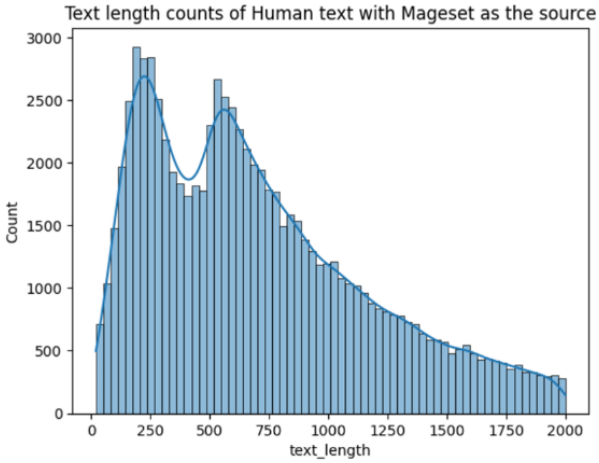
Upon further inspection of the dataset, this is what we found. Below are the plots showing the distribution of the dataset based on sources for AI and Human-written text.

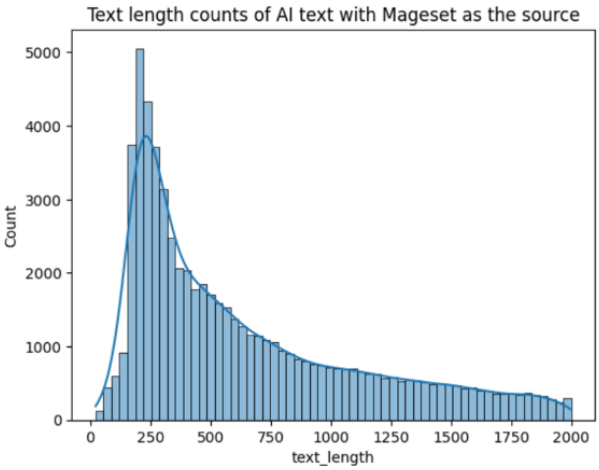


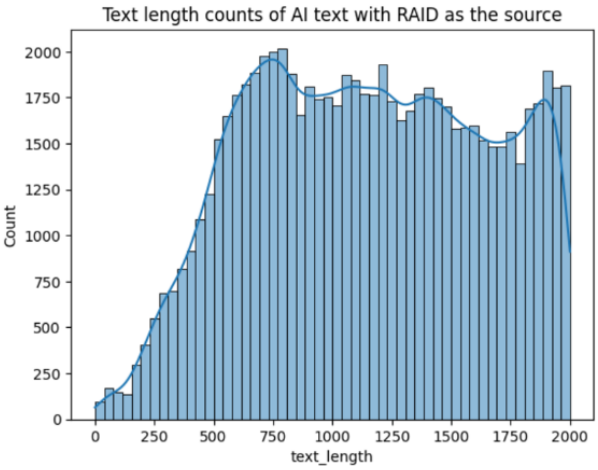


The above plot reveals that both Human and AI text have relatively few records at very short lengths, with AI text having even fewer than Human text in that range. In the 180-550 range, AI text tends to dominate over Human text overall, and past that point, both AI and Human text appear to follow a similar distribution.









From the above plots, it is clear that the Kaggle set, which from the EDA we know dominates the most for human-written text, is heavily skewed to the right, with very few samples below a

text length of 750. The MAGE set, the second largest contributor to human-written records, is concentrated at shorter lengths, where AI text tends to dominate overall. The RAID dataset has good representation across the 200-1000 range, including substantial number of samples in the 300-800 zone.

Our final hypothesis for why the model struggles to predict human text within the 200–500-character range is the sheer count of AI text records from the MAGE set in that range, combined with AI-generated text from the RAID set that closely resembles human writing.

We hypothesize that at short text lengths, where there is insufficient signal for the model to rely on, RAID's human-like AI text occupies a similar feature distribution as the human-written text from MAGE set. This overlap effectively introduces label noise within that length range, makes the model bias toward predicting AI-generated text in that range.

Conclusion

Overall, our project showed that it's possible to reliably tell apart human-written and AI-generated text by combining solid preprocessing, thoughtful feature engineering, and a mix of traditional and deep learning models. We built a full end-to-end pipeline and found that even simple models like Logistic Regression and Random Forest performed well, while our fine-tuned BERT model pushed accuracy above 93% and proved especially strong at catching AI-generated content. These results show that our approach is both practical and scalable for real-world use. Moving forward, it would be exciting to explore multimodal detection such as using metadata or working with images, to make the system more adaptable over time as more capable AI models emerge.