Chinese Language Web Classification Model
Capstone Project Report

Mingjia “Jacky” Guan

2025-08-21

Table of contents

List of Figures

List of Tables

Abstract

Digital technology integration in education has created significant challenges for content filtering, particularly in non-English educational environments where existing filtering technologies demonstrate limited effectiveness. This study presents the development of an enhanced Chinese Web Classification Model (CWCM) designed to address critical limitations in current web filtering systems used within the global K-12 education sector by constructing a comprehensive dataset of 400k+ Chinese web content samples with focused enhancement of three critical harmful categories: drugs, tobacco, and weapons. Using Jieba-based text segmentation and custom TF-IDF vectorization with strategically extracted feature dictionaries, we evaluated over 34 machine learning algorithms through rigorous 10-fold cross-validation, with methodology incorporating statistical significance testing and correlation analysis to examine the relationship between dataset size and model performance across content categories. The optimized Logistic Regression model achieved a macro-averaged F1-score of 0.9183, demonstrating statistically significant superior performance compared to baseline models including XGBoost ($p < 0.000001$) and LinearSVC ($p < 0.000001$), while statistical analysis revealed strong positive correlations between dataset size and performance metrics ($\rho = 0.6868$ for F1-score, $p = 0.009$), providing empirical evidence for strategic data collection priorities. The model successfully addresses deployment constraints through client-side JavaScript implementation, enabling real-time inference on resource-limited devices while maintaining privacy and eliminating API dependencies, thus contributing to educational technology by providing an effective, deployable solution for Chinese content filtering that balances accuracy, computational efficiency, and practical implementation requirements for protecting digital learning environments.

Keywords: web content classification, Chinese natural language processing, educational technology, machine learning, content filtering, digital safety

Acknowledgments

I would like to thank Prof. Zheng Qu and Prof. Miao Yu for guidance, Mr. A. Y. and Mr. B. L. for continued support and providing directions, DLD Technologies Inc. for bare metal services, and collaborators who provided vital feedback, without all of whom this project would not be possible.

1 Introduction

Digital connectivity now reaches more than half of the world's lower-secondary classrooms, yet the very screens intended to expand learning often become the prime source of distraction for students. The next generation is now constantly immersed in technology, with the average student-to-computer steadily rising from 7:1 in 2012 to about 3:2 in 2022 across 38 countries. Educational departments worldwide have also been eager to adapt technology, with over 85% of OECD countries adopting or promoting explicit strategies to digitize their classrooms. This effort was only compounded by the COVID-19 pandemic, leading to over 1 billion students worldwide adopting digital learning methods. With the gradual paradigm shift–as evident by 85% of countries having plans to increase connectivity in the education sector–the digital age has redefined the modern educational framework, propelling digital learning into the mainstream.

Over the past decade, investors, governments, and the education sector have all shown great interest and confidence in education technology (ed tech). For instance, the global market for ed tech is expected to reach $404 billion in 2025. There has also been a 21.5% year-on-year increase in related research from 2014-2023 regarding digital education. While there is strong enthusiasm to digitize our classrooms, inadequate implementations of technology in education can negatively impact student achievement, mental wellbeing, and cognitive performance.

There is substantial evidence showing that high levels of media multitasking in an education setting–defined as simultaneously engaging in classwork and digital media–resulted in lower overall grades and increased distractibility. Students often feel compelled to simultaneously juggle academic materials, social media notifications, as well as audio-visual content when presented with technology in an academic setting. Laboratory and field experiments demonstrate that task-switching to unrelated digital content overloads cognitive resources, resulting in diminished comprehension, memory retention, and critical thinking during learning tasks, also known as "brain drain."

The decline in attention span is strongly linked to the presence and increasing influence of digital technology, both inside and outside the classroom. According to a 2024 study conducted by the Programme for International Student Assessment (PISA), nearly one in three students (30%) across OECD countries reported being distracted by digital devices in "most or every" math lesson. In countries such as Argentina, Uruguay, and Chile, over 50% of students have

reported a high frequency of digital distraction in a classroom setting. A study published by the Institute for Labor Economics in April 2018 already found that 80% of students engage with two electronic devices simultaneously while studying. Given the expansion of multimedia device use, these figures are likely to be higher today. Most of these studies do not point to technology being the issue – rather, it is the distraction factor that comes with technology, a factor which children have a difficult time reducing by themselves. One often finds existing studies focusing on digital prevalence and outcomes thereof, with less attention to how novel technological interventions could mitigate distraction and underperformance.

1.1 The Loopholes of Legacy Technology

As a direct result of this international trend to digitize education, the ed tech sector should have improved the development of resources that are effective at filtering the internet for appropriate use in classrooms. Legacy web filtering technologies typically operate through one of three primary mechanisms: Uniform Resource Locator (URL) filtering, Domain Name System (DNS) filtering, or proxy-based filtering. URL filtering works by comparing requested web addresses against predetermined block-or-allow lists. DNS filtering intervenes during the domain name resolution process, blocking or allowing connections from entire domains before content is accessed. Proxy-based filtering routes traffic through intermediary servers that inspect and filter requests based on predefined rules.

While these approaches offer advantages such as operational speed, reduced processing overhead, and high scalability when implemented with comprehensive blocklists, they suffer from significant technical limitations. The most notable constraint is their limited granularity, typically operating at the URL/domain level rather than analyzing specific content on individual webpages. This binary approach to access control (either allowing or blocking entire domains) creates particular challenges in educational environments where nuanced content distinctions are essential. For instance, a traditional filter might block or allow an entire news website, without distinguishing between educational news articles, inappropriate content, or entertainment sections on the same domain. Similarly, video streaming platforms contain both valuable educational content and material unsuitable for classroom settings, yet traditional filters lack the capability to differentiate between these content types within the same platform.

Legacy filtering technologies are also vulnerable to various circumvention techniques. Students with basic networking knowledge are able to use public DNS servers to bypass such implementations by IT administrators. Virtual Private Networks (VPNs) provide another evasion route by encrypting traffic and routing it through external servers. Another common method is through web-proxies, where students access websites through intermediary servers that fetch content on behalf of the student-operated device, thus avoiding direct connections, which would potentially flag inappropriate use. Additionally, specialized

circumvention websites and browser extensions continuously emerge, designed specifically to bypass school filtering systems. Even if a combination of such technologies is used, it is still difficult for schools to effectively manage technology access within their network.

However, the educational-technological landscape is perhaps one of the most dynamic markets in today's economy, with complete product updates happening every 36 months on average. In response to these limitations, next-generation filtering technologies have emerged that provide more sophisticated content analysis capabilities. Unlike traditional methods that simply block entire domains, content-aware filtering systems analyze the actual content of web pages, including text, images, and videos in real time. This ensures that there is little to no delay between the harmful content appearing on the device and a filter blocking it. This approach is particularly valuable in educational settings where new harmful content can emerge rapidly, and traditional methods become ineffective or obsolete in the ed tech landscape.

1.2 Integrating Machine Learning

The integration of artificial intelligence and machine learning has revolutionized web filtering capabilities, enabling systems to recognize patterns, learn from new threats, and make increasingly accurate content classifications without relying solely on predetermined rules. One enhancement that could be made to legacy technology is to intercept the network request traffic at the DNS resolution step before into a pre-trained machine learning (ML) model to determine whether to allow or deny access to the requested site. The present study investigates the development of a client-level interceptor application capable of real-time parsing and analysis of website source code through the application of pre-trained machine learning algorithms. This algorithm will be able to separate websites into several pre-determined categories, leaving it up to the individual school administrator to block or unblock, not based on granular websites, but rather on refined categories. As a result, our ML-based web filter will keep students on track and instantly prohibit technological distractions, offering real-time results when delivering results and providing a significant improvement in performance over legacy filtering technology.

2 Background and Related Work

The landscape of machine learning and natural language processing has undergone a remarkable transformation over the past decade, evolving from traditional statistical methods to sophisticated deep learning architectures capable of understanding complex linguistic patterns and contextual relationships. Earlier generation models, particularly sequence-to-sequence architectures, struggled with fundamental limitations in contextual understanding, often failing to capture long-range dependencies and nuanced semantic relationships within text. The introduction of the attention mechanism, popularized by the sem-

inal "Attention is All You Need" paper, marked a pivotal breakthrough by enabling models to dynamically focus on relevant parts of input sequences when generating output tokens, effectively learning the relevance and relationships between words without explicit supervision through the use of additional context vectors. This fundamental shift toward attention-based architectures not only revolutionized natural language processing but also laid the groundwork for the transformer-based models that dominate today's AI landscape. To understand this evolution fully, it is essential to examine both the traditional machine learning approaches that preceded these advances—including boosted trees and support vector machines—as well as the progression of deep learning architectures from recurrent neural networks through to modern transformer-based models like BERT and its variants.

2.1 Logistic Regression

Logistic regression transforms linear combinations of features into probabilities using the sigmoid function, an S-shaped curve that maps any real number to a value between 0 and 1. Unlike linear regression, this approach directly models probabilities, ensuring outputs remain within valid ranges. The model finds optimal parameters through Maximum Likelihood Estimation (MLE), which maximizes the likelihood of observing the training data. The resulting cost function (cross-entropy loss) is convex, guaranteeing that optimization algorithms find the global optimum rather than getting stuck in local minima.

For multi-class problems, two main approaches exist: One-vs-Rest creates separate binary classifiers for each class, while multinomial logistic regression directly models probabilities across all classes using the softmax function. Multinomial approaches generally provide better calibrated probabilities. High-dimensional datasets with thousands of features face the curse of dimensionality, where data becomes sparse and overfitting increases. Three regularization techniques address this: L1 (Lasso) drives coefficients to zero for automatic feature selection, L2 (Ridge) shrinks coefficients while maintaining numerical stability, and Elastic Net combines both approaches to balance feature selection with grouping of correlated features.

Scikit-learn's LogisticRegression class effectively handles high-dimensional sparse data through key parameters: the penalty type, the SAGA solver (best for scalability and all penalty types), regularization strength C (typically starting at 1.0 and inversely proportional to the strength of the regularization), and increased maximum iterations (1,000-10,000 for convergence).

2.2 Boosted Trees

During the initial model selection round, XGBoost was selected as a representative of the boosted trees family of classifiers. Boosted trees are ensemble machine learning models that combine multiple simple decision trees through a process called boosting. The algorithm works by sequentially fitting trees in a

forward, stagewise fashion, where each new tree is trained to correct the errors made by the previous trees. This creates an additive model where individual terms are simple trees that together form a powerful predictor.

Boosted trees overcome the biggest drawback of single tree models - their relatively poor predictive performance. While maintaining the strengths of tree-based methods (handling different variable types, accommodating missing data, fitting complex nonlinear relationships, and automatically handling interactions), boosted trees achieve much higher predictive performance compared to single trees, often surpassing most traditional modeling methods in terms of out-of-box performance. They're particularly dominant for tabular data tasks and can achieve both accurate prediction and meaningful explanation simultaneously, making them one of the most successful traditional machine learning approaches available.

At each iteration during the training process for boosted training algorithms, new trees are trained on pseudo-residuals (negative gradients of the loss function), effectively moving the ensemble toward optimal predictions through sequential learning. Most modern boosting frameworks enhance basic gradient boosting through various optimization techniques including second-order derivatives, regularization terms, and sophisticated tree-building strategies. The regularization components prevent overfitting through tree structure constraints (controlling tree depth, minimum leaf weights, and split requirements) and weight penalties using L1 and L2 regularization methods.

For multiclass classification, boosting frameworks typically employ softmax objectives where class probabilities are computed using exponential functions normalized across all classes, optimizing multiclass log-loss across all classes simultaneously. Different frameworks implement multiclass classification through various strategies including one-vs-rest, one-vs-one, or direct multiclass approaches. When handling high-dimensional sparse data, modern boosting frameworks leverage compressed storage formats for memory efficiency, sparse-aware algorithms that handle missing values intelligently, and feature subsampling strategies at multiple granularities to achieve significant computational savings while maintaining performance.

Hyperparameter tuning across boosting frameworks generally follows hierarchical approaches: first optimizing core tree parameters (maximum depth, minimum samples), then regularization parameters (weight penalties, subsampling ratios), and finally efficiency parameters specific to each implementation. The learning rate controls how much each weak learner contributes to the final ensemble, where lower learning rates require more iterations but generally produce better generalization. The optimal number of boosting rounds is typically inversely related to the learning rate, with frameworks balancing training speed and generalization through regularization and early stopping mechanisms.

2.2.1 LightGBM

LightGBM is a gradient boosting framework that achieves high speed and memory efficiency via histogram-based split finding, leaf-wise (best-first) tree growth, and two core innovations: Gradient-based One-Side Sampling (GOSS) to prioritize high-gradient instances and Exclusive Feature Bundling (EFB) to compress sparse, mutually exclusive features. These design choices typically yield faster training than conventional level-wise boosters at comparable accuracy, with strong scalability and distributed/GPU support reported across applications. Key weaknesses include sensitivity to hyperparameters due to leaf-wise growth, which can overfit without depth/leaf constraints; practical tuning difficulty; and potential memory intensity on very large, high-dimensional data despite overall efficiency gains.

2.2.2 XGBoost

XGBoost is a regularized gradient boosting implementation known for robust accuracy on tabular data, with features such as L1/L2 regularization much like LightGBM, but also sparsity-aware split finding that learns default directions for missing values, and approximate split methods such as weighted quantile sketch to scale training. Like LightGBM, models still require careful tuning and can overfit if regularization and early stopping are not used appropriately. Limitations often cited are heavier computational and memory costs relative to newer histogram/leaf-wise designs and a comparatively steeper configuration burden to reach peak performance on large datasets.

2.3 LinearSVC

Support Vector Machines represent one of the most theoretically grounded and practically effective machine learning algorithms, particularly well-suited for high-dimensional sparse data classification tasks. The mathematical elegance of SVMs stems from their foundation in statistical learning theory and convex optimization, making them especially powerful for datasets with thousands of features where traditional methods often struggle. Unlike many algorithms that suffer from the curse of dimensionality, SVMs often improve performance as dimensionality increases, making them ideal candidates for modern data science challenges involving feature-rich yet extremely sparse datasets.

The core strength of linear SVMs lies in their ability to find optimal decision boundaries through a process that balances margin maximization with empirical risk minimization. This optimization problem seeks to identify the hyperplane that best separates different classes with modest error tolerance with slack variables. The resulting model depends only on a subset of training points called support vectors, which provides both computational efficiency (with the exploitation of the kernel trick) and strong generalization capabilities. This sparse representation is particularly advantageous when working with high-dimensional data, as the model complexity depends on the number of support vectors rather

than the total dimensionality of the feature space.

For high-dimensional sparse data applications, particularly those involving thousands of features, scikit-learn's LinearSVC implementation provides significant computational advantages over general SVM approaches. The algorithm leverages coordinate descent optimization that scales linearly with both samples and features, making it highly efficient for large-scale problems. Additionally, the ability to apply L1 regularization enables automatic feature selection, effectively identifying the most relevant predictors while maintaining model interpretability. This combination of computational efficiency, theoretical foundation, and practical effectiveness makes SVMs an excellent choice for classification tasks in high-dimensional sparse environments where traditional methods may falter.

2.4 Modern Deep Learning Algorithms

The evolution from traditional machine learning approaches to modern deep learning represents a fundamental shift in how machines process and understand language. While boosted trees and linear SVMs excel with structured tabular data, the complexity of natural language understanding demanded more sophisticated architectures capable of capturing contextual relationships and semantic meaning at scale.

2.4.1 Recurrent Neural Networks and Sequential Processing

The foundation of modern NLP began with Recurrent Neural Networks (RNNs), which introduced the revolutionary concept of sequential information processing Schmidt, 2019. Unlike feedforward networks that process inputs independently, RNNs maintain hidden states that carry information across time steps, enabling them to understand context within sequences. The mathematical formulation shows this clearly:

$$H_t = \phi_h X_t W_{xh} + H_{t-1} W_{hh} + b_h$$

However, vanilla RNNs suffered from the vanishing gradient problem, where information from distant past tokens would effectively disappear during backpropagation through time This limitation was partially addressed by Long Short-Term Memory (LSTM) units, which introduced gating mechanisms to selectively retain and forget information across longer sequences.

2.5 Long Short-Term Memory (LSTM) Networks: Solving the Memory Problem

While vanilla RNNs introduced the groundbreaking concept of sequential processing, they suffered from a fundamental limitation that severely restricted their practical applications: the vanishing gradient problem. As information

propagated backward through time during training, error signals would exponentially decay, making it nearly impossible for networks to learn dependencies spanning more than a few time steps. The mathematical foundation shows that gradients scale by products of weight matrices and activation function derivatives across time steps, causing exponential decay when values are consistently less than 1.0 or explosive growth when greater than 1.0.

Long Short-Term Memory (LSTM) networks, introduced by Hochreiter and Schmidhuber in 1997, revolutionized sequential modeling by introducing sophisticated gating mechanisms that enable constant error flow through specially designed memory cells. The LSTM architecture centers around memory cells with linear self-connections called "Constant Error Carousels" (CECs), protected by three types of gates: input gates that control when new information should be stored, forget gates that decide what information should be discarded, and output gates that control when stored information should influence other units. The genius of LSTM lies in these multiplicative gates, which solve the conflicting weight update problem by providing context-sensitive control mechanisms—the input gate learns when to write to memory, the forget gate learns what to discard, and the output gate learns when to read from memory, all while the memory cell maintains constant error flow through its linear self-connection.

LSTM's ability to bridge time intervals exceeding 1000 steps fundamentally changed what was possible in language modeling and sequential tasks. Unlike previous approaches that could only capture short-term dependencies, LSTMs could maintain coherent representations of linguistic context across entire paragraphs or documents, enabling practical applications in machine translation, where source sentence information needed to be preserved while generating target translations, and in language modeling, where long-range syntactic and semantic dependencies could finally be captured effectively. LSTM variants and improvements continued to dominate sequence modeling until the transformer revolution, with architectures like GRUs (Gated Recurrent Units) simplifying the gating mechanism while maintaining the core benefits of selective memory management.

bidirectional approach proved crucial for tasks requiring comprehensive understanding of linguistic context, laying the groundwork for the LSTM and subsequent transformer revolution that followed.

2.5.1 The Transformer Architecture Revolution

The seminal "Attention is All You Need" paper fundamentally changed the landscape by introducing the Transformer architecture, abandoning recurrence entirely in favor of self-attention mechanisms. This paradigm shift enabled models to attend to all positions in a sequence simultaneously, dramatically improving both training efficiency and the model's ability to capture long-range dependencies.

The attention mechanism allows the model to dynamically focus on relevant

parts of the input when processing each token, similar to how humans selectively attend to different words when comprehending text. Multi-head attention extends this capability by allowing the model to attend to information from different representation subspaces simultaneously, creating a rich understanding of contextual relationships.

2.5.2 BERT and Bidirectional Language Understanding

BERT (Bidirectional Encoder Representations from Transformers) represented a revolutionary paradigm shift in natural language processing by introducing truly bidirectional pre-training through its innovative masked language modeling approach. Unlike its predecessors, such as GPT and ELMo, which processed text in a unidirectional manner or used shallow concatenation of independently trained left-to-right and right-to-left models, BERT's architecture enables the model to condition on both left and right context simultaneously across all layers of the network.

The fundamental innovation lies in BERT's use of the Transformer encoder architecture with bidirectional self-attention. In traditional left-to-right language models, each token can only attend to previous tokens in the self-attention layers, creating a fundamental limitation in understanding context. BERT overcomes this constraint by employing a "masked language model" (MLM) pre-training objective inspired by the Cloze task. During training, the model randomly masks 15% of the input tokens and attempts to predict these masked tokens based on the complete bidirectional context. This approach forces the model to learn rich representations that incorporate information from both directions of the sequence.

The masking strategy itself is carefully designed to prevent the model from simply memorizing token positions. When a token is selected for masking, 80% of the time it is replaced with the special [MASK] token, 10% of the time it is replaced with a random token from the vocabulary, and 10% of the time it remains unchanged. This stochastic approach ensures that the model learns to predict tokens based on contextual understanding rather than positional cues, while also addressing the mismatch between pre-training (where [MASK] tokens appear) and fine-tuning (where they do not).

The second pre-training objective, Next Sentence Prediction (NSP), was designed to help the model understand relationships between sentence pairs, which is crucial for downstream tasks like question answering and natural language inference. During pre-training, the model receives pairs of sentences where 50% are consecutive sentences from the same document (labeled as "IsNext") and 50% are random sentence pairs from different documents (labeled as "NotNext"). This objective enables BERT to learn discourse-level representations that capture inter-sentence relationships.

The architectural foundation of BERT relies heavily on the Transformer's multi-head self-attention mechanism, which allows each token to attend to all other

tokens in the sequence simultaneously. This bidirectional attention enables the model to build rich contextual representations where each token's representation is influenced by the entire sequence context. The model processes input through multiple layers of self-attention and feed-forward networks, with each layer refining the representations by incorporating increasingly complex linguistic patterns and relationships.

BERT's input representation is particularly sophisticated, combining token embeddings, segment embeddings, and positional embeddings. Token embeddings represent the actual words or subwords, segment embeddings distinguish between different sentences in a pair, and positional embeddings provide sequence order information. This multi-faceted representation allows the model to understand not just what tokens appear in the sequence, but also their roles and relationships within the broader context.

The impact of this bidirectional approach was immediately apparent in BERT's performance across a wide range of NLP tasks. The model achieved new state-of-the-art results on eleven different tasks, including a remarkable 7.7% absolute improvement on the GLUE benchmark. This success demonstrated that bidirectional pre-training could capture linguistic phenomena that were previously difficult for unidirectional models to learn, such as syntactic dependencies that span long distances and semantic relationships that require understanding of both preceding and following context.

2.6 Optimization and Scaling Improvements

RoBERTa (Robustly Optimized BERT Pretraining Approach) emerged as a systematic investigation into the training procedures and hyperparameters that contribute to BERT's success, revealing that many of the performance gains attributed to architectural innovations were actually due to suboptimal training configurations in the original BERT implementation. The research demonstrated that BERT was significantly undertrained and that careful optimization of the training process could yield substantial improvements without any architectural modifications.

One of the most significant findings was the impact of training duration and batch size scaling. While the original BERT was trained for 1 million steps with a batch size of 256 sequences, RoBERTa experiments showed that training with larger batches (up to 8,000 sequences) for extended periods dramatically improved performance. The research revealed that training with large batches not only improved optimization speed but also enhanced end-task performance when the learning rate was appropriately adjusted. This finding challenged conventional wisdom about batch size limitations and demonstrated that modern optimization techniques could effectively handle much larger batch sizes than previously thought practical.

The data scaling experiments provided equally compelling insights. While BERT was originally trained on approximately 16GB of text from BookCor-

pus and English Wikipedia, RoBERTa systematically expanded this to over 160GB of diverse text data, including CC-NEWS (76GB of English news articles), OpenWebText (38GB of web content), and Stories (31GB of story-like text). Each increase in data volume corresponded to measurable improvements in downstream task performance, with the model continuing to benefit from additional data even at the largest scales tested. This scaling behavior suggested that the model had not yet reached its capacity limits and could potentially benefit from even larger datasets.

Dynamic masking represented another crucial optimization that addressed a fundamental limitation in BERT's original training approach. The original implementation performed masking once during data preprocessing, creating static masks that were reused throughout training. This meant that each training sequence was seen with the same mask pattern multiple times across epochs, limiting the diversity of training signals. RoBERTa introduced dynamic masking, where new masking patterns are generated each time a sequence is fed to the model. This approach provides richer training signals and becomes increasingly important when training for longer periods or with larger datasets, as it prevents the model from memorizing specific mask patterns.

The removal of the Next Sentence Prediction (NSP) objective emerged as a surprising but significant improvement. While the original BERT paper suggested that NSP was crucial for performance, RoBERTa's systematic analysis revealed that removing this objective actually improved or maintained performance on downstream tasks. The research showed that training with full-length sequences packed from multiple documents (without NSP) outperformed the original segment-pair format with NSP. This finding suggested that the model could learn inter-sentence relationships implicitly through the masked language modeling objective alone, without requiring an explicit sentence-level prediction task.

The investigation into sequence length optimization revealed that training with longer sequences throughout the entire training process, rather than using shorter sequences for the initial phases, led to better contextual understanding. BERT's original training procedure used sequences of 128 tokens for 90% of training steps and only used full 512-token sequences for the final 10% of training. RoBERTa demonstrated that using full-length sequences from the beginning of training, despite the increased computational cost, resulted in superior performance on tasks requiring long-range contextual understanding.

Text encoding improvements through the adoption of byte-level Byte-Pair Encoding (BPE) addressed vocabulary limitations in the original BERT implementation. While BERT used a character-level BPE vocabulary of 30,000 subwords with heuristic tokenization rules, RoBERTa adopted a larger byte-level BPE vocabulary of 50,000 units without additional preprocessing. This approach eliminated unknown tokens entirely and provided more robust handling of diverse text inputs, though it required larger vocabulary matrices and slightly increased model parameters.

The cumulative effect of these optimizations was remarkable. RoBERTa achieved 88.5% on the GLUE leaderboard while using the same masked language modeling objective as BERT, demonstrating that training procedures and data quality were often more important than architectural innovations. This finding had profound implications for the field, suggesting that many reported improvements in subsequent models might be attributable to better training practices rather than fundamental architectural advances.

2.7 Knowledge Distillation and Efficiency

The emergence of increasingly large language models created a critical challenge for practical deployment, particularly in resource-constrained environments such as mobile devices, edge computing scenarios, and applications requiring real-time inference. DistilBERT addressed this challenge by pioneering the application of knowledge distillation techniques specifically during the pre-training phase, creating a fundamentally new approach to model compression that maintained strong performance while dramatically reducing computational requirements.

Knowledge distillation, originally developed for supervised learning tasks, involves training a compact "student" model to reproduce the behavior of a larger "teacher" model. The key insight is that the teacher model's output distribution contains richer information than simple one-hot target labels. Even when the teacher model predicts the correct class with high confidence, the relative probabilities assigned to other classes encode valuable information about the model's understanding of input similarity and generalization capabilities. These "dark knowledge" patterns in the softmax output distribution capture nuanced relationships that would be lost in hard classification targets.

DistilBERT adapted this concept for self-supervised pre-training by creating a student model with the same general architecture as BERT but with significant parameter reduction. The student model retained BERT's token and position embeddings but eliminated the token-type embeddings and pooler layer while reducing the number of Transformer layers by half. This architectural choice was informed by empirical observations that variations in the hidden dimension had less impact on computational efficiency than reductions in the number of layers, making layer reduction the most effective approach for achieving meaningful speedups.

The distillation process employed a sophisticated triple loss function that combined multiple learning objectives to maximize knowledge transfer from the teacher to the student. The primary component was the distillation loss, calculated as the cross-entropy between the student's predictions and the teacher's soft target probabilities. This loss function used temperature-controlled softmax outputs, where higher temperatures create smoother probability distributions that emphasize the relative differences between class predictions rather than just the most likely class. The temperature parameter T was applied equally

to both teacher and student during training but reset to 1 during inference to recover standard softmax behavior.

The second component of the loss function was the traditional masked language modeling loss, ensuring that the student model could perform the core pre-training task independently. This dual objective structure allowed the student to learn both from the teacher's knowledge and from the original training signal, preventing over-reliance on the teacher's potentially imperfect predictions while maintaining the ability to generalize to new contexts.

The third and most innovative component was the cosine embedding loss, which aligned the directions of the teacher and student hidden state vectors. This geometric constraint encouraged the student model to learn similar internal representations to the teacher, not just similar output predictions. By aligning the high-dimensional representation spaces, this loss component helped preserve the rich semantic relationships that BERT had learned to encode in its hidden states, ensuring that the compressed model maintained similar capabilities for downstream fine-tuning.

The training procedure for DistilBERT required careful initialization strategies to ensure convergence. Rather than starting with random weights, the student model was initialized by taking alternating layers from the pre-trained teacher model, leveraging the dimensional compatibility between the architectures. This initialization approach provided the student with a substantial head start in learning useful representations and significantly accelerated the distillation process.

The results of this distillation approach were remarkable from both performance and efficiency perspectives. DistilBERT retained 97% of BERT's language understanding capabilities while reducing the parameter count by 40% and achieving 60% faster inference speeds. On the GLUE benchmark, DistilBERT consistently outperformed the ELMo baseline across all tasks and remained competitive with the full BERT model on most evaluations. The model achieved 92.82% accuracy on IMDb sentiment classification compared to BERT's 93.46%, and on SQuAD question answering, it reached 85.8 F1 score compared to BERT's 88.5.

The practical implications of these efficiency gains extended far beyond simple computational savings. DistilBERT enabled deployment scenarios that were previously impossible with full-sized models, including on-device inference for mobile applications where network connectivity might be limited or privacy concerns precluded cloud-based processing. The reduced memory footprint and faster inference speeds made real-time applications feasible, opening new possibilities for interactive AI systems that could respond to user input with minimal latency.

The success of DistilBERT's approach also validated the broader principle that knowledge distillation could be effectively applied during pre-training rather than just fine-tuning. This insight paved the way for subsequent research into

efficient model architectures and training procedures, demonstrating that the benefits of large-scale pre-training could be captured in more compact models through careful distillation techniques. The methodology established a template for balancing model performance with practical deployment constraints, proving that significant efficiency improvements were possible without sacrificing the core capabilities that made large language models valuable for downstream applications.

2.7.1 JinaBERT and Extended Context Processing

Modern embedding models like `jina-embeddings-v2-base-zh` represent the latest evolution in this progression, incorporating architectural innovations to handle longer sequences. Built on the BERT architecture, JinaBERT integrates Attention with Linear Biases (ALiBi) to support sequence lengths up to 8,192 tokens—a 16x improvement over standard BERT's 512-token limit.

ALiBi enables this extension by replacing traditional position embeddings with linear biases applied directly to attention scores, allowing the model to extrapolate to longer sequences than those seen during training. This is particularly crucial for embedding applications where documents may significantly exceed traditional length constraints.

The bilingual design of `jina-embeddings-v2-base-zh` specifically addresses cross-lingual retrieval challenges, enabling effective similarity computation between Chinese and English texts without the bias towards either language that typically characterizes multilingual models. This capability is essential for modern information retrieval systems operating in multilingual environments.

2.7.2 Fine-tuning Strategies and Task Adaptation

Research into fine-tuning methodologies has revealed sophisticated strategies for adapting pre-trained models to specific tasks. Key considerations include:

- **Single-Taks Learning Advantages:** When fine-tuning on multiple tasks simultaneously, models often exhibit performance trade-offs between tasks. This supports our use case to train a hyper-efficient model specialized in categorizing websites based on extracted content.
- **Architecture Selection:** Different downstream tasks benefit from different architectural approaches. For semantic textual similarity, cross-encoder architectures that process sentence pairs jointly often outperform bi-encoder approaches that encode sentences independently.
- **Parameter Update Strategies:** The frequency and methodology of parameter updates during fine-tuning significantly impact final performance. More frequent updates generally yield better results than batch-wise updates across multiple tasks.

With these considerations in mind, a modern NLP-enahnced model should capture longer contextual information beyond token frequency. Due to the limited dataset size, the model should use fine-tuning of a pre-trained model rather than training from scratch, which offers the advantage ofbeing able to detect embedded sensitive content.

2.8 Existing Model

The current model is a Logistic Regression model trained using the one-vs-rest algorithm with one sigmoid activation function for each category with varying activation thresholds–low, medium, and high–that balances recall and false positive rate metrics while maintaining key characteristics of being a rapid inference model. The current model faces severe limitations due to training data size, causing it to perform poorly in production and miscategorize websites that should be blocked, such as google searches for "I want to eat heroin" or "I want to eat amphetamine." However, while the next generation model should improve upon the performance of the current model, it should nevertheless maintain its inference speed and deployment agility in order to maintain the ability to perform real-time detection.

We theorize that the current Chinese Web Classification Model (CWCM) faces limitations in accurately detecting sensitive or harmful content within Chinese websites due to its reliance on token frequency analysis (TF-IDF) rather than contextual understanding, as well as being highly due for a refresh. This approach sometimes fails to capture localized sensitive information and lacks the ability to analyze longer contextual relationships across multiple phrases, resulting in potential misclassification of websites that contain subtle but critical harmful content dispersed throughout seemingly innocuous material.

In order to address these deficiencies, there is a need to develop an enhanced machine learning model (CWCM-V2) that can process longer contextual information and detect nuanced patterns within Chinese web content while meeting specific deployment constraints such as resource consumption and computational speed for a significant number of simultaneous requests. As we face significant limitations regarding the quantity of available training data, the CWCM-V2 must overcome dataset limitations through fine-tuning pre-trained models and enhance classification capabilities for the following categories: Drugs, Tobacco, and Weapons.

CWCM-V2 will function as a secondary detection layer to improve overall accuracy and not replace CWCM. Another requirement for this solution is that it must navigate complex technical requirements including cross-platform compatibility, dual-language tokenizer implementation (Python for training, JavaScript for deployment), and seamless integration with existing serving infrastructures such as TFX or TorchServe, all while maintaining the ability to identify sensitive content that current statistical approaches miss. The end goal is for our model to be able to effectively flag all harmful websites visited by the user while

allowing the user to visit all websites within non-harmful categories.

2.8.1 Categories

Harmful Categories	Non-Harmful Categories
DRUGS: including illegal drugs, drug abuse, recreational and psychedelic drugs, and related topics.	**Arts & Entertainment**: Covers topics related to music, movies, television, visual arts, performing arts, and the broader entertainment industry.
TOBACCO: Include vaping and traditional tobacco products, including stores and advocacy.	**Business**: Encompasses business news, corporate information, entrepreneurship, management, and industry developments.
WEAPONS: Cover BB guns, airsoft, real firearms, as well as other items that can be used to harm others, including but not limited to knives and other melee weapons.	**Community & Society**: Includes topics about social issues, local communities, organizations, and societal trends.
ABORTION: Topics related to abortion, including laws, rights, and advocacy.	**Computer**: Focuses on computing technology, programming, software, hardware, and IT-related subjects.
ADULT: Content related to adult themes, including pornography and adult entertainment.	**Ecommerce & Shopping**: Relates to online and offline retail, marketplaces, shopping guides, and consumer goods.
ALCOHOL: Topics covering alcoholic beverages, consumption, abuse, and related issues.	**Finance**: Covers banking, investing, personal finance, insurance, and economic trends.
GAMBLING: Includes casinos, online gambling, betting, and related activities.	**Food**: Encompasses cooking, recipes, nutrition, restaurants, and culinary culture.
GAMES: Topics related to video games, board games, and gaming culture.	**Health**: Includes medical information, wellness, mental health, fitness, and healthcare services.
Sexuality: Topics related to personal sexuality, LGBTQ+ advocacy, etc.	**Hobby**: Covers leisure activities, crafts, collecting, and personal interests.
LINGERIE: Content related to lingerie, intimate apparel, and associated topics.	**Home & Garden**: Focuses on home improvement, interior design, gardening, and landscaping.
SELF-HARM: Topics covering self-harm, mental health, and support resources.	**Industry**: Relates to manufacturing, industrial sectors, logistics, and B2B services.

Harmful Categories	Non-Harmful Categories
SEX EDUCATION: Topics related to sexual education, health, and awareness. **VIOLENCE**: Topics covering violent acts, crime, and prevention.	**Jobs & Career**: Covers employment, job searching, workplace advice, and career development. **Law & Government**: Encompasses legal topics, government agencies, public policy, and civic information. **Lifestyle**: Includes fashion, beauty, personal development, relationships, and daily living. **News & Media**: Covers news reporting, journalism, media outlets, and current events. **Pets & Animals**: Focuses on pet care, animal welfare, wildlife, and zoological topics. **Reference**: Encompasses encyclopedias, dictionaries, educational resources, and general knowledge. **Science & Education**: Covers scientific research, discoveries, academic institutions, and educational topics. **Sports**: Includes professional and amateur sports, teams, athletes, and sporting events. **Travel**: Focuses on destinations, travel guides, tourism, and transportation. **Vehicle**: Covers automobiles, motorcycles, public transit, and transportation-related topics.

2.9 Target Market

In the context of our target market, the categories Drugs, Tobacco, and Weapons are all highly relevant to Taiwan's K-12 Education System. Illegal drug use among Taiwanese adolescents is a significant concern, with the average age of first use around 12.5 years old. Drug use prevalence is higher among certain groups such as students in night vocational high schools. Schools in Taiwan play a central role in preventing and integrating anti-drug education training into the curriculum, the efforts of which can be supported by our web filter.

Tobacco is often the gateway substance used by youth who later use alcohol or other drugs and is closely monitored through the Taiwan Global Youth Tobacco Survey (GYTS). While the government enforces strict smoking bans on minors, and the GYTS supports ongoing efforts to reduce youth smoking, detection in

schools can play a key role in more localized efforts at preventing early tobacco abuse.

In general, weapons (including BB guns and airsoft) are identified as a category of concern for school filtering and safety systems. The need to enhance monitoring and detection in this regard relates more directly to student safety across the board.

3 Problem Statement

In this current project, we will create a drop-in Machine Learning model for detecting websites in our Chinese customer base with a refreshed training dataset and explore potential replacement algorithms. The updated CWCM model will inherit the advantages of the current model, including achieving real-time detection during inference–speed perceived as "instantaneous" to humans–and is deployable on individual computers with limited computational resources, such as school-issued chromebooks. Our insights demonstrate that effective deployment of modern language models requires careful consideration of both architectural design and training methodologies, with the optimal approach depending heavily on the specific use case and resource constraints.

This project aims to enhance the current Chinese website classification system by expanding data coverage and improving model accuracy through a secondary detection layer. By leveraging advanced contextual analysis enabled by improved machine learning techniques, the updated classification model will provide more robust filtering capabilities while maintaining flexibility for deployment in real-world applications. While the target implementation of the model is to have both a primary detection layer (CWCM) for fast inference and a secondary detection layer (CWCM-V2) for enhanced precision, it is unfortunately not within the submission deadline of the current senior project to complete the secondary layer. As a result, the goal is to create an enhanced version of CWCM by mid-August of 2025 and work on CWCM-V2 in conjunction with the deployment results of the updated CWCM.

4 Methodology

4.1 Data Collection

Our dataset was constructed by appending data to a proprietary dataset provided by the company when the first version of the current baseline model was created. While the final dataset consists of 429'066 datapoints, it is worth noting that roughly 76.11% are in the "White" category, which are considered general purpose websites and mostly harmless. For the purpose of our project, since we are requested to enhance the existing model and focus on the performance on a subset of categories in the "Black," or harmful, categories. Most of the data

collection procedure did not include adjusting or validating the content of the harmless categories and instead focused on the harmful categories.

The first step in data collection consisted of relabeling former harmful URLs used in the first CWCM model from the existing database in order to ensure validity. All URLs are generally separated into two categories, which are hosts and pages, with the latter composed of individual URL websites (such as newspaper articles, forum posts, etc.) with related information bound to the content of the page itself while the former consists of a collection of pages (a main page for a drug-related website, the homepage to an abortion clinic, etc.) containing embedded hyperlinks referencing more websites of the same category. While the current V2 model focuses on the categories of Drugs, Tobacco, and Weapons, the requirement was still to go through all existing URLs present in the existing source file and label them as one of the following categories:

- **P:** Valid URLs considered pages.
- **H:** Valid URLs considered hosts.
- **U:** Unrelated URLs (e.g., domains for sale, fraudulent/malicious sites, 404 errors).
- **I:** Inaccessible URLs (potentially accessible via proxy).

After evaluating the existing dataset of URLs after labeling, we concluded that six categories (Abortion, Drug, Gambling, Selfharm, Tobacco, Weapon) contained an insufficient number of usable URLs and demanded more data to be mined in order to construct a dataset with a sufficient amount of datapoints such the baseline model could be trained.

In order to collect a sufficient amount of URLs with brevity, an option was devised using a selenium-based google search result crawler. Based on a list of keywords related to a certain topic X, we generated permutations of the keywords and append it to the URL of a google search URL root url for the Taiwanese version of the popular search engine. We would exploit the fact that google searches have the results separated by unique `div` tags in the HTML source code in order to extract all relevant hyperlinks to related URLs in the search results. We would also save the embedded hyperlinks for the next few pages with pagination features. A random timer was introduced between clicks in order to avoid the CAPTCHA filter by mimicking human browsing activity.

Under the assumption that the root domain of the URL itself could be a likely candidate for a host page, a script was devised in order to extract the domain from the individual URLs and save them to a separate text file. For both the newly collected pages and hosts, since the quality of the mass-collected URLs remained of concern, it was still necessary to label them by hand. However, as brevity was still a dominant factor under consideration, Google's *Gemini 2.5-Flash(-Lite)* Large Language Model (LLM) Application Programming Interface (API) with a suitable prompt was used as a prescreening method to filter through the URLs before manual labeling, significantly reducing the amount of URLs under consideration. We therefore needed to manually label the extracted URLs

by hand using the same approach as the initial step of the model.

A final collection of URLs consisting of both hosts and pages was finalized by taking the union of the newly collected URLs and the existing set of URLs from the proprietary dataset, all belonging to the harmful category. This finalized URL collection was crawled using an echo server, effectively replicating the url screening process on our production servers for text extraction. The newly extracted harmful text would be unioned with the existing dataset of both harmful and harmless text in order to create the most updated and current dataset, which will hereafter be referred to as "the dataset."

4.2 Preprocessing

While it may be tempting to segment data based on individual words like in the English language, the issue with text segmentation in the Chinese language lies in the nuance that phrases offer. Rather than segmenting the words by character, there is a need to segment based on phrases and how they are commonly employed in the Chinese language. For this, we used Jieba's `dict.txt.big` library to separate text by a predefined dictionary with support for both traditional and simplified Chinese phrases and characters. This allows the text to be segmented into phrases with unique meaning for further representation.

The next step after segmentation is to vectorize the text in order to feed individual documents into a classifier. Several methods, including One Hot Encoding, Count Vectorizer, Hashing Vectorizer, and others were considered, yet we eventually settled on the tried-and-tested TF-IDF Vectorizer due to its considerable ability to capture feature importance in text data. However, the next question that came to mind was the dictionary size and features considered important to the model. After experimenting with various vocabulary sizes (ranging from 5000 – 12'000), number of ngrams per features (ranging from 1 to 5), we quickly realized that there were an abundance of features extracted that did not contribute in any meaningful manner to the classification task at hand as it consisted of integers, stopwords, and the like.

In order to resolve this, a custom dictionary was constructed from text belonging to each category of our current dataset using two distinct feature-extraction methods, which were a simple TF-IDF algorithm and textrank, both implementations native to the Jieba library. For each text file in each category of our dataset, the top 20 features were extracted from the individual text file and added to the general dictionary for voting, with each of the 20 features in the text casting one vote for itself. With each datapoint casting 20 votes, this process was repeated for all datapoints. After all votes have been cast for a given category, the top 200 features with the most votes were taken from each category and saved to a large dictionary.

As for the white category containing the harmless text, given the significant number of subcategories and number of datapoints, the number of top features was increased to 2400 features, providing us with a theoretical maximum of

5000 features per extraction method and 10'000 features for the final dictionary, which is within the acceptable limit in terms of inference speed. In addition to this dictionary, each category's text was concatenated together and the top 200 features were extracted using the TF-IDF method. This monolithic-style dictionary (another theoretical maximum of 2'400 additional features) was also added to the final dictionary, from which a final custom preprocessor class was created, taking in segmented text and producing a vector representation of the corpus that has been vectorized and standardized.

In addition to some rudimentary EDA, some extensive topic modelling was performed using scikit-learn's implementation of Latent Dirichlet Allocation (LDA) and visualized with t-distributed Stochastic Neighbor Embedding(t-SNE) using a linear method (Principal Component Analysis/PCA) as the inference engine. A range of topic counts (from 15 to 150) will be used in order to observe whether there is a difference in separation quality, and more importantly, whether there are topics that are difficult to linearly separate in order to take note of CWCM-V2.

4.3 Model Training

During the model selection process, several algorithms have been put under consideration to create a baseline model suitable for the current application. Using 10-fold cross validation (CV), over 34 models were trained as an initial baseline metric to compare the performance of models with their default hyperparameters. This includes methods such as Logistic Regression, Support Vector Machines (SVM), implementations of boosted trees (such as LGBM, XGBoost), Naive Bayes variations, etc.

During this process, the models with the most promising baseline results were selected including `LinearSVM`, `LogisticRegression`, and some tree-based models such as `RandomForest` and `XGBoost`. Several models such as `SVC` with the `RBF` kernel or `NuSVC` were sifted out due to the difficulty of implementation in production. Other models such as `MultinomialBayes` were excluded due to poor baseline performance, leaving us with the three final models for further tuning: `LogisticRegression`, `LinearSVM`, and `XGBoost`.

A mixture of methods were used for efficient hyperparameter space, including `RandomSearchCV`, `GridSearchCV`, and `BayesSearchCV`, with a host of regularization hyperparameters scanned and iterated through related to each individual model itself. During the final rounds of training, a series of custom weights were introduced for categories with sparse data and potential overlap. This was achieved through obtaining balanced model weights with scikit-learn's default implementation of computing class weights and increasing the importance of the categories that tend to perform poorly and reducing the weight of the category that tends to perform well, usually by a scale of roughly 0.1-0.25.

In the end, while most models had similar performance, `LogisticRegression` was selected due to the ease of explainability of the model and as most models,

while having stellar performance during training even with holdout rounds, often encounter steep drops in performance due to low data quantity and quality when compared with the distribution of real-world data during production. At the same time, custom categorization thresholds were also introduced for the final sigmoid output of the decision probabilities in order to adjust for deployment considerations, catering to sensitivities of certain categories while ensuring that other categories are more likely to be classified.

4.4 Model Deployment and Observation

While most of the model training process is completed in python due to the ease of implementation and compatibility with many machine learning libraries, the model will be deployed in the form of a chrome extension and therefore requires JavaScript as a deployment language. There are three components to the model, which consist of tokenization, vectorization, and inference step, which are all required to be deployed in a singular JavaScript file with the necessary extensions. While tokenization can be handled by a TypeScript implementation of jieba (`nodejieba`), vectorization and inference can both be converted into deployment format using the `TF.js` library. The custom model weights are parsed to a tf.js model, which are then saved to a production-ready format and sent over to the MLOps department.

After deployment, model performance is monitored on Kibana using Elastic-Search in our production environments. With the continuous inflow of student data and administrator responses, the model will be continuously monitored and tuned with new data in order to adapt to the real-world data distribution while improving model precision and recall metrics.

5 Results

5.1 Data Collection

During the initial phase of data relabeling, we discovered that there was insufficient data for the categories Abortion, Drugs, Gambling, Selfharm, Tobacco, and Weapons. The script, using the list of keywords generated by permuting keywords related to the topic at hand, allowed us to extract roughly 100 URLs per keyword, resulting in significant gains for URL count.

Table 2: Unfiltered URLs collected with Selenium script.

Category	New URLs collected as of 05/2025
Abortion	14,056
Drugs	4,600
Gambling	4,287
Self-harm	4,859

Category	New URLs collected as of 05/2025
Tobacco	2,248
Weapons	2,482

However, as it is difficult to ascertain whether the URLs collected are of good quality, we still had to manually relabel each individual URL that we were going to use. Out of the additional 3'408 URLs collected and screened both by LLM and hand, an additional 2'460 URLs were marked as relevant and added to the existing database of URLs.

Table 3: Valid URLs after labeling.

Topic	Host Count	URL Count
Abortion	26	311
Drugs	52	332
Gambling	172	235
Self-harm	7	337
Tobacco	135	272
Weapon	57	344

Figure 1: Overview of all black URLs after additional collection period.

Table 4: Summary of all valid URLs by category and type.

Topic	Hosts	Pages
Abortion	40	483
Adult	181	482
Alcohol	65	235
Drugs	62	357
Gambling	207	233
Games	147	3
LGBT	18	531
Lingerie	19	1474
Self-harm	33	540
Sexedu	5	173
Tobacco	146	275
Violence	39	602
Weapons	65	346

The harmful/black URLs were crawled for content using the echo server and parsed to text format, making up the datapoints in our dataset. After crawling, our final dataset was created by taking the union of the previous dataset and the current dataset, combining and removing the duplicate black texts and appending the harmless (white) text to the current dataset. The result is a modest dataset (~2.54 GB in compressed format, ~11.08 GB after decompression) with 412'984 non-null entries and 76.10% of the data belonging to the white category.

IMAGE Distribution of data points by category; percentage of data by black/white categories.

IMAGE Barplot/Pie Chart distribution of black text (note: LGBT, Lingerie, Selfharm, and Sexedu were combined in the Pie Chart for a concise representation).

As we zoom in on the black data, we see that alcohol contains the most data points out of all black categories with 28,825 datapoints, double the number of datapoints from the next category down. The four categories down from alcohol are games, adult, tobacco, and gambling, all hovering around or above the 10 datapoint mark. Rounding off the bottom 3 categories are lingerie (1701), selfharm (1620), and sexed (398).

Table 5: Count of data points by category.

Label	Distribution
white	314,316
alcohol	28,825
games	14,026

Label	Distribution
adult	11,885
tobacco	10,070
gambling	9,526
violence	5,465
drugs	4,946
weapon	4,421
abortion	3,641
lgbt	2,144
lingerie	1,701
selfharm	1,620
sexedu	398

IMAGE Histogram plot of unprocessed text length, unprocessed sentence count, segmented/processed word count, and boxplot of text length by category.

Taking a closer look at the unprocessed and segmented text data, we see that most text samples have a sentence count of up to 2500-3000 words, with the distribution of the data resembling a rather skewed, almost poisson distribution. Before stopwords are removed, sentences have up to 12'000 words per datapoint, that number decreasing to roughly 10'000 after segmentation using Jieba. In the box plot, we can see that most data points contain roughly anywhere between 1'000-10'000 characters on average and rarely exceed that range, meaning that most data points collected contain a usable amount of data that can be further processed and vectorized. For further details on text distribution, please refer to the tables below.

Table 6: Text length statistics by category.

Label	Count	Mean	Std Dev	Min	25%	50%	75%	Max
abortion	3641.0	6709.99	3915.70	104.0	2480.0	7024.0	10236.0	15901.0
adult	11885.0	3389.30	2439.68	102.0	1432.0	2899.0	4708.0	13017.0
alcohol	28825.0	5252.35	3611.40	100.0	2310.0	4007.0	9412.0	12549.0
drugs	4946.0	4904.20	2924.90	111.0	2724.0	4411.5	6318.0	13839.0
gambling	9526.0	3744.82	2675.06	105.0	1748.5	3174.0	4708.0	12950.0
games	14026.0	3475.50	2189.05	106.0	1992.25	2975.0	4522.75	12461.0
lgbt	2144.0	3047.85	2236.94	111.0	1299.0	2586.5	4299.0	12374.0
lingerie	1701.0	3146.06	1997.06	100.0	1554.0	2982.0	4509.0	11296.0
selfharm	1620.0	2736.99	2066.88	136.0	1244.0	2004.0	3821.5	12816.0
sexedu	398.0	3138.07	1941.70	130.0	2114.25	2845.5	3801.25	11934.0
tobacco	10070.0	5453.66	3483.81	101.0	2674.25	4543.0	8769.75	12716.0
violence	5465.0	6089.82	3337.75	105.0	3005.0	5802.0	9336.0	12387.0
weapon	4421.0	5745.53	3504.39	130.0	2541.0	4974.0	9333.0	14301.0

Label	Count	Mean	Std Dev	Min	25%	50%	75%	Max
white	314316.0	4204.18	3678.12	100.0	1765.0	3481.0	5528.0	760072.0

Table 7: Raw text length statistics by category.

Label	Count	Mean	Std Dev	Min	25%	50%	75%	Max
abortion	3641.0	1102.49	841.44	1.0	181.0	1054.0	1991.0	3002.0
adult	11885.0	430.67	320.18	2.0	202.0	372.0	593.0	3441.0
alcohol	28825.0	793.85	567.15	11.0	313.0	599.0	1323.0	2234.0
drugs	4946.0	668.25	501.40	10.0	263.0	581.0	866.75	2397.0
gambling	9526.0	411.21	445.44	2.0	146.0	281.0	470.0	5052.0
games	14026.0	416.83	297.77	2.0	233.0	357.0	506.0	2941.0
lgbt	2144.0	328.73	279.58	1.0	151.75	253.0	401.0	3354.0
lingerie	1701.0	409.06	271.78	1.0	192.0	401.0	592.0	1611.0
selfharm	1620.0	278.06	231.19	3.0	159.0	206.0	324.25	1806.0
sexedu	398.0	342.11	232.02	16.0	190.5	349.0	392.75	1754.0
tobacco	10070.0	861.14	657.20	5.0	319.0	663.0	1396.0	2762.0
violence	5465.0	930.16	598.08	3.0	335.0	876.0	1503.0	2158.0
weapon	4421.0	841.94	572.94	15.0	310.0	727.0	1424.0	2240.0
white	314316.0	565.48	483.99	0.0	202.0	388.0	781.0	4092.0

Table 8: Sentence count statistics by category.

Label	Count	Mean	Std Dev	Min	25%	50%	75%	Max
abortion	3641.0	1685.08	902.47	32.0	1048.0	1952.0	2123.0	9295.0
adult	11885.0	1549.58	1229.17	17.0	595.0	1228.0	2195.0	7686.0
alcohol	28825.0	1790.69	1231.77	19.0	830.0	1380.0	2796.0	5429.0
drugs	4946.0	1690.11	1121.55	26.0	927.25	1428.5	2180.0	9408.0
gambling	9526.0	1533.26	1041.43	23.0	771.0	1210.0	1940.0	5052.0
games	14026.0	1467.12	978.33	21.0	710.0	1179.0	2056.0	4920.0
lgbt	2144.0	1338.59	967.40	18.0	621.0	1015.0	1873.0	4597.0
lingerie	1701.0	1284.71	842.12	20.0	580.0	1042.0	1820.0	4210.0
selfharm	1620.0	1107.33	786.25	19.0	515.0	880.0	1580.0	3975.0
sexedu	398.0	1256.41	803.29	22.0	585.0	1066.0	1740.0	3899.0
tobacco	10070.0	1628.47	1044.12	19.0	771.0	1279.0	2128.0	4897.0
violence	5465.0	1710.89	1052.17	18.0	811.0	1328.0	2207.0	5189.0
weapon	4421.0	1684.72	1089.21	20.0	790.0	1277.0	2165.0	5021.0
white	314316.0	1420.53	1007.39	15.0	650.0	1148.0	1959.0	4811.0

IMAGE Histogram plot of unprocessed text length, unprocessed sentence count, segmented/processed word count, and boxplot of text length by category.

It is interesting to note that with our dataset containing just over 400'000 datapoints, we only have roughly 309'000 datapoints that contain Chinese. After further analysis, it seems that the white category contains a number of datapoints from other languages, including but not limited to English, Russian, Spanish, and other languages, accounting for the discrepancy.

5.2 Feature Extraction

Initially, a trial run with 5'000, 10'000, and 12'000 features were tried for a baseline LinearSVC model as a test, yet a more rigorous method was needed for the TF-IDF vectorizer dictionary. Using both the TF-IDF and textrank algorithm (both natively implemented in jieba), the top 200 features from the individual categories were extracted and compared. On average, the two methods produced rather diverging results, where the average jaccard similarity is around 0.2743. The most similar category, which just happens to be the category with the least number of datapoints, had the highest Jaccard similarity of 0.434 while the categories tobacco and weapon are tied for the lowest jaccard score of 0.190 each.

Table 9: Jaccard similarity between top 200 features extracted.

Category	Extract Tag Count	TextRank Tag Count	Intersection Count	Jaccard Similarity
abortion	200	200	73	0.223
adult	200	200	91	0.294
alcohol	200	200	82	0.258
drugs	200	200	78	0.242
gambling	200	200	82	0.258
games	200	200	94	0.307
lgbt	200	200	95	0.311
lingerie	200	200	96	0.316
selfharm	200	200	73	0.223
sexedu	200	200	121	0.434
tobacco	200	200	64	0.190
violence	200	200	97	0.320
weapon	200	200	64	0.190

Additionally, 2'400 features were extracted from the white category using the TF-IDF method, which provided us with a theoretical maximum of 10'000 features. However, after taking the union of the resulting textrank and TF-IDF dictionaries, we arrived at 2561 features for the black category. Combined with 2'500 features (instead of 2'400 features) extracted from the white category us-

ing TF-IDF as well as the additional features extracted on a category-wide basis, we arrived at a final TF-IDF dictionary size of 4'783.

5.2.1 Topic Modelling

During the process of topic modelling using Latent Dirichlet Allocation (LDA), it was discovered that some topics and datapoints separated themselves quite well after applying LDA such as Alcohol (Red), Games (Purple), and Gambling (light pink).

Figure 2: Excerpt #1 from LDA with 150 topics, with noticeable confusion clusters between lgbt (dark green), selfharm (grey), drugs (orange), gambling (salmon), adult (pink), and lingerie (light pink) after t-SNE projection.

However, other topics are more difficult to separate, with many localized clusters containing data points from multiple topics, such as between games (purple) and violence (light purple), alcohol (red), selfharm (light brown), lgbt (dark green) and tobacco (light orange), as well as lgbt (dark green), adult (pink), and violence (light purple).

5.3 Initial Model Selection

The result of evaluating the 30+ baseline model candidates showed that linear models (logistic regression, linear support vectors, multilayer perceptron, etc.) and boosted decision trees performed the best out of all the models. It is also worth noting that certain methods outperformed others when it comes to default parameter performance.

(a) Excerpt #2 from LDA with 150 topics, with noticeable confusion clusters between lgbt (dark green), alcohol (red), and self-harm (brown), as well as between games (purple) and violence (light purple) after t-SNE projection.

(a) Excerpt #3 from LDA with 150 topics, with noticeable confusion clusters between lgbt (dark green), adult (pink), and violence (light purple) after t-SNE projection.

Table 10: Best performing model by class.

Class Name	Best Model	F1 Score	Precision	Recall
abortion	RidgeClassifier	0.871	0.846	0.897
adult	PassiveAggressive	0.967	0.964	0.970
alcohol	ExtraTrees	0.997	0.999	0.996
drugs	SVC_RBF	0.871	0.894	0.850
gambling	PassiveAggressive	0.990	0.994	0.987
games	PassiveAggressive	0.986	0.985	0.988
lgbt	RidgeClassifier	0.834	0.960	0.737
lingerie	PassiveAggressive	0.938	0.946	0.929
selfharm	XGBoost	0.903	0.946	0.864
sexedu	SGDClassifier	0.947	1.000	0.900
tobacco	PassiveAggressive	0.992	0.991	0.994
violence	XGBoost	0.863	0.897	0.832
weapon	XGBoost	0.880	0.897	0.864
white	PassiveAggressive	0.997	0.997	0.998

The best model was the passive aggressive classifier, a textbook linear model implementation designed for large-scale learning using squared hinge loss without the need of a learning rate. Similar to LinearSVC and Logistic Regression, the passive aggressive classifier has a regularization hyperparameter C. Nevertheless, Logistic Regression, however, was the most balanced model out of all tested models with a macro averaged F1-Score of 0.9062. In fact, most well-performing models seem to hit a performance bottleneck around the 0.9 mark

and don't seem to improve much in terms of F1-macro performance.

Figure 5: Comparison of all models by class.

Table 11: Top 10 models ranked by accuracy.

Model	Accuracy	Macro Avg F1
PassiveAggressive	0.986	0.907
ExtraTrees	0.985	0.895
Perceptron	0.985	0.900
XGBoost	0.985	0.905
LogisticRegression	0.984	0.910
MLP_Large	0.984	0.904
MLP_Small	0.983	0.902
SVC_RBF	0.983	0.892
LinearSVC	0.983	0.879
RandomForest	0.982	0.889

Based solely on accuracy and macro average F1-scores, the best-performing model is PassiveAggressive, achieving the highest accuracy of 0.986 and a strong macro F1 of 0.907, indicating both overall precision and balanced class-wise performance. ExtraTrees, Perceptron, and XGBoost follow closely with 0.985 accuracy, with XGBoost showing the highest macro F1 among them at 0.905, suggesting more consistent performance across classes. LogisticRegression stands out with a macro F1 of 0.910— the highest among all— despite a slightly lower accuracy of 0.984, reflecting excellent balance across class predictions. The neural models, MLP_Large and MLP_Small, perform comparably with accuracies

of 0.984 and 0.983, and macro F1 scores of 0.904 and 0.902 respectively, indicating strong generalization. SVC_RBF and LinearSVC both achieve 0.983 accuracy, though LinearSVC lags in macro F1 at 0.879. Lastly, RandomForest shows the lowest accuracy (0.982) and a macro F1 of 0.889, still solid but slightly behind the others. Overall, all models perform well, with minor tradeoffs between accuracy and macro-level F1 scores.

Figure 6: Scatterplot of best performing models (accuracy) vs weighted F1-Score.

Figure 7: Macro-averaged precision vs recall of all data categories.

Table 12: F1 score statistics for each class, showing mean, standard deviation, maximum, and support.

Class Name	F1 Score Mean	F1 Score Std	F1 Score Max	Support
abortion	0.718	0.279	0.871	182
adult	0.803	0.314	0.967	594
alcohol	0.901	0.269	0.997	1441
drugs	0.712	0.278	0.871	247
gambling	0.836	0.325	0.990	477
games	0.795	0.329	0.986	702
lgbt	0.596	0.290	0.834	107
lingerie	0.724	0.316	0.938	85
selfharm	0.652	0.300	0.903	81
sexedu	0.704	0.284	0.947	20
tobacco	0.842	0.325	0.992	504
violence	0.677	0.272	0.863	273
weapon	0.706	0.281	0.880	221
white	0.962	0.064	0.997	15716

At the same time, it is worth noting that some categories are relatively easier to classify compared to other models. For example, categories such as alcohol, adult, and gambling, all have a relatively high F1-Macro score while classes such as violence, lgbt, and selfharm all perform rather poorly in terms of model performance. The precision vs recall metrics reveal more about the classes them-

selves, highlighting the difficulty models have with classifying certain categories of the data, highlighting the need for further fine-tuning.

Figure 8: Model complexity vs performance.

It is also worth noting that more complex models perform better when compared to simple models, yet that statement fails to hold beyond moderately complex models.

Figure 9: Top: Model performance along with average performance evaluation. Bottom: Model rank and consistency.

Figure 10: Cross Validation distribution across the top 3 models after initial default search.

Six of the best models were selected and compared across 10-fold CV, with the top 3 models consistently achieving a mean F1-Macro score of around 0.9. As a result, boosted trees, logistic regression, and LinearSVC were selected for the purpose of further fine tuning.

5.4 Boosted Trees

5.4.1 LightGBM

Figure 11: LightGBM randomized hyperparameter search results.

The results from a simple hyperparameter search with 40 iterations of LightGBM revealed that the best validation Macro-Averaged F1-score would still show significant signs of overfitting due to the steep difference between train and validation iteration scores consistently across all models tested. A large max-depth and a small number of leaves for each tree also indicates that overfitting is highly likely in this setup.

5.4.2 XGBoost

Figure 12: XGBoost randomized hyperparameter search results.

Similar to LightGBM, XGBoost also shows similar traits of overfitting as evident by similar train-test F1-macro scores. However, it seems that XGBoost is less reliant on regularization parameters and seems to achieve better performance with less reliance on regularization hyperparameters.

Figure 13: Underfitted model (iteration #15) for XGBoost and train-test F1-macro score comparison.

During the initial exploration process for XGBoost, we can see that there are some underfitted models that show promising results and even signs of generalization (see lingerie category).

Figure 14: XGBoost 3rd round of training with custom weights.

After a more in-depth search of the XGBoost model with custom weights, more

promising model candidates were discovered that perform well yet do not show obvious signs of overfitting. Such a model was submitted as a candidate for the final XGBoost Model.

Figure 15: Confusion matrix of iteration #21 for XGBoost localized hyperparameter search with custom hyperparameter weights.

5.5 LinearSVC

(a) V4.0.0, setting `cut_all` to `True` (a) V4.0.0, using the `cut_for_search` mode

Initially, the baseline model of choice was LinearSVC, implemented with both vectorized text using jieba's `cut_all` and `cut_for_search` mode. These two methods demonstrated poor performance with a F1-macro score of 0.8065 for the `cut_all` method and similar performance for the `cut_for_search` method

before vectorization. It was during this time that there were a few outlier confusion categories, namely abortion and drugs, as well as drugs and lgbt.

Figure 18: V4.4.0 confusion matrix for train set after processing confusion category data.

After using two LLMs and a suitable prompt to adjust the labels of the aforementioned categories, our vectorizer was also recalibrated with the adjusted vectorizer dictionary, resulting in a significant performance increase for our LinearSVC model (F1-macro score of 0.9040).

Figure 19: V4.4.3 holdout set confusion matrix after data and vectorizer recalibration.

A final LinearSVC model was trained using optimal parameters including l2 regularization, and a 8/1/1 train/test/holdout split, demonstrating the model's ability to generalize to a holdout dataset and performance on-par with other model candidates.

Figure 20: Confusion matrix for version 5 (final version) of our LR model.

5.6 Logistic Regression

At this point during the model selection process, both the vectorizer dictionary and the existing issue of the mislabeled data has already been solved. Initial model performance with minimal hyperparameter tuning resulted in an F1-macro score of 0.8955, with one subsequent tune for high recall achieving a recall-macro score of 0.9592. However, the best performing model to date achieved an F1-macro score of 0.9183, precision-macro of 0.9341 and recall-macro of 0.9040, on par with the performance of other model finalists.

Figure 21: Per Class train-test F1-macro, precision-macro, and recall-macro values for V5 (final) version of our LogisticRegression Model.

5.7 Model Selection

5.7.1 Comparing XGBoost, LR, LinearSVC, and LR Confusion Matrices.

When looking at the confusion matrices for our models, it seems that XGBoost generally has a greater tendency to struggle with outlier classification and lower precision performance than the other two models.

Figure 22: Comparing XGBoost, LR, and LinearSVC train-test performances by category.

When comparing by category, it seems that LinearSVC has a slight edge over the other models both in terms of reduced overfitting and general performance.
Final Model

Figure 23: Comparing finalist candidates for Logistic Regression Model: v0 (initial baseline), v2/v3 (optimized for high recall), and v5 (optimized for f1-macro performance); top: per class train-test F1-macro score comparison, middle: per class test F1-macro score comparison and recall-precision trade-off with support indication, bottom: confusion matrix for top 4 model candidates.

The finalist candidate for the Logistic Regression model, designated as V5, emerged from the final round of random hyperparameter searches and represents the optimal configuration for the CWCM refresh. It exhibits a low regularization profile with a C parameter of 220.77 and trained for 154 rounds, indicating a slow and incremental fitting process while maintaining generalization capabilities. The final model employs L2 regularization (`penalty='l2'`) to prevent overfitting and a multinomial decision boundary. The conversion tolerance was also set to 0.0020 (tol=0.0020046685232624705) that provided an optimal balance between training time and solution precision.

6 Discussion

6.1 Model Selection

Linear methods in NLP have long been the norm given the complexity of computational cost associated with non-linear methods and before the arrival of advanced computing resources and methods during the previous decade. While it may be tempting to reach for non-linear methods for text classification, given the limitation of client-side computational resources still relying on low-end CPUs and integrated graphics for processing, we require a fast and efficient linear implementation of the CWCM classifier, both during the initial stage of

standalone model deployment as well as during the second phase where CWCM works in conjunction with CWCM-V2.

As seen during the initial phases of general model comparison and selection, the Passive Aggressive Classifier(PAC) exhibited the best raw performance with untuned hyperparameters. While scikit-learn's implementation of PAC is a method wrapper around a linear model, what is interesting to note is that both PAC and LinearSVC share the same hinge loss function. While there are some non-trivial differences between squared hinge and hinge loss, the loss function no longer considers a datapoint relevant to updating the models parameters once a datapoint has been correctly classified, contributing to how the model effectively generalizes to unseen data in production. The shared loss function means that while the two models are not identical in practice, they are both linear models that share enough traits to substitute one for the other during training. In addition, LinearSVC also has the additional advantage of being particularly robust to previously unseen data as it relies on support vectors as its method of classification.

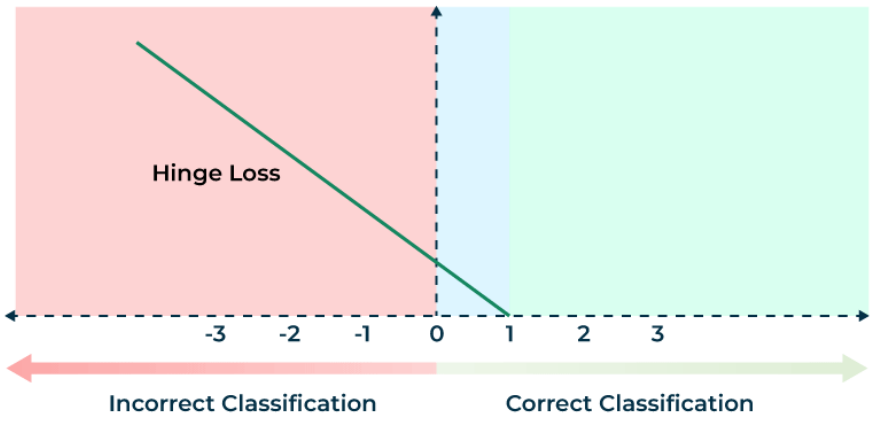

Figure 24: Hinge Loss in SVM

This is what sets the final LinearSVC candidate apart from the two other modes, one being Logistic Regression (linear model using cross entropy loss) and the other being XGBoost (boosted decision tree model also using categorical cross-entropy loss), both of which did not outperform the former. It is also worth noting that while during testing that the LinearSVC model performs slightly better in terms of F1-macro score (0.9270) against Logistic Regression V5 (0.9183) as well as XGBoost #21 (0.9177) even without custom model weights that emphasise the confusion categories. However, while using custom weights for LinearSVC did go unexplored and despite the slight empirical advantage of LinearSVC's F1-macro score, we still opted for Logistic Regression due to the ease of explainability of model performance as well as the fact that hinge loss focuses

on accuracy and margin maximization and not predicting probabilities, which plays a significant role in adjusting sensitivities during model deployment. At the same time, while there might be some academic significance behind a 1-2% increase in model performance, this difference is almost rendered obsolete during production as most linear models tend to overfit and degrade significantly in performance once deployed, especially given the performance of the current CWCM model.

6.2 Logistic Regression vs the Rest

Before diving into the statistical significance of the model itself, there is a noteworthy characteristic of the current model to discuss. It relates to choosing the multinomial solving algorithm for multi-class classification as opposed to using the one-vs-rest (OVR) method. As we know, multinomial regression assumes the Independence of Irrelevant Alternatives (IIA), which is difficult to achieve in real-world applications, especially when it comes to NLP classification tasks. However, despite the theoretical disadvantage of using multinomial classification and the former CWCM model using OVR as a solving method, the multinomial method produced the most optimal results when searching for optimal hyperparameters during this round.

6.3 Statistical Performance Analysis

The comprehensive evaluation of Logistic Regression against two baseline models reveals statistically significant performance differences through rigorous hypothesis testing. When comparing Logistic Regression to XGBoost as the baseline model ($\mu_0 = 0.9183$), the analysis of 130 cross-validation iterations demonstrated a statistically significant underperformance with a sample mean F1 score of 0.91551262. The one-sample t-test yielded a test statistic of $t = -7.60$, which far exceeds the critical value threshold, resulting in a p-value approaching zero ($p < 0.000001$).

The effect size analysis reveals a Cohen's d of -0.66684, indicating a medium practical effect beyond mere statistical significance. This suggests that while XGBoost performs measurably worse than Logistic Regression, the difference represents a meaningful but not extreme performance gap. The 95% confidence interval [0.91478727, 0.91623796] notably excludes the baseline F1 score, providing additional evidence for the significant difference.

At the same time, the comparison with LinearSVC as baseline proves even more pronounced, with Logistic Regression achieving a mean F1 score of 0.90592817 across 66 cross-validation iterations. This second analysis produced a more extreme test statistic ($t = -9.05$) and a substantially larger effect size (Cohen's d = -1.11), categorized as a large effect.

The statistical power analysis confirms that both tests achieved near-perfect power (≥ 0.999), indicating that the sample sizes were more than adequate to

detect the observed performance differences. This high statistical power eliminates concerns about Type II errors and provides confidence that the detected differences represent genuine performance gaps rather than sampling artifacts.

6.4 K-Fold Cross-Validation Performance

The cross-validation results demonstrate notable differences in model stability between the two comparison scenarios. In the XGBoost comparison, the boosted tree model exhibited relatively stable performance with F1 scores ranging from 0.87315251 to 0.91841217 and a standard deviation of 0.00417999, indicating consistent but suboptimal performance across folds. However, the Shapiro-Wilk normality test ($W = 0.315$, $p < 0.001$) revealed severe non-normality in the F1 score distribution, suggesting the presence of outliers or skewed performance patterns that may indicate instability under certain data conditions.

Conversely, the LinearSVC comparison showed greater performance variability with F1 scores spanning from 0.87741934 to 0.92930788 and a higher standard deviation of 0.01111018, yet maintained distributional normality ($W = 0.989$, $p = 0.836$). This pattern suggests that while LinearSVC demonstrates more variable performance against Logistic Regression, the variability follows expected statistical patterns without systematic bias or extreme outliers.

The cross-validation methodology successfully captured the inherent performance uncertainty through repeated sampling, with the larger sample size ($n = 130$) in the XGBoost comparison providing more precise estimates of the true performance difference, as evidenced by the smaller standard error (SE = 0.00036661 vs. 0.00136757).

Table 13: Statistical Metrics for both models.

Statistical Metric	XGBoost Baseline	LinearSVC Baseline
Sample Size (n)	130	66
Baseline F1 Score (μ_0)	0.9183	0.9183
Mean F1 per Model	0.91551	0.90593
Standard Deviation	0.00418	0.01111
Standard Error	0.00037	0.00137
Test Statistic (t)	-7.603	-9.047
Degrees of Freedom	129	65
P-value	< 0.000001	< 0.000001
Effect Size (Cohen's d)	-0.667	-1.114
Effect Magnitude	Medium	Large
95% Confidence Interval	[0.9148, 0.9162]	[0.9032, 0.9087]
Performance Range	0.8732 - 0.9184	0.8774 - 0.9293
Normality Test (p-value)	< 0.001	0.836
Statistical Power	> 0.999	1.000

Statistical Metric	XGBoost Baseline	LinearSVC Baseline
Statistical Decision	Reject H_0	Reject $H_0
Performance Gap	-0.0028	-0.0124

This table clearly demonstrates that Logistic Regression significantly outperforms both baseline models, with a more pronounced surplus against LinearSVC (large effect) compared to XGBoost (medium effect). The statistical evidence is overwhelming in both cases, with near-perfect statistical power ensuring reliable detection of the performance differences. However, while it is worth noting that while LinearSVC shows strong indications of normality, the conclusion for XGBoost's normality test skews strongly in favor of rejection, indicating that further analysis may be required.

Figure 25: Comparing mean F1-Scores and QQ plots for both models against LR baseline metric.

6.5 Statistical Significance of Dataset Size Constraints

As noted in previous sections, there are significant concerns regarding dataset size constraints across the 13 content categories. The dataset exhibits extreme imbalance, with certain categories containing substantially fewer data points compared to others, ranging from 398 samples (sexedu) to 28,825 samples (alcohol) - a ratio of approximately 72:1. It is therefore of great interest whether dataset size systematically impacts model performance across categories, exposing potentially significant correlations that warrant careful consideration for future data collection efforts.

The statistical correlation analysis demonstrates significant positive relationships between dataset size and key performance metrics. Using Spearman rank correlation to account for the extreme class imbalance and potential nonlinear relationships, the analysis revealed medium-to-strong correlations across multiple performance dimensions. F1-macro exhibits a significant correlation with log-transformed sample size ($\rho = 0.6868$, $p = 0.009509$), indicating that categories with larger datasets tend to achieve substantially better F1-macro performance. This relationship exceeds the Bonferroni-corrected significance threshold ($\alpha = 0.0167$), providing strong evidence for a systematic relationship between data availability and model effectiveness.

Figure 26: F1-macro, recall, and precision fitted against log(sample size);all metrics plotted against log(sample size)

Similarly, recall performance demonstrates an even stronger correlation with dataset size ($\rho = 0.6978$, $p = 0.008001$), suggesting that the model's ability to correctly identify positive instances improves markedly with increased training examples. This finding aligns with machine learning theory, where larger datasets typically enable better boundary definition and reduced overfitting, particularly for minority classes.

However, precision shows a more complex relationship ($\rho = 0.5495$, $p = 0.051771$), failing to reach statistical significance after multiple comparison correction, indicating that the relationship between dataset size and precision may be influenced by category-specific characteristics beyond sample size alone.

The extreme disparity in category representation becomes evident when examining the logarithmic distribution of sample sizes across categories. The analysis

reveals that categories with fewer than 2,000 samples (lgbt, lingerie, selfharm, sexedu) demonstrate highly variable performance patterns, with some achieving surprisingly strong results despite limited data. Notably, sexedu and lingerie achieve F1 scores of 0.9292 and 0.9267 respectively, suggesting that certain categories may be inherently more discriminable or contain higher-quality features that compensate for limited sample sizes. Dataset size by category compared with performance metrics by category.

Conversely, categories with intermediate sample sizes (2,000-6,000 samples) show more consistent underperformance, particularly violence ($F1 = 0.8382$), abortion ($F1 = 0.8469$), and selfharm ($F1 = 0.8305$). This pattern suggests a complex, non-linear relationship where moderate increases in sample size may not uniformly translate to performance improvements, potentially due to inherent content complexity within these specific categories.

Figure 27: F1-macro score vs log(sample size) against regression line with 95% CI.

The linear regression analysis quantifies the relationship between log sample size and F1 performance with $R^2 = 0.2871$, meaning that the model explains approximately 28.7 of the variance in F1 performance through sample size alone, indicating that while dataset size represents a meaningful predictor of performance, other factors account for the majority of performance variation. The positive slope coefficient (0.0317) confirms that each unit increase in log sample size corresponds to an expected 0.0317 improvement in F1 score, providing a quantitative basis for data collection prioritization.

The regression confidence intervals reveal substantial uncertainty around the prediction line, particularly for categories with extreme sample sizes. This un-

certainty underscores the complexity of the relationship and suggests that simple data augmentation may not uniformly resolve performance disparities across all categories. The relatively low R^2 value suggests that category-specific factors such as content complexity, feature discriminability, and linguistic characteristics play substantial roles in determining classification performance beyond mere sample quantity.

These statistical findings provide strong empirical evidence for prioritizing strategic data collection efforts, particularly for underrepresented categories showing both small sample sizes and suboptimal performance. The significant correlations ($p < 0.01$) for both F1 and recall metrics demonstrate that targeted data augmentation can yield measurable performance improvements. However, the analysis also reveals that a purely quantitative approach to data collection may be insufficient, as evidenced by the strong performance of some small categories and the variable relationship observed in precision metrics.

6.6 Implementation Considerations

The implementation of our enhanced Chinese Web Classification Model reveals several critical considerations that balance technical constraints with real-world applicability. Model assumptions present both challenges and opportunities within our specific context. While the dataset exhibits significant multicollinearity due to the inherent nature of overlapping content categories—such as violence and weapons, adult content and sex education, or adult material and lingerie—our preprocessing pipeline effectively mitigates these concerns through strategic feature selection. The combination of TextRank and TF-IDF-based feature extraction substantially reduces dimensionality while preserving discriminative power, transforming a potentially problematic characteristic into a manageable constraint. However, logistic regression's susceptibility to overfitting, particularly given our loss function formulation and the sparse nature of certain harmful categories, remains a limitation that requires ongoing monitoring through our production deployment metrics.

The practical impact of this classification system extends far beyond technical performance metrics, directly affecting the digital safety of hundreds of thousands of students across Taiwan's K-12 education system. Our decision threshold analysis reflects this responsibility through category-specific calibration: maintaining higher thresholds for definitively inappropriate categories like games, adult content, and sex education to ensure robust filtering, while implementing lower thresholds for self-harm related content. This nuanced approach acknowledges that self-harm detection serves a dual purpose, feeding into our ActivePulse wellness monitoring system where false negatives could have severe consequences for student wellbeing, making sensitivity more critical than specificity in this particular domain.

From a cost-benefit perspective, the local deployment architecture of CWCM provides significant advantages over alternative approaches. The entirely client-

side implementation eliminates ongoing API costs while maintaining privacy and reducing latency—critical factors when processing the browsing activities of hundreds of thousands of users simultaneously. This approach represents substantial cost savings compared to monolithic cloud-based solutions or LLM API calls that would introduce prohibitive overhead and scalability challenges. While logistic regression offers implementation simplicity comparable to LinearSVM, it serves as our baseline model for comparison, with production performance metrics ultimately determining the optimal approach for our secondary detection layer development. The real-world validation of these modeling decisions will emerge through continuous monitoring of classification accuracy, user feedback, and administrator responses in our deployed environment, informing the iterative refinement necessary for maintaining effective content filtering in Taiwan's evolving digital education landscape.

7 Ethical Considerations

The development and deployment of automated content filtering systems in educational environments raises significant ethical considerations encompassing privacy implications, potential biases, cultural sensitivity, and the balance between digital safety and access to information. The CWCM operates primarily through client-side processing to minimize data exposure, yet still requires access to web content for classification tasks, representing a form of digital surveillance managed through privacy-protective measures including zero-trust data handling that ensures personally identifiable information is not linked to browsing patterns. Our pursuit of ISO 27001 certification demonstrates commitment to maintaining robust information security standards, though technical safeguards alone are insufficient, requiring transparency with school administrators about system operation, data collection practices, and decision-making processes to maintain trust and ensure informed consent.

Machine learning systems inevitably reflect biases present in their training data and development processes, with our dataset constructed primarily from web content identified through automated collection methods and manual labeling processes that may introduce systematic biases, complicated by the cultural and linguistic context of Traditional Chinese content that may not represent the full diversity of perspectives within Taiwan's educational community. Of particular concern is the system's potential for differential performance across content categories, as evidenced by our statistical analysis showing significant performance variations correlated with dataset size, where categories with limited training data may experience higher rates of misclassification, potentially leading to inconsistent filtering decisions. To mitigate these risks, we have implemented ongoing performance monitoring across all content categories and maintain detailed classification logs enabling retrospective analysis of system decisions, with the client-side deployment architecture allowing for category-specific threshold adjustments that enable educational administrators to calibrate system sensi-

tivity based on local context and institutional policies.

The classification of content as "harmful" or "appropriate" is inherently subjective and culturally dependent, with our category definitions reflecting contemporary Taiwanese educational policy and cultural norms that may evolve over time and may not align with all community perspectives. We acknowledge that different educational contexts may require different approaches to content filtering, and the system's effectiveness depends on its alignment with local educational objectives and community values, addressed through the modular architecture of CWCM that allows for customization of category definitions and classification thresholds, enabling institutions to adapt the system to their specific needs while maintaining the underlying technical infrastructure.

Content filtering systems face the fundamental challenge of balancing student protection against the risk of over-censorship that could limit educational opportunities and intellectual freedom, where overly restrictive filtering may prevent students from accessing legitimate educational resources, while insufficient filtering may expose students to inappropriate content. This is addressed through several mechanisms including the dual-layer architecture (CWCM and planned CWCM-V2) designed to provide nuanced classification that can distinguish between different types of content within the same domain, category-specific threshold systems that allow for fine-tuned control over filtering sensitivity, and most importantly, the system's design to support rather than replace human judgment, providing administrators and educators with tools to make informed decisions about content access.

The imperative for effective content filtering extends beyond technical considerations to encompass well-documented public health concerns, as extensive research demonstrates the profound impact of inappropriate digital content exposure on student academic performance, mental health, and cognitive development. Violence exposure in educational contexts has been shown to significantly impair academic achievement through disrupted school attachment and reduced motivation to succeed, while also causing sleep disturbances that further compromise learning outcomes (Lepore & Kliewer, 2013; Sonsteng-Person et al., 2023). The proliferation of digital media has introduced additional challenges, with problematic internet use emerging as a growing concern for adolescent health and well-being in our increasingly connected world (Dadi et al., 2024).

Recent health advisories from major international organizations, including the World Health Organization's warnings about teens, screens, and mental health, and the U.S. Surgeon General's advisory on social media and youth mental health, underscore the critical nature of these concerns at a policy level (Office of the Surgeon General, 2023; World Health Organization, 2024). The phenomenon of "brain rot" caused by digital overload has gained recognition as a legitimate threat to cognitive functioning, with research indicating that excessive exposure to low-quality digital content can impair attention, memory, and critical thinking skills essential for academic success (*Demystifying the New Dilemma of Brain Rot in the Digital Era*, 2025; Lotkowski, 2025; Shanmuga-

sundaram & Tamilarasu, 2023). While some studies have identified potential positive effects of certain types of digital media under specific conditions (Egami et al., 2024), the overwhelming consensus in the literature emphasizes the need for careful curation of digital environments to protect developing minds from harmful content (Heiden et al., 2019). These findings provide compelling justification for implementing sophisticated content filtering systems like CWCM, as the educational and psychological benefits of protecting students from inappropriate content far outweigh the minimal privacy considerations involved in automated classification systems designed with robust data protection measures.

8 Future Work

The evolution of CWCM encompasses immediate technical enhancements and longer-term architectural improvements. Our immediate priorities focus on deploying CWCM-V2 as a secondary detection layer using transformer-based architectures to capture contextual information that TF-IDF approaches miss. This dual-layer system will enable CWCM to operate in high-recall mode while CWCM-V2 provides precision refinement for challenging categories including self-harm, violence, and weapons.

Production evaluation will inform critical architectural decisions through systematic comparison of Logistic Regression against LinearSVC and XGBoost under real-world conditions. Key areas for investigation include one-vs-rest versus multinomial classification strategies, ensemble methods combining multiple linear models, and detailed analysis of sigmoid versus softmax decision functions. The integration of online learning mechanisms will enable continuous model adaptation based on administrator feedback and emerging content patterns.

Data collection improvements address current dataset limitations through tiered collection strategies prioritizing categories with significant sample size-performance correlations. The training data contains substantial noise, including non-Chinese datapoints, which requires systematic cleaning to improve CWCM-V2 performance during fine-tuning. Automated data collection pipelines will target underperforming categories, while considerations for user-based data collection will expedite the acquisition process. With CWCM-V2 deployment, CWCM requirements shift from high F1-score performance to high recall performance, necessitating continued experimentation with linear model architectures for optimal integration with transformer-based attention models.

The system's long-term development includes multilingual support extending beyond Traditional Chinese, multimodal analysis capabilities for harmful content in images and videos, and federated learning frameworks enabling knowledge sharing across educational institutions while maintaining privacy requirements. Integration with large language models for content understanding, real-time sentiment analysis for early intervention, and predictive modeling for proactive

content curation will transform reactive filtering into anticipatory protection. The architecture will evolve to support personalized learning pathways balancing content accessibility with age-appropriate safeguards while maintaining integration with existing educational technology infrastructure.

This development roadmap positions CWCM as an adaptive content filtering system capable of evolving with educational needs while maintaining effective protection of digital learning environments.

9 Conclusion

This study successfully developed a comprehensive enhancement to the Chinese Web Classification Model (CWCM), addressing critical limitations in content filtering for educational environments through systematic data collection, rigorous algorithmic evaluation, and statistical validation. The project constructed a substantial dataset of 429,066 Chinese web content samples with focused augmentation of harmful categories including drugs, tobacco, and weapons, establishing a robust foundation for machine learning-based content classification in educational technology applications.

Our methodology evaluated over 34 machine learning algorithms through rigorous 10-fold cross-validation, employing sophisticated preprocessing techniques including Jieba-based text segmentation and custom TF-IDF vectorization with strategically extracted feature dictionaries totaling 4,783 features. The optimized Logistic Regression model achieved a macro-averaged F1-score of 0.9183, demonstrating statistically significant superior performance compared to baseline models including XGBoost ($p < 0.000001$) and LinearSVC ($p < 0.000001$). Statistical correlation analysis revealed strong positive relationships between dataset size and performance metrics ($\rho = 0.6868$ for F1-score, $p = 0.009$), providing empirical evidence for strategic data collection priorities and informing future model development approaches.

The technical implementation successfully addresses critical deployment constraints through client-side JavaScript architecture, enabling real-time inference on resource-limited devices while maintaining privacy and eliminating API dependencies. This approach represents substantial advantages over cloud-based solutions in terms of cost, latency, and scalability for educational institutions serving hundreds of thousands of students. The model's multinomial classification approach and category-specific threshold calibration demonstrate practical considerations for balancing precision and recall across diverse content categories, with particular attention to sensitive areas such as self-harm detection for student wellness monitoring.

While the enhanced CWCM model represents significant theoretical and practical improvements over existing approaches, definitive performance validation requires extensive production deployment and AB-testing in real educational environments. The project's comprehensive analysis of traditional machine

learning approaches—including support vector machines, boosted trees, and linear methods—demonstrated that statistical approaches achieve substantial classification performance but remain limited in capturing complex semantic relationships inherent in natural language content. This finding establishes a clear foundation for future work incorporating transformer-based architectures in CWCM-V2, designed to complement the current linear model through enhanced contextual understanding.

The broader implications of this research extend beyond technical achievement to address fundamental challenges in educational technology deployment, digital safety, and ethical AI implementation in sensitive environments. By providing an effective, deployable solution for Chinese content filtering that balances accuracy, computational efficiency, and practical implementation requirements, this work contributes meaningfully to protecting digital learning environments while maintaining educational accessibility and intellectual freedom. The established methodology and findings provide a replicable framework for content classification systems in diverse linguistic and cultural contexts, supporting the global expansion of effective educational technology solutions.

Appendices

9.1 Libraries Used

import os import time import random from tqdm import tqdm from selenium import webdriver from selenium.webdriver.chrome.service import Service from selenium.webdriver.chrome.options import Options from selenium.webdriver.common.by import By from selenium.webdriver.support.ui import WebDriverWait from selenium.webdriver.support import expected_conditions as EC from urllib.parse import quote_plus

from app.scrape_logger import ScrapeLogger from app.browser import Browser, user_agents

import os import sys import base64 import requests from tqdm import tqdm import concurrent.futures from bs4 import BeautifulSoup from dotenv import load_dotenv from urllib.parse import urlparse

from openai import OpenAI from google import genai from google.genai import types

from error_logger import ErrorLogger from extractor import extract_text, is_valid_website from topics import topic_dict_small, topic_dict_medium, topic_dict_max

References

Dadi, A. F., Dachew, B. A., & Tessema, G. A. (2024). Problematic internet use: A growing concern for adolescent health and well-being in a digital era. *Journal of Global Health*, *14*, 03034. https://doi.org/10.7189/jogh.14.03034

Demystifying the new dilemma of brain rot in the digital era. (2025). https://pmc.ncbi.nlm.nih.gov/articles/PMC11939997/

Egami, H., Rahman, Md. S., Yamamoto, T., Egami, C., & Wakabayashi, T. (2024). Causal effect of video gaming on mental well-being in japan 2020–2022. *Nature Human Behaviour*, *8*(10), 1943–1956. https://doi.org/10.1038/s41562-024-01948-y

Heiden, J. M. von der, Braun, B., Müller, K. W., & Egloff, B. (2019). The association between video gaming and psychological functioning. *Frontiers in Psychology*, *10*, 1731. https://doi.org/10.3389/fpsyg.2019.01731

Lepore, S. J., & Kliewer, W. (2013). Violence exposure, sleep disturbance, and poor academic performance in middle school. *Journal of Abnormal Child Psychology*, *41*(8), 1179–1189. https://doi.org/10.1007/s10802-013-9709-0

Lotkowski, S. (2025). *Brain rot explained: How digital overload affects your mind*. Inspira Health Network. https://www.inspirahealthnetwork.org/news/healthy-living/brain-rot-explained-how-digital-overload-affects-your-mind

Office of the Surgeon General. (2023). *Social media and youth mental health: The u.s. Surgeon general's advisory*. U.S. Department of Health; Human Services. https://www.hhs.gov/surgeongeneral/reports-and-publications/youth-mental-health/social-media/index.html

Shanmugasundaram, M., & Tamilarasu, A. (2023). The impact of digital technology, social media, and artificial intelligence on cognitive functions: A review. *Frontiers in Cognition*, *2*, 1203077. https://doi.org/10.3389/fcogn.2023.1203077

Sonsteng-Person, M., Jaggers, J. W., & Loomis, A. M. (2023). Academic achievement after violence exposure: The indirect effects of school attachment and motivation to succeed. *Journal of Child and Adolescent Trauma*, *16*(3), 717–729. https://doi.org/10.1007/s40653-023-00546-w

World Health Organization. (2024). *Teens, screens and mental health*. https://www.who.int/europe/news/item/25-09-2024-teens--screens-and-mental-health