Final Project Report: Identifying Predatory Lending Practices

By: Karrington Riley & Oluwaseun Ademiloye

Introduction

Predatory lending refers to practices that impose unfair or abusive loan terms on borrowers. Payday loans, in particular, are a common vehicle for these practices and disproportionately impact low-income and marginalized communities. Our goal is to build a machine learning pipeline to classify loan agreements, online posts, and consumer complaints as either predatory or non-predatory based on their text content.

This project uses natural language processing (NLP) techniques, traditional machine learning models, and transformer-based models (BERT) to tackle this classification problem. By comparing model performance, analyzing feature importance, and integrating multiple data sources, we aim to develop a tool that can aid regulators, researchers, and consumers in identifying problematic lending behavior.

Data Sources

We curated and processed data from the following sources:

Reddit: Posts from r/personalfinance, r/Debt, r/loans, and more discussing payday loan experiences (labeled as predatory)

CFPB Complaints: A small sample of consumer complaints about payday loans and similar products

Predatory Loan Agreements: Scraped from watchdog and regulatory sites

Fair Loan Agreements: Sourced from credit unions and nonprofits

These texts were cleaned, deduplicated, and labeled to build a diverse dataset.

Methodology

Preprocessing

* Removed known keyword leakage (e.g., "payday", "loanmart")
* Lowercased and tokenized text
* Deduplicated text samples
* Balanced datasets to avoid class imbalance

Traditional ML Models

* TF-IDF Vectorization (top 5,000 features, unigrams and bigrams)
* Models:
* Naive Bayes
* Logistic Regression
* Support Vector Machine
* Random Forest (with and without grid search tuning)
* Evaluated using precision, recall, F1-score, and confusion matrices

BERT Models

* Used HuggingFace Trainer API (Seun's notebook) and PyTorch manual training loop (Karrington's Colab notebook)
* Data tokenized using bert-base-uncased tokenizer
* Trained for 3–4 epochs on binary classification task
* Evaluated on validation set using classification metrics and confusion matrix

Results

Traditional ML Model Comparison (TF-IDF)

\*\*\*insert model evaluation\*\*\*

Feature importance analysis revealed that high-weight features for predatory loans included terms like "fees", "advance", and "cash", while non-predatory loans often included "savings", "credit union", and "terms".

BERT Results

* Validation Accuracy: 100% (on small test sets, likely overfit)
* Confusion Matrix: Perfect classification for both predatory and non-predatory classes

While BERT showed strong performance, the small sample size makes it difficult to fully trust the generalization. Further testing on held-out data is needed.

Challenges

Overfitting: Many models, especially BERT, showed unrealistically high performance, suggesting overfitting.

Keyword Leakage: Removing terms like "payday" and "loanmart" was essential to avoid inflated performance.

Large File Sizes: Managing transformer checkpoints caused Git issues; we ultimately excluded model weights from version control.

Conclusions

Our system successfully distinguishes between fair and predatory loan texts using both traditional and transformer-based NLP models.

Classical models (like SVM and Logistic Regression) performed surprisingly well when properly cleaned and balanced.

BERT showed promise but requires more rigorous testing with larger datasets to validate its generalization.

Future Work

Integrate full CFPB complaint dataset

Explore document summarization before classification

Use NER to extract problematic terms (e.g., interest rates, hidden fees)

Develop a lightweight web tool for real-time document flagging

Repository Overview

notebooks/ – Main analysis notebooks (\*\*list the main notebooks when finalized\*\*\*)

data/processed/ – Cleaned datasets used in experiments

scripts/ – Scraping and preprocessing scripts

reports/ – This report and final presentation

Run instructions are in the updated README.md.