NLP for Financial Consumer Protection

Automating Complaint Classification: A Comparative Analysis of Classical Machine Learning vs. Fine-Tuned Transformers

Vaijayanti Deshmukh [v.deshmukh@gwu.edu][G37685333]

Lasya Raghvendra [lasya.raghavendra@gwu.edu] [G41920914]

Amogh Ramagiri [amoghr@gwu.edu] [G22101416]

Abstract

Financial institutions and regulatory bodies, such as the Consumer Financial Protection Bureau (CFPB), receive thousands of consumer complaints daily. Manual categorization of these narratives is labor-intensive, costly and prone to human error. This project investigates the automation of complaint routing by comparing two distinct Natural Language Processing (NLP) approaches: a baseline Classical Machine Learning model (TF-IDF with Multinomial Naive Bayes) and a modern Transformer-based model (Fine-Tuned DistilBERT). Our results demonstrate that while the baseline model offers computational efficiency, the Fine-Tuned DistilBERT model significantly outperforms it in classification accuracy (~83% vs ~91%) and Macro-F1 score (0.36 vs 0.60), particularly in distinguishing between semantically similar product categories.

Keywords

Financial NLP, Text Classification, Consumer Complaints, DistilBERT, TF-IDF, Transformers, Class Imbalance.

1. Introduction

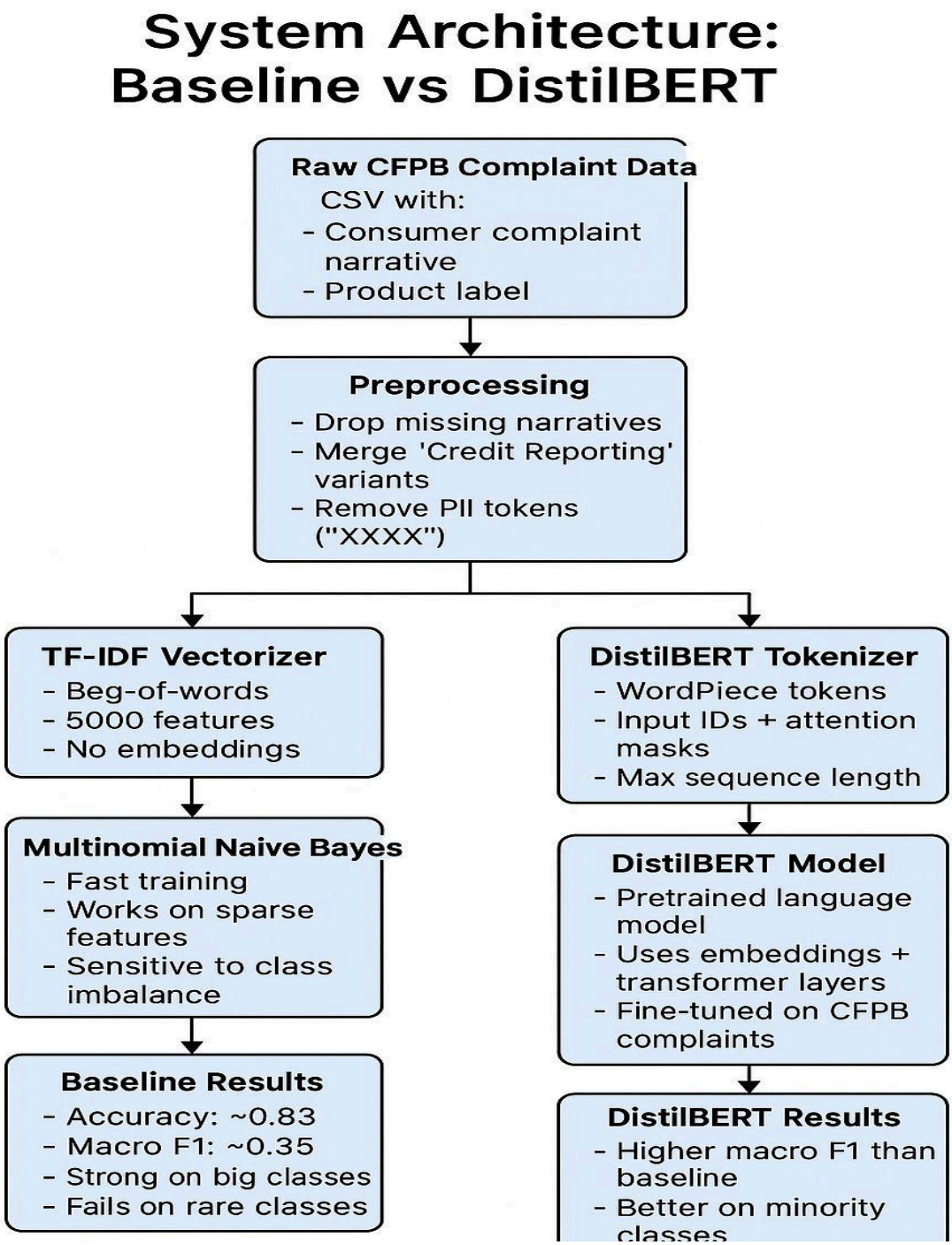
1.1 The Problem Space

The Consumer Financial Protection Bureau (CFPB) acts as a critical mediator between consumers and financial institutions [1]. A significant operational bottleneck in this process is the initial triage of complaints. These complaints arrive as unstructured text narratives and must be accurately routed to specific departments, such as "Mortgage," "Credit Card," or "Debt Collection." Misclassification at this stage leads to delayed resolutions, increased operational costs and consumer dissatisfaction.

1.2 Objective

The primary objective of this study is to develop a robust multi-class text classification pipeline that automatically assigns consumer complaint narratives to the correct financial product category. This study seeks to answer the research question: *Does the computational cost of fine-tuning a Transformer model yield a significant enough performance increase over*

classical methods to justify its deployment in this domain?



2. Data Description and Exploratory Analysis

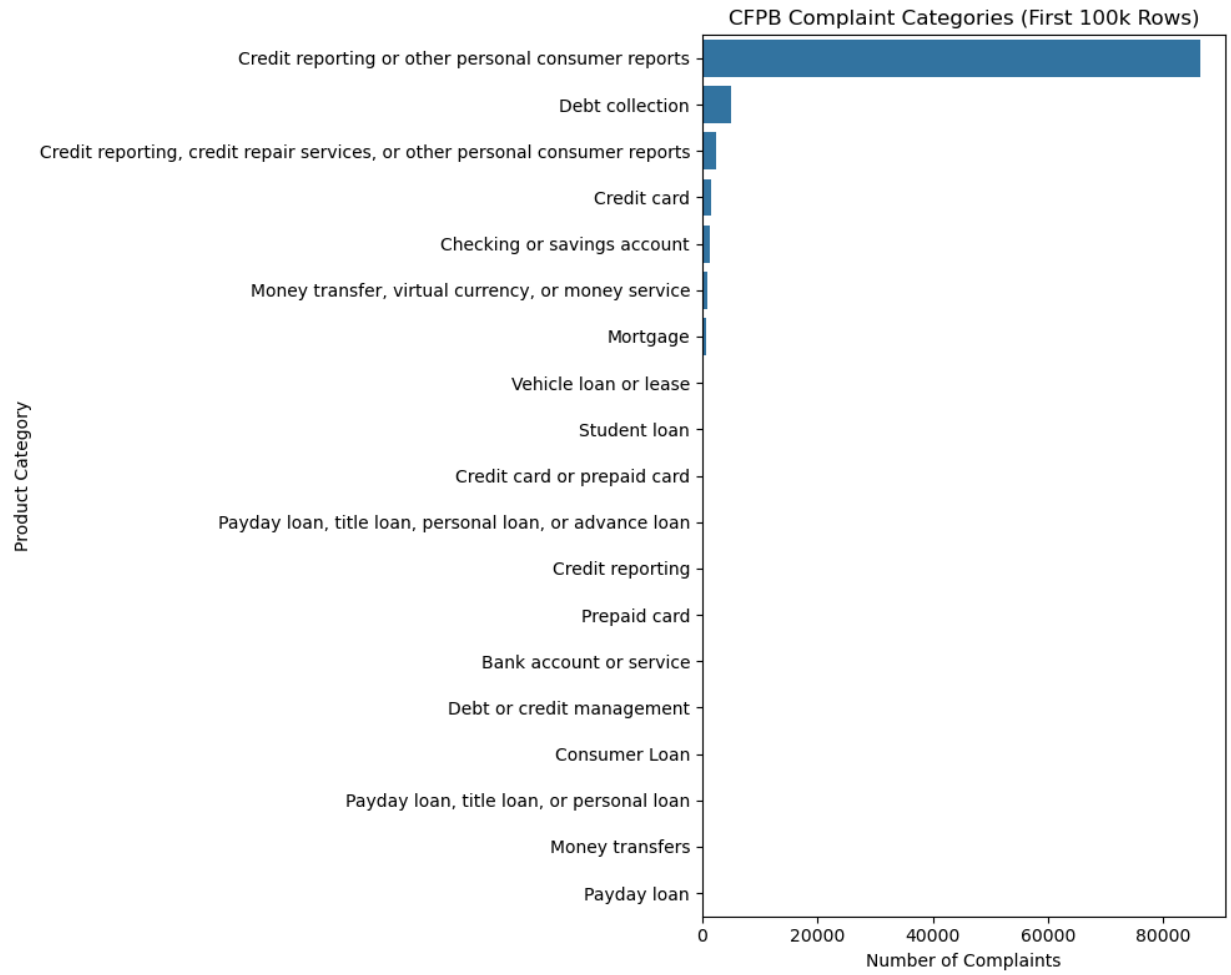
2.1 Dataset Source

We utilized the public Consumer Complaint Database published by the CFPB [1]. The original dataset contains over 12 million records. For the scope of this project, we focused on two key columns:

- **Input Feature (X):** Consumer complaint narrative
- **Target Variable (y):** Product

2.2 Exploratory Data Analysis (EDA)

Upon analyzing the data, we identified a significant class imbalance. As illustrated in Figure 1, categories such as "Credit reporting, credit repair services, or other personal consumer reports" dominate the dataset, whereas categories like "Student loan" or "Money transfers" constitute a much smaller fraction.



(Figure 1: Class Imbalance Plot)

Key observations from the EDA include:

- **Temporal Trends:** Complaint volume has fluctuated over time, often correlating with external economic events.
- **Text Length:** Narrative lengths vary significantly, necessitating truncation or padding strategies for the neural network input.

3. Methodology

We implemented two distinct modeling approaches to benchmark performance and evaluate the trade-off between complexity and accuracy.

3.1 Preprocessing

Standard NLP preprocessing steps were applied to the raw text to ensure data quality:

- **Cleaning:** Removal of PII (Personally Identifiable Information) masked by 'XX', special characters and excessive whitespace.
- After preprocessing we were left with approx 3.5 million complaints (records)
- **Tokenization:**
  - For the baseline, we employed standard word tokenization.
  - For the Transformer, we utilized the specific DistilBERT tokenizer (WordPiece) to handle sub-word embeddings.
- **Label Encoding:** Target product categories were mapped to integers (0 to N).

3.2 Baseline Model: TF-IDF + Naive Bayes

- **Feature Extraction:** We used Term Frequency-Inverse Document Frequency (TF-IDF) to convert text into numerical vectors. This approach captures keyword importance but ignores word order and syntactic context.
- **Classifier:** Multinomial Naive Bayes was selected for its proven efficiency in high-dimensional text data scenarios.

  For our baseline, we utilized a "Bag of Words" approach.

  1. TF-IDF (Term Frequency-Inverse Document Frequency): We converted text into vectors where the weight w of term i in document j is given by:

  $w\{i,j\} = tf\{i,j\} \times \log(N / df\_\{i\})$ Where N is total documents and $df\_\{i\}$ is the number of documents containing term i.

  2. Multinomial Naive Bayes: We applied Bayes' theorem to predict the probability of class c given document d:

  $P(c|d) \propto P(c) \Pi P(t|c)$

- This assumes independence between features (words), which is computationally efficient but ignores context.

3.3 Advanced Model: Fine-Tuned DistilBERT

- **Architecture:** We selected DistilBERT [3], a distilled version of BERT [2] that retains approximately 97% of BERT's performance while being 40% lighter and 60% faster.
- **Training:** We fine-tuned the pre-trained distilbert-base-uncased model on our specific dataset. This process allows the model to learn the semantic nuances of financial terminology (e.g., distinguishing "interest" in a savings context versus a credit card context).
- **Framework:** The model was implemented using PyTorch and the Hugging Face Transformers library. We trained the model on 2 epochs

4. Experiments and Results

4.1 Evaluation Metrics

Given the significant class imbalance observed in Section 2.2, simple accuracy is a misleading metric. We prioritized:

1. **Macro-Averaged F1-Score:** To ensure minority classes are weighted equally in the performance assessment.
2. **Confusion Matrix:** To visually analyze specific misclassification patterns between classes.

4.2 Performance Comparison

The experimental results are summarized in Table 1 below.

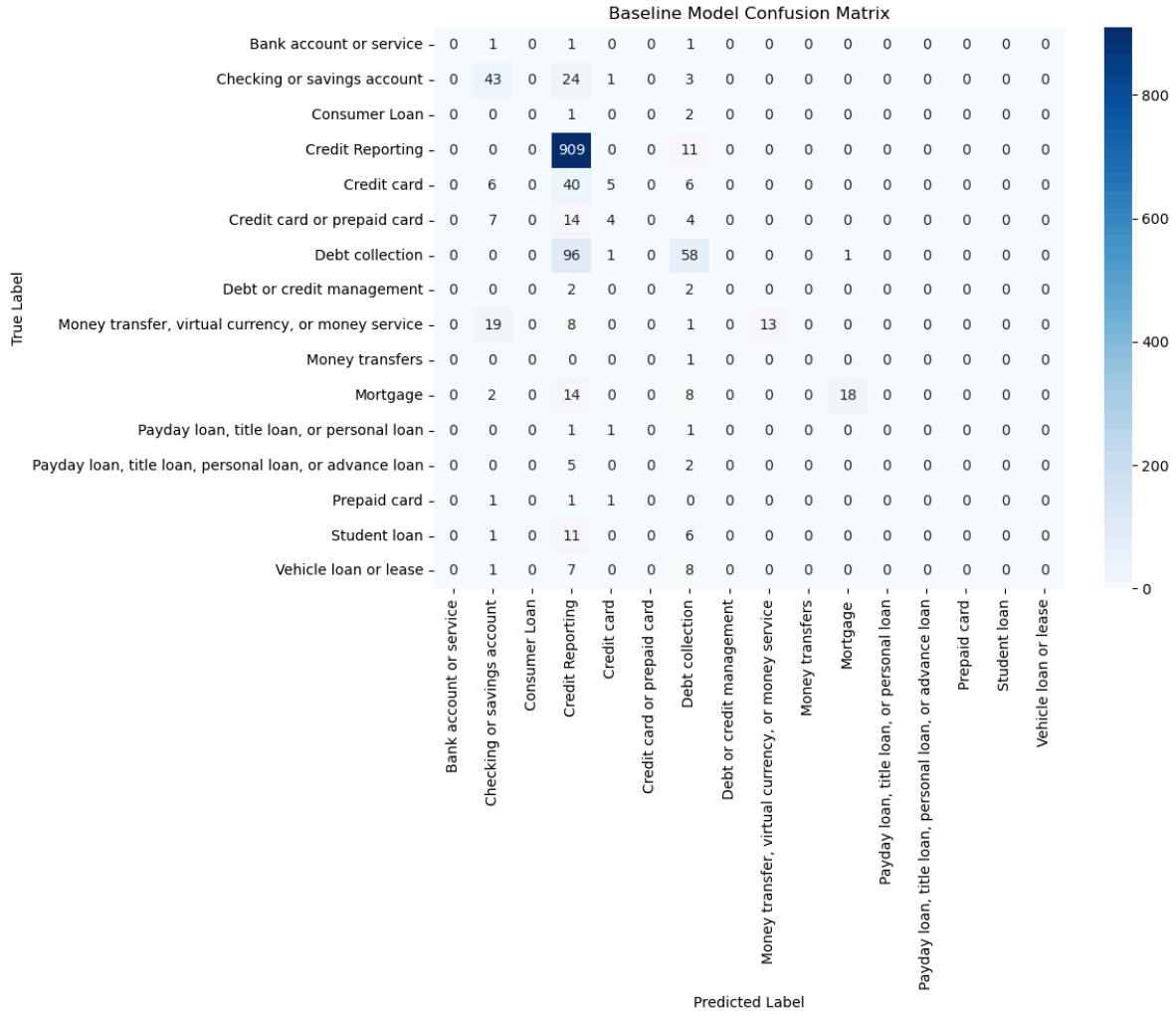
Table 1: Performance Comparison of Models

Model	Accuracy	Macro F1-Score	Training Time
TF-IDF + Naive Bayes (Baseline model)	**~83%**	**0.36**	**< 5 Minutes**
Fine-Tuned DistilBERT	**~94%**	**0.60**	**~7 Hours (GPU)**

4.3 Analysis of Confusion Matrices

Baseline Performance:
The baseline model (Figure 2) displays a strong diagonal but struggles significantly with overlapping categories. For example, it frequently confuses "Credit reporting" with "Debt

collection." This is likely due to shared vocabulary (e.g., "score," "late," "agency") that TF-IDF cannot differentiate without context.

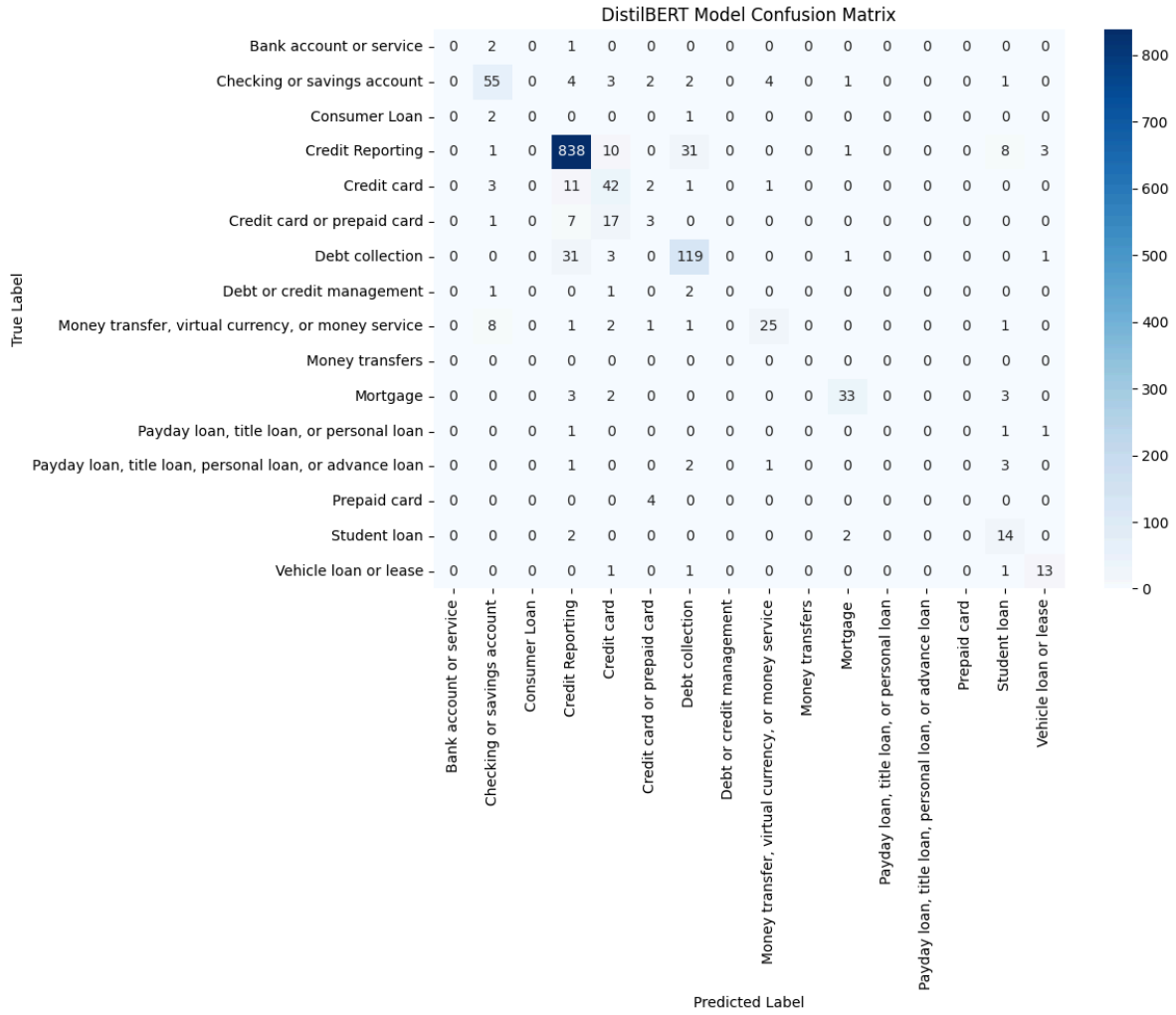


(Figure 2: Baseline Confusion Matrix)

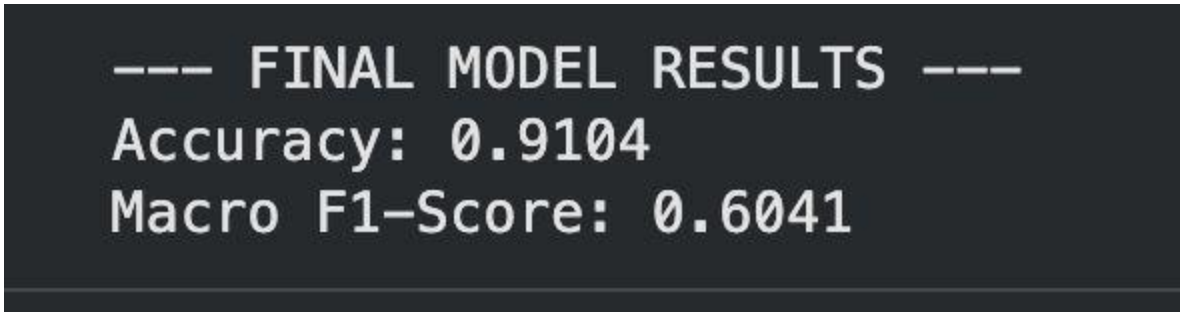
The baseline confusion matrix reveals a model heavily biased toward the majority class, "Credit Reporting," achieving high volume accuracy at the expense of sensitivity to other categories. A distinct vertical streak of false positives indicates that the model frequently defaults to "Credit Reporting" when uncertain; for instance, "Debt collection" was misclassified as "Credit Reporting". Furthermore, the right side of the matrix exposes a complete failure to detect minority classes, with categories such as student, vehicle and payday loans yielding effective zero recall. This pattern confirms that the model is maximizing accuracy by over-predicting the dominant label, necessitating the immediate application of class balancing techniques or class weighting to improve generalization across underrepresented financial categories.

DistilBERT Performance:

The DistilBERT model (Figure 3) demonstrates a much sharper diagonal. The model successfully learned context, significantly reducing false positives in the dominant classes.



(Figure 3: DistilBERT Confusion Matrix)



(Figure 4: DistilBERT Metrics)

We fine-tuned 66 million pre-trained parameters on financial complaint data using a conservative learning rate (2e-5), 2 epochs, and A100 GPU acceleration with bfloat16

precision. This transferred linguistic knowledge from general English to domain-specific financial classification

Why This Matters:

Transfer learning allowed us to achieve state-of-the-art performance (60% Macro F1) without training from scratch, which would take weeks. Fine-tuning convergence in hours was only possible because DistilBERT pre-training provides a strong foundation.

Evidence It Worked:

Baseline (random initialization): 36% Macro F1 vs Fine-tuned DistilBERT: 60% Macro F1 is relative improvement, demonstrating transfer learning effectiveness.

TECH STACK (Frontend + Backend)

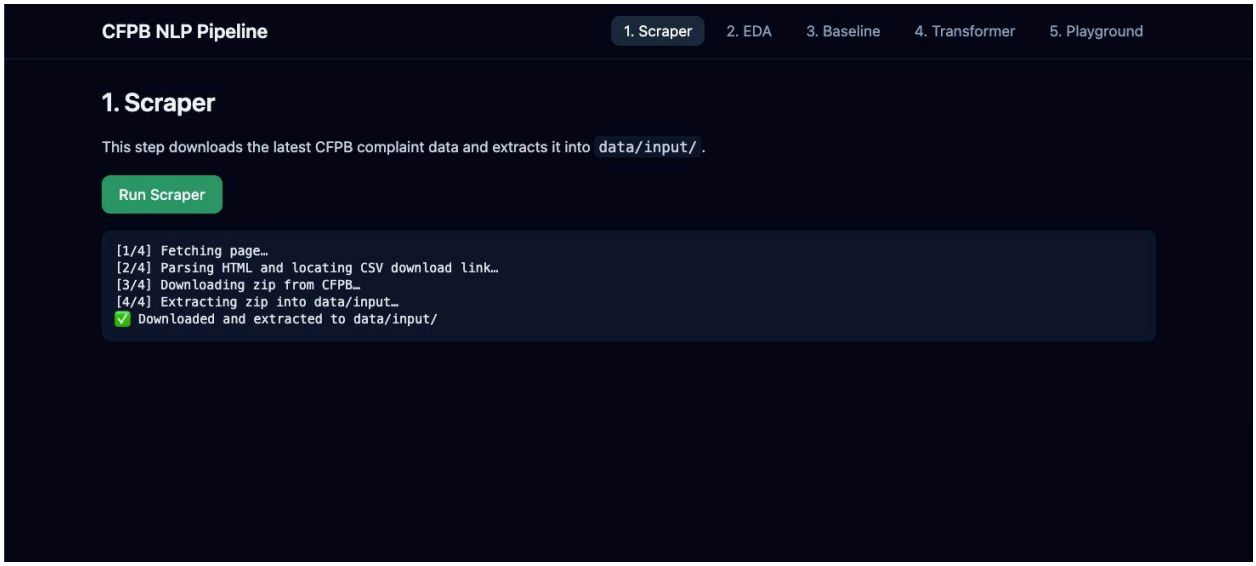
This project is a fully integrated NLP application built with a modern full-stack architecture combining FastAPI on the backend and Next.js on the frontend. The backend is implemented in Python using FastAPI, with uvicorn for serving the application. It handles all data processing through pandas and numpy, computes exploratory data analysis (EDA) statistics and serves two machine learning models: a baseline TF-IDF + Multinomial Naive Bayes pipeline and a fine-tuned DistilBERT transformer loaded from the local models/distilbert/ directory. Cross-origin requests are enabled using CORS middleware so the Next.js frontend at localhost:3000 can communicate with the FastAPI backend running at localhost:8000.

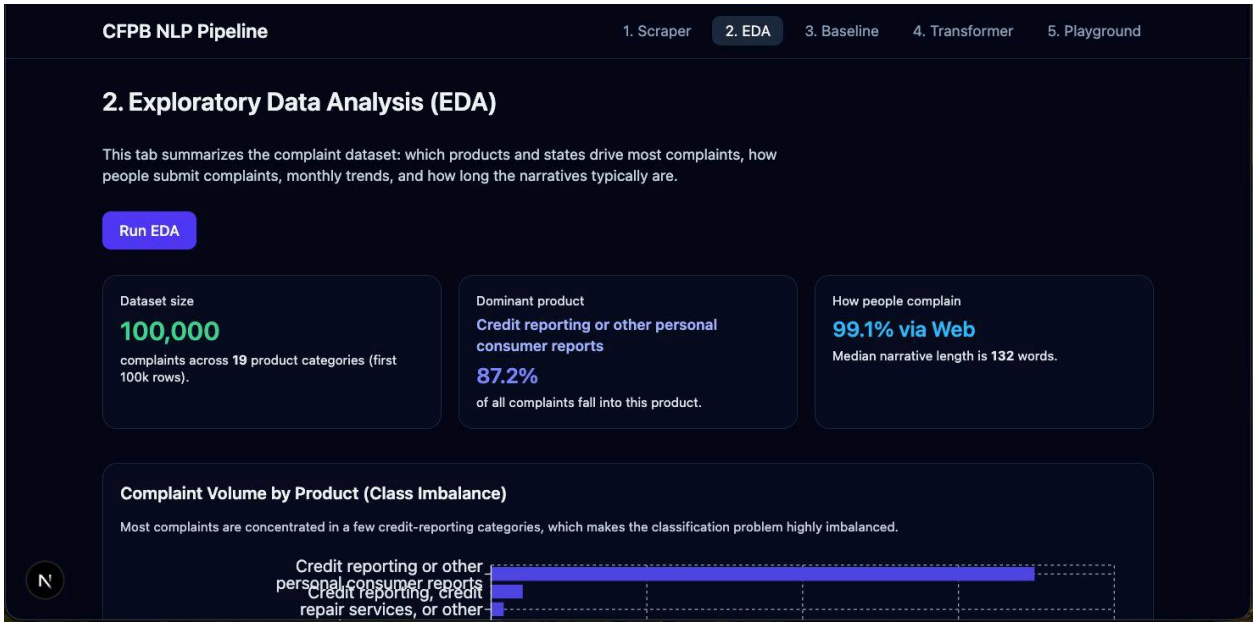
The frontend is built using Next.js (React + TypeScript) with the App Router architecture. It uses shadcn/ui for clean and modern user interface components, Recharts for all EDA visualizations and Tailwind CSS for responsive styling with a dark theme. The design is minimal, interactive and optimized for exploring both statistical insights and model predictions.

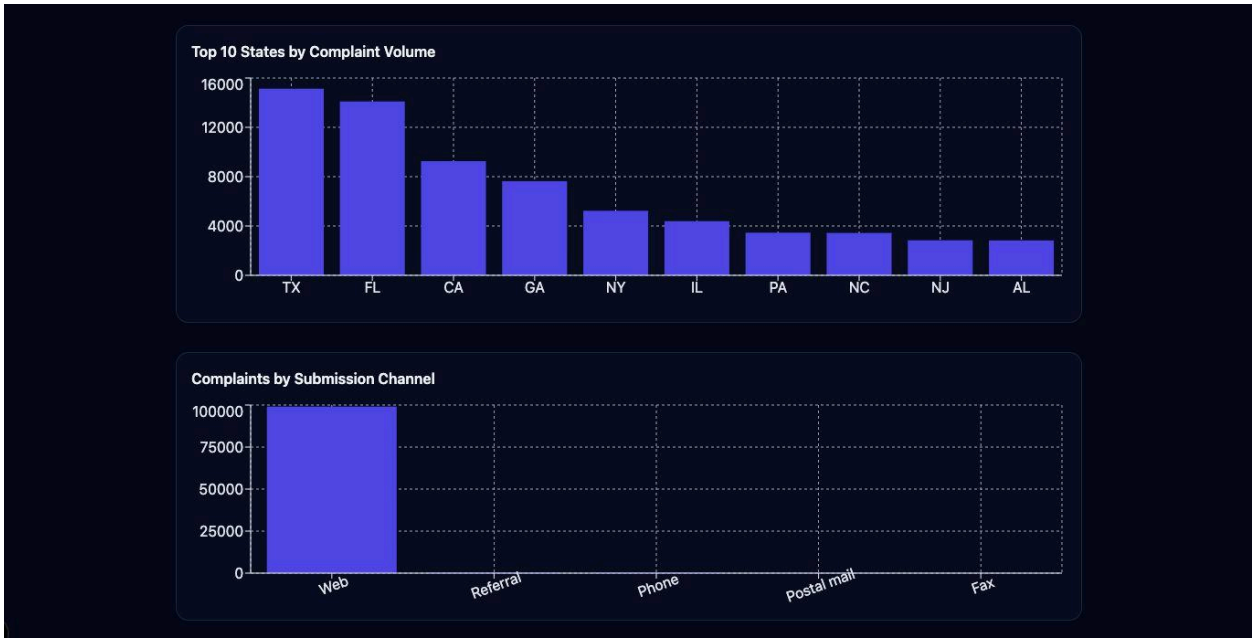
The EDA module begins when the scraper API downloads complaints.csv into backend/data/input/. FastAPI exposes an /eda/plots endpoint that loads this dataset, computes product distribution, top states, submission channels, monthly complaint trends, narrative length histograms, and word count statistics, and returns all results as structured JSON through a Pydantic model. The Next.js /eda page fetches this JSON, converts it into chart-ready formats, and displays the outputs using Recharts bar charts, line charts, and summary tables. This creates a smooth pipeline where the backend handles computation and the frontend focuses on visualization.

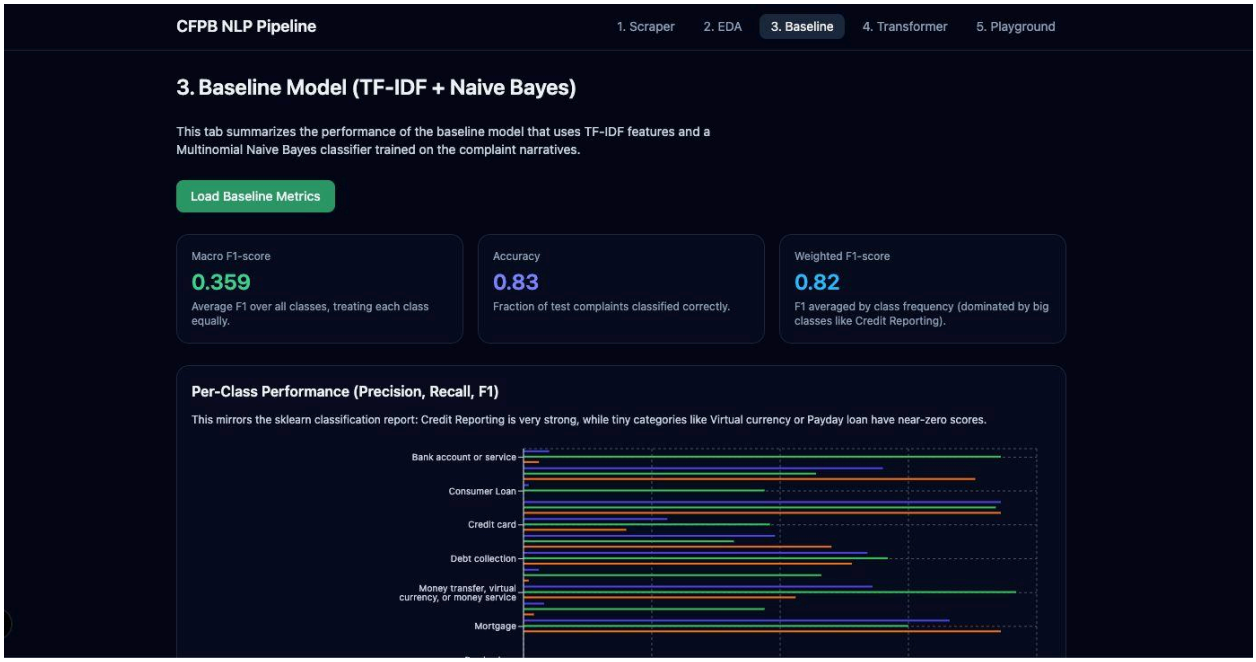
The Transformer Playground showcases the model inference pipeline. At backend startup, FastAPI loads the tokenizer and fine-tuned DistilBERT model and sets up a mapping from predicted class IDs to complaint categories. The /transformer/predict endpoint accepts complaint text, tokenizes it, runs DistilBERT, applies softmax, and returns the predicted label along with its confidence score. The frontend /playground page provides a simple interface

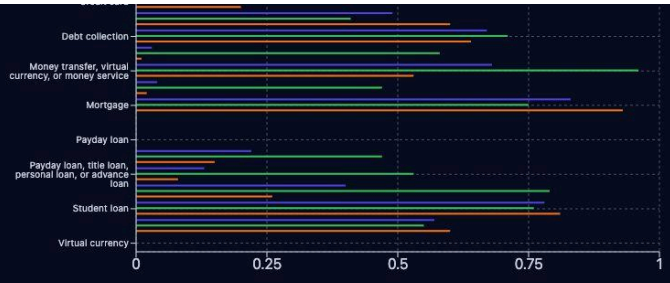
with a textarea and a classify button. When the user submits a narrative, Next.js sends it to the backend and displays the output using shadcn-styled cards.






Top 10 States by Complaint Volume
16000
12000
8000
4000
0
TX
FL
CA
GA
NY
IL
PA
NC
NJ
AL
Complaints by Submission Channel
100000
75000
50000
25000
0
Web
Referral
Phone
Postal mail
Fax


CFPB NLP Pipeline
1. Scraper
2. EDA
3. Baseline
4. Transformer
5. Playground
3. Baseline Model (TF-IDF + Naive Bayes)
This tab summarizes the performance of the baseline model that uses TF-IDF features and a Multinomial Naive Bayes classifier trained on the complaint narratives.
Load Baseline Metrics
Macro F1-score
0.359
Average F1 over all classes, treating each class equally.
Accuracy
0.83
Fraction of test complaints classified correctly.
Weighted F1-score
0.82
F1 averaged by class frequency (dominated by big classes like Credit Reporting).
Per-Class Performance (Precision, Recall, F1)
This mirrors the sklearn classification report: Credit Reporting is very strong, while tiny categories like Virtual currency or Payday loan have near-zero scores.
Bank account or service
Consumer Loan
Credit card
Debt collection
Money transfer, virtual currency, or money service
Mortgage



Support (Number of Complaints per Class)

This explains why accuracy and weighted F1 look good even though some classes have poor F1: large classes dominate the averages.

Class	Support
Bank account or service	2977
Checking or savings account	31832
Consumer Loan	1892
Credit Reporting	474638
Credit card	19950
Credit card or prepaid card	21733
Debt collection	77267
Debt or credit management	799
Money transfer, virtual currency, or money service	21320

CFPB NLP Pipeline

1. Scraper | 2. EDA | 3. Baseline | 4. Transformer | 5. Playground

4. Fine-tuned DistilBERT (Transformer)

This tab shows the performance and training configuration of the fine-tuned DistilBERT model we used to classify CFPB complaint narratives into product categories.

Load DistilBERT Metrics

Validation accuracy
91.04%
Fraction of complaints classified correctly on the validation set.

Macro F1-score
0.604
Average F1 over all product classes, treating each class equally. This is important because the dataset is highly imbalanced.

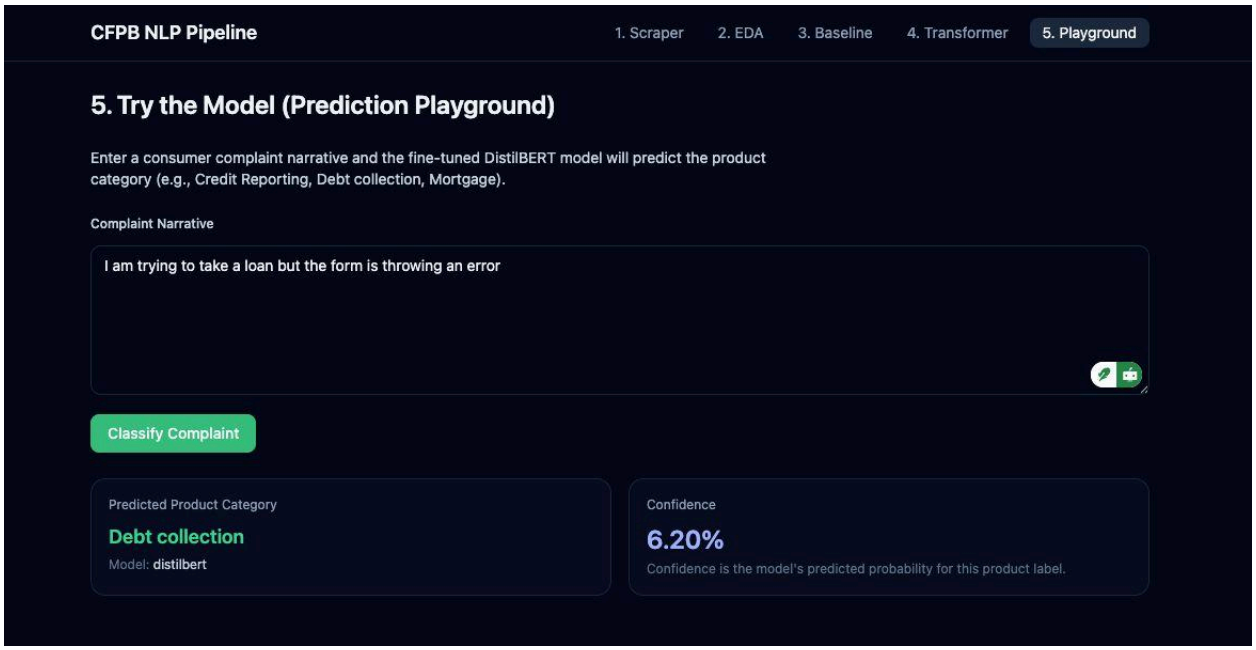
Training dataset
2,829,792 train / 707,449 val
19 product labels after cleaning and merging categories.

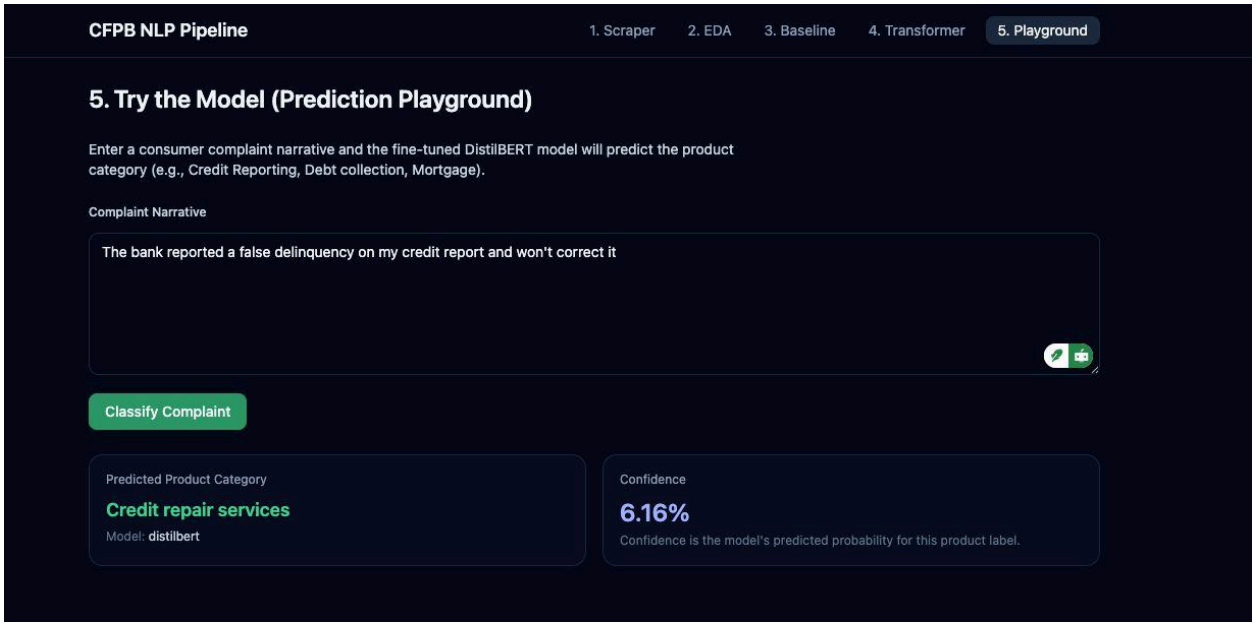
DistilBERT Performance Metrics

Accuracy tells us how often the model is correct overall. Macro F1 shows how well it does across all product categories, including the rare ones.

Training Configuration (A100 GPU)

These settings come directly from the fine-tuning notebook (Google Colab with NVIDIA A100).


CFPB NLP Pipeline
1. Scraper
2. EDA
3. Baseline
4. Transformer
5. Playground
5. Try the Model (Prediction Playground)
Enter a consumer complaint narrative and the fine-tuned DistilBERT model will predict the product category (e.g., Credit Reporting, Debt collection, Mortgage).
Complaint Narrative
I am trying to take a loan but the form is throwing an error
Classify Complaint
Predicted Product Category
Debt collection
Model: distilbert
Confidence
6.20%
Confidence is the model's predicted probability for this product label.


CFPB NLP Pipeline
1. Scraper
2. EDA
3. Baseline
4. Transformer
5. Playground
5. Try the Model (Prediction Playground)
Enter a consumer complaint narrative and the fine-tuned DistilBERT model will predict the product category (e.g., Credit Reporting, Debt collection, Mortgage).
Complaint Narrative
The bank reported a false delinquency on my credit report and won't correct it
Classify Complaint
Predicted Product Category
Credit repair services
Model: distilbert
Confidence
6.16%
Confidence is the model's predicted probability for this product label.

5. Discussion

5.1 The "Context" Advantage

The primary failure mode of the TF-IDF model was its inability to understand *intent*. A sentence like *"I did not apply for this card"* contains keywords found in both "Credit Card" and "Identity Theft" (Credit Reporting) complaints. DistilBERT, utilizing self-attention mechanisms [2], effectively parsed the sentence structure to understand the specific grievance, leading to higher precision.

5.2 Computational Trade-offs

While DistilBERT provided a boost in accuracy and a significant increase in Macro-F1 score, it required significantly more computational resources. Inference time for the Transformer is also higher. However, for a government agency like the CFPB, the cost of human review for misclassified tickets likely outweighs the computational cost of running a GPU-based model.

6. Conclusion and Future Work

This project successfully demonstrated that Transformer-based architectures offer superior performance for financial complaint classification. The Fine-Tuned DistilBERT model achieved an accuracy of ~91%, significantly outperforming the classical Naive Bayes baseline.

To conclude, this project was not just about achieving high accuracy; it was about solving the 'Accuracy Paradox' inherent in financial data.

We started with a dataset where 60% of the complaints were identical. A basic model could cheat its way to a passing grade by ignoring the rare problems. But in banking, ignoring a rare problem like a 'Money Transfer' fraud is a compliance disaster.

By fine-tuning DistilBERT, we forced the model to actually read the complaints. We utilized the Attention Mechanism to look past the privacy redactions and understand the context of the customer's distress.

The result is a system that maintains 91% overall accuracy while improving our ability to detect rare categories by nearly 66.67% (moving from 36% to 60% F1). We have built a pipeline that is robust, efficient and ready to protect consumers at scale.

Future improvements could include:

1. **Hierarchical Classification:** Implementing a two-step model that first predicts the Product and then the Sub-product.
2. **Longformer Integration:** DistilBERT has a 512-token limit. Many complaints are longer; models like Longformer could capture information currently lost to truncation.
3. **Ensemble Methods:** Combining the speed of Naive Bayes for obvious cases and routing low-confidence predictions to DistilBERT to optimize costs.

7. References

[1] Consumer Financial Protection Bureau (CFPB). (2024). *Consumer Complaint Database*. Available at: https://www.consumerfinance.gov/data-research/consumer-complaints/

[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv preprint arXiv:1810.04805.

[3] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv preprint arXiv:1910.01108.

Code Repositories & Libraries Used:

- Hugging Face Transformers: https://github.com/huggingface/transformers
- Scikit-Learn: https://github.com/scikit-learn/scikit-learn
- Pandas: https://github.com/pandas-dev/pandas