





The Team:



Benjer Shoval



Berkovich Alon



Amar Adir

Customer Service Sucks!

Endless Ping-Ponging

72% of customers say that explaining their problems to multiple people is **poor customer service**

33% of customers are most frustrated by having to repeat themselves to multiple support reps



Annoying Chatbots

90% of people prefer interacting with a human for customer service over a chatbot

69% of customers would use a chatbot if they knew it could resolve the issue more quickly



Inefficient Processes

The average customer service response time is 12 hours and 10 minutes

Companies using AI report a **37% drop in first response times** compared to those without automation

AI adoption leads to a **35% cost reduction in** customer service operations and a **32% revenue increase**

For every \$1 investment in AI, businesses see an average return of \$3.5



Direct Competitors



Zendesk

Enterprise leader with
OpenAI integration.
\$55/month.



Intercom

**SaaS platform with NLP
focus.** From **\$39/month.**



Ada

Self-learning bots with
brand customization.
Enterprise pricing.



Tidio

**SMB solution with high
automation.** From
\$29/month.

Competitive Analysis: Strengths, Weaknesses, Opportunities, and Threats (SWOT)



Strengths

Mixed customer satisfaction for Ada

Tidio offers limited integration capabilities

Lack of targeted solutions for niche industries



Weaknesses

AI-powered chatbots and automation are standard across all competitors

Zendesk and Ada are market leaders with strong enterprise-level scalability



Opportunities

Increasing demand for AI-driven customer service

Potential in **niche markets** and specialized industries

Room for CRM expansion and third-party integrations

Underserved SMB markets



Threats

Intense competition from direct and indirect competitors

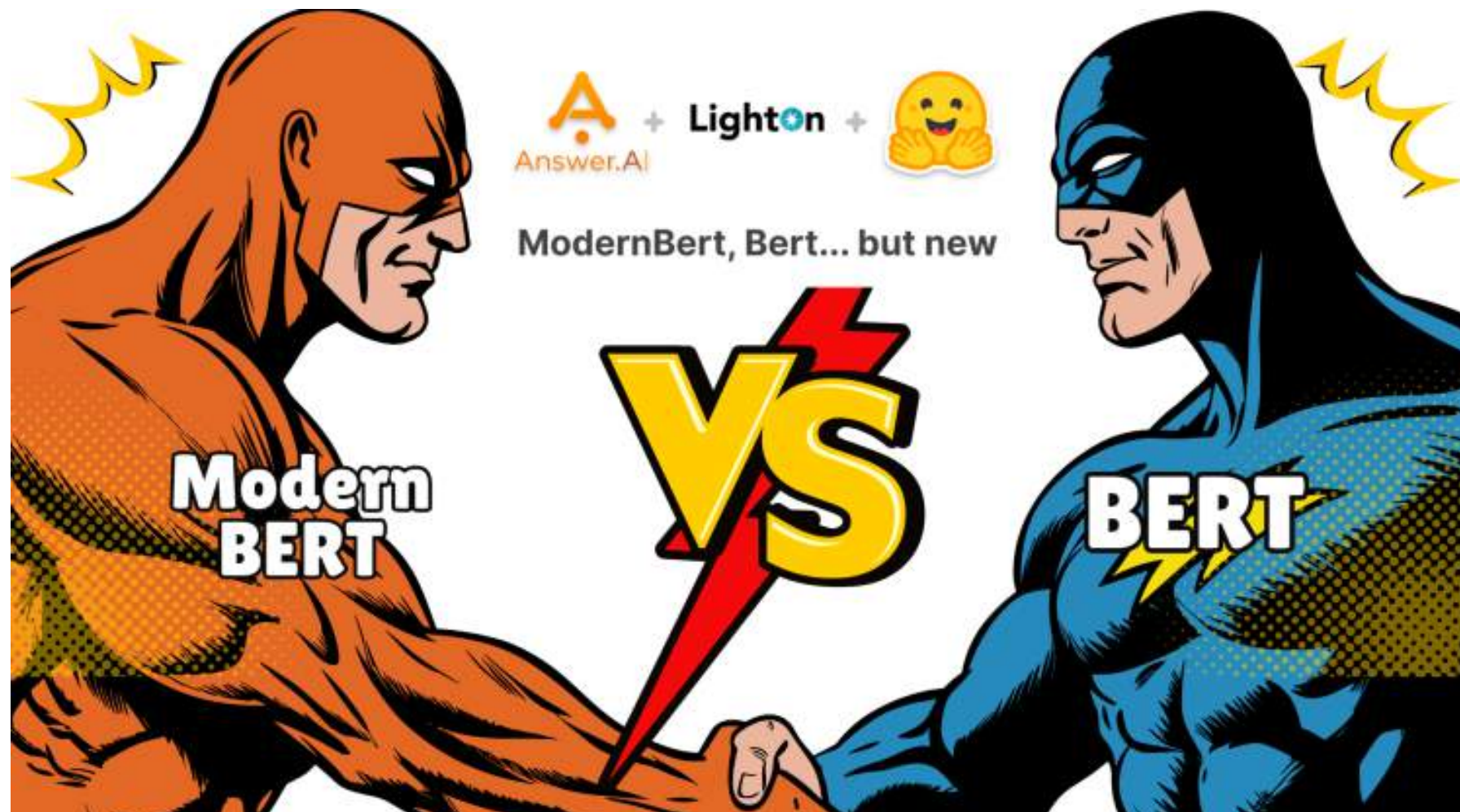
Rapid technological advancements risk obsolescence

2018

Google



Bidirectional
Encoder
Representations from
Transformers



- 🚀 8K Tokens: ModernBERT handles 8,192 tokens, ideal for long texts like summarization.
- ⚡ 4x Faster: Boosts speed, uses 80% less memory, optimized for GPUs.
- 📚 Diverse Data: Trained on 2T tokens, excels in code and scientific tasks.
- 🧠 Smart Design: Features ROPE, GeGLU, and Flash Attention for precision.
- 🌟 Easy Switch: Replace BERT with no code changes for enhanced NLP.

Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference

Benjamin Warner^{1†} Antoine Chaffin^{2†} Benjamin Clavié^{1†}
Orion Weller³ Oskar Hallström² Said Taghadouini²
Alexis Gallagher¹ Raja Biswas¹ Faisal Ladhak^{4*} Tom Aarsen⁵
Nathan Cooper¹ Griffin Adams¹ Jeremy Howard¹ Iacopo Poli²

¹Answer.AI ²LightOn ³Johns Hopkins University ⁴NVIDIA ⁵HuggingFace

†: core authors, *: work done while at Answer.AI

Correspondence: {bw,bc}@answer.ai, antoine.chaffin@lighton.ai

Abstract

Encoder-only transformer models such as BERT offer a great performance-size tradeoff for retrieval and classification tasks with respect to larger decoder-only models. Despite being the workhorse of numerous production pipelines, there have been limited Pareto improvements to BERT since its release. In this paper, we introduce ModernBERT, bringing modern model optimizations to encoder-only models and representing a major Pareto improvement over older encoders. Trained on 2 trillion tokens with a native 8192 sequence length, ModernBERT models exhibit state-of-the-art results on a large pool of evaluations encompassing diverse classification tasks and both single and multi-vector retrieval on different domains (including code). In addition to strong downstream performance, ModernBERT is also the most speed and memory efficient encoder and is designed for inference on common GPUs.

1 Introduction

After the release of BERT (Devlin et al., 2019), encoder-only transformer-based (Vaswani et al., 2017) language models dominated most applications of modern Natural Language Processing

option against encoder-decoder and decoder-only language models when dealing with substantial amounts of data (Penedo et al., 2024).

Encoder models are particularly popular in Information Retrieval (IR) applications, e.g., semantic search, with notable progress on leveraging encoders for this task (Karpukhin et al., 2020; Khat-tab and Zaharia, 2020). While LLMs have taken the spotlight in recent years, they have also motivated a renewed interest in encoder-only models for IR. Indeed, encoder-based semantic search is a core component of Retrieval-Augmented Generation (RAG) pipelines (Lewis et al., 2020), where encoder models are used to retrieve and feed LLMs with context relevant to user queries.

Encoder-only models are also still frequently used for a variety of discriminative tasks such as classification (Tunstall et al., 2022) or Natural Entity Recognition (NER) (Zaratiana et al., 2024), where they often match the performance of specialized LLMs. Here again, they can be used in conjunction with LLMs, for example detecting toxic prompts (Ji et al., 2023; Jiang et al., 2024b) and preventing responses, or routing queries in an agentic framework (Yao et al., 2023; Schick et al., 2023).

Surprisingly, these pipelines currently rely on older models, and quite often on the original BERT



Phase 1: MVP Development Overview

Overview of Key Objectives and Features



Project Duration

The MVP development phase is planned for a span of 0 to 3 months, allowing for a rapid iteration cycle that facilitates quick feedback and adjustments.



Goal of the Phase

The primary objective is to establish a Basic Complaint Handling System that serves as the foundation for user interactions and issue resolution.



AI Categorization Feature

Implementing AI-driven categorization will enhance the efficiency of the complaint handling system by automatically sorting complaints into predefined categories, thereby streamlining the resolution process.



API Integration

Integrating APIs is crucial for connecting the complaint handling system with external services and databases, enabling seamless data exchange and enhancing functionality.



User Experience Focus

The MVP aims to prioritize user experience, ensuring that the interface is intuitive and easy to navigate, which is essential for user satisfaction and engagement.



Feedback Mechanism

Establishing a feedback mechanism during this phase will allow users to provide insights on system performance, which is vital for future improvements and iterations.



Testing and Validation

Rigorous testing and validation will be conducted to ensure that the Basic Complaint Handling System functions correctly and meets the established requirements before launch.



Stakeholder Involvement

Engaging stakeholders throughout the MVP development process is critical for aligning the project with business objectives and user needs, ensuring the system's relevance.

Step 1: Set Up Supabase Backend

A Comprehensive Guide to Initial Setup



Create a Supabase Project

Begin by creating a new project on the Supabase platform. This will serve as the foundation for your backend infrastructure, enabling you to utilize its powerful features.

Enable Authentication

Set up authentication options such as email sign-ups and social logins to secure user access. This is crucial for managing user accounts and ensuring data privacy.

Set Up Database Schema

Design and implement a database schema that includes tables for Users, Complaints, Categories, and Complaint History. This structure will help organize data

Implement Row Level Security (RLS) Policies

Establish Row Level Security policies to protect sensitive data. RLS helps control access to rows in a database table based on user roles, enhancing overall security.

Step 2: Implement API for Complaint Management

Integrating advanced features for efficient complaint handling

01

Utilize Supabase's Auto-Generated REST API

Supabase provides an efficient way to handle CRUD operations through its automatically generated REST API. This allows developers to quickly implement functionality for creating, reading, updating, and deleting complaints without having to build the backend from scratch.

02

Enable Realtime Subscriptions

By enabling Realtime Subscriptions in Supabase, teams can track and respond to complaint updates instantly. This feature enhances user experience by providing immediate feedback and updates, making the complaint management process more dynamic and responsive.

Step 3: AI-Powered Categorization

Leveraging AI Techniques for Effective Complaint Management

01 Utilization of ModernBERT Embeddings

ModernBERT embeddings are advanced representations of text that capture semantic meaning. They enable the system to understand the context of complaints, ensuring accurate classification.

02 Application of HDBSCAN Clustering

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is employed to auto-classify complaints based on the density of data points. This technique allows for discovering clusters of similar complaints without prior knowledge of the number of clusters.

03 Storage of Embeddings in Supabase Vector

Storing embeddings in Supabase Vector facilitates efficient similarity searches. This structure allows for rapid retrieval of similar complaints, enhancing the ability to address issues effectively.

04 Automated Complaint Classification

The integration of ModernBERT and HDBSCAN results in an automated system that classifies complaints in real-time. This reduces manual effort and speeds up response times.

05 Improved Customer Insights

By categorizing complaints accurately, organizations can gain better insights into customer issues, leading to informed decision-making and improved service quality.

Step 4: Integrate Llama 3.2 for Friendly Responses

Enhancing User Interaction with AI

01 Deploy Llama 3.2 (3B version)

Initiate the deployment of Llama 3.2 using the 3B model on a lightweight Llama.cpp setup. This step ensures the system is agile and ready for quick processing, which is essential for generating timely responses.

02 Generate Clarifying Questions

Utilize Llama 3.2 to create clarifying questions that address user complaints effectively. This functionality enhances interaction quality by ensuring that responses are relevant and tailored to specific user needs.

03 Crafting Responses for Complaints

Develop structured responses for various complaints using Llama 3.2 to ensure that users feel understood and valued. This process maximizes user satisfaction and promotes positive engagement.

04 Implement LangChain for Prompt Handling

Leverage LangChain to manage structured prompts effectively, which streamlines the interaction process. This integration allows for a more organized approach to handling user inquiries.

Step 5: Implement Basic Resolution Engine

Enhancing Customer Support through Technology

Train a GNN model to match complaints to resolutions.

Developing a Graph Neural Network (GNN) model is crucial for efficiently linking customer complaints to appropriate resolutions. This involves training the model on a dataset of past complaints and their corresponding resolutions, allowing it to learn patterns and make accurate predictions for new cases. Such automation can significantly reduce response times and improve customer satisfaction.

Implement rule-based fallback system for known complaints.

In cases where the GNN model encounters complaints that it cannot resolve, a rule-based fallback system ensures that known complaints are handled correctly. This system leverages predefined rules and historical data to provide quick solutions, minimizing frustration for users and maintaining service quality.

Step 6: Frontend Integration with Lovable.dev

Integrating Supabase for Enhanced User Experience

01 Integrate Supabase Auth

Leverage Supabase's authentication capabilities to securely manage user access and identities within the Lovable.dev platform. This integration ensures that users can seamlessly sign in and maintain their session while using the application.

02 Connect to Supabase Database

Establish a connection to Supabase's database to enable storage and retrieval of user data. This integration empowers Lovable.dev to efficiently manage user complaints and track their statuses over

03 Utilize Real-Time API

Implement Supabase's real-time API to facilitate instant updates and notifications regarding complaint submissions and statuses. This feature enhances user engagement by providing timely information.

04 Design Basic UI for Complaint Submission

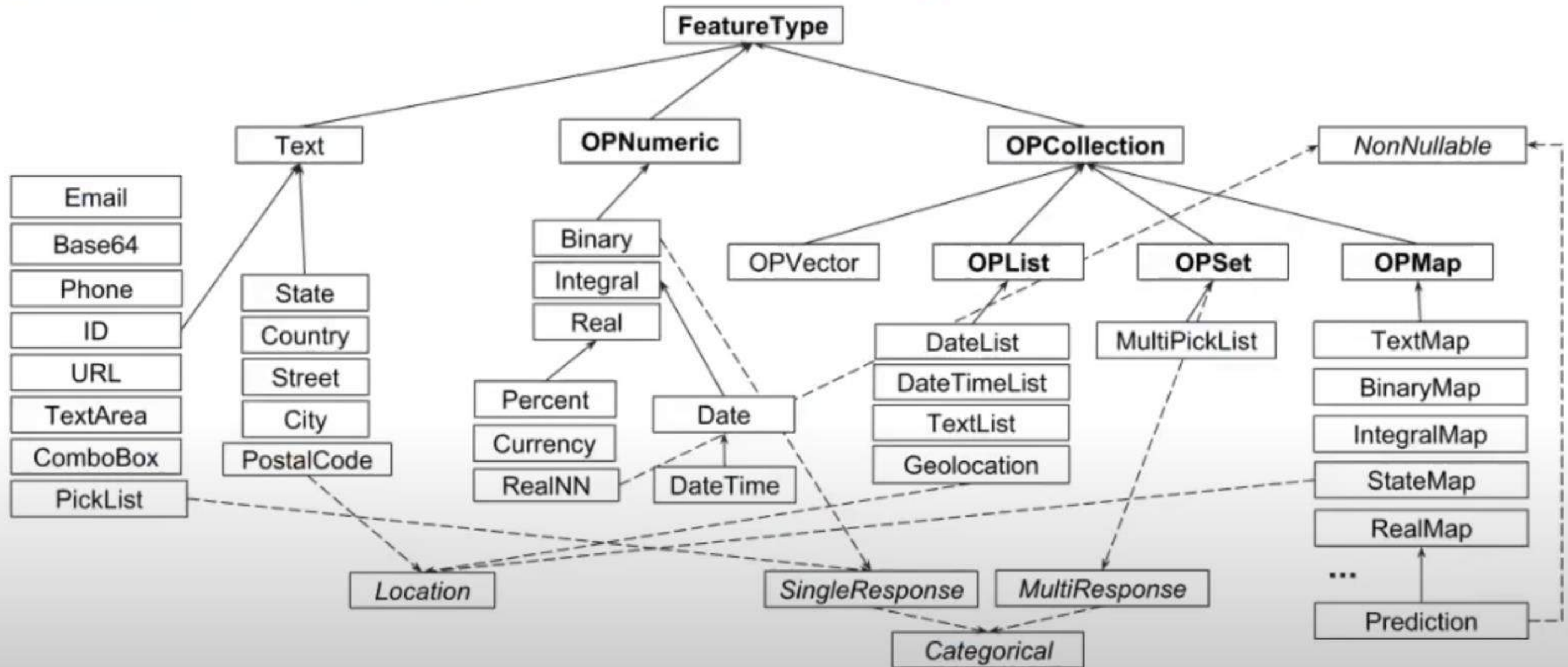
Create a user-friendly interface that allows users to submit complaints easily. The design should prioritize simplicity and accessibility to ensure a smooth user experience.

05 Implement Complaint Tracking Feature

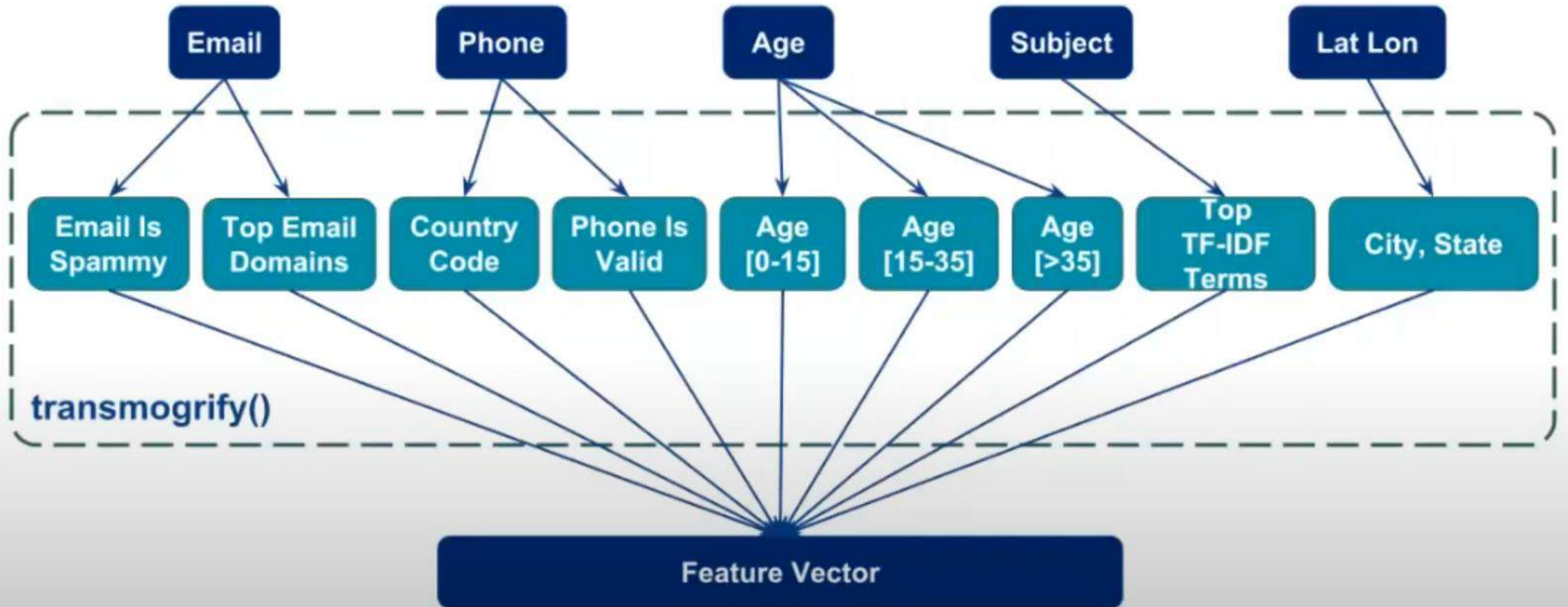
Incorporate a tracking system within the UI that allows users to monitor the status of their complaints. This feature promotes transparency and keeps users informed about the progress of their submissions.

Using TransmogrifAI

Type Hierarchy For Machine Learning

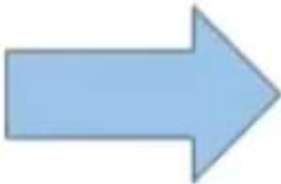


Automatic Feature Engineering



Numeric – Imputation and Null value tracking

Feature
34,200.03
14.001.02
22,430.11
47,895.66



Feature	Null Indicator
34,200.03	0
14.001.02	0
16,045.21	1
22,430.11	0
16,045.21	1
47,895.66	0

Categorical: One Hot Encoding

state									
NY	0	...	0	...	1	...	0	...	0
WA	0	...	0	...	0	...	1	...	0
CA	0	...	1	...	0	...	0	...	0

Automatic Feature Engineering

Numeric

Imputation
Track null value
Scaling - zNormalize,
log, linear
Smart Binning

Categorical

Imputation
Track null value
One Hot Encoding
Dynamic Top K pivot

Text

Language Detection
Language-wise
Tokenization
Hash Encoding
Tf-Idf
Word2Vec
Name Entity
Resolution
Smart Categorical

Temporal


Time difference
Circular Statistics
Time extraction (day,
week, month, year)

Spatial

Reverse Geocoding
Nearest POI

Main Data Source :

Consumer Financial Protection Bureau Dataset

 An official website of the United States government

[Español](#) [中文](#) [Tiếng Việt](#) [한국어](#) [Tagalog](#) [Русский](#) [العربية](#) [Kreyòl Ayisyen](#) (855) 411-2372



Consumer Education ▾

On your side through
life's financial moments.

We're the Consumer Financial Protection Bureau, a U.S. government agency dedicated to making sure you are treated fairly by banks, lenders and other financial institutions.



Dataset Contains **7.6M Data Rows**
We will Train on **2.4M Rows** After Cleaning

[Different Complaint Categories of Dataset for more info](#)

There are several accounts falsely reported to my credit report. I complained to the FTC, and directly to the credit bureaus. I submitted copies of my DL, SS Card, Current Bills, ETC all to prove my identity. I asked them to match												
	L	K	J	I	H	G	F	E	D	C	B	A
	Sub-product	Product	Date received	Complaint ID	Consumer disputed?	Complaint Status	Channel of Complaint	Complaint Source Location	Consumer complaint narrative	Sub-issue	Issue	missing data by nlp model
	Credit rep	Credit rep	03/23/20	3189109		Closed with	Web	IL	The Summer of XX/XX/20	Account in	Incorrect information	
	Credit rep	Credit rep	03/22/20	3187982		Closed with	Web	VA	There are many mistakes	Account in	Incorrect information	
	Credit rep	Credit rep	03/22/20	3187954		Closed with	Web	TX	There are many mistakes	Account in	Incorrect information	
	Credit rep	Credit rep	03/22/20	3188091		Closed with	Web	TX	There are many mistakes	Account in	Incorrect information	



Our Dataset: Unveiling the Details



Content

Consumer complaint narratives and optional fields like date, channel, status, and user metadata.



Structure

CSV or JSON format with columns such as complaint ID, narrative, date/time, and location.



Source

Public dataset - **CFPB consumer complaints**

Unstructured Text in the Wild

The Challenge

Handling large volumes of unstructured consumer complaints to identify trends and potential high-risk issues.

The Data columns – text based

Field name	Description
Date received	The date the CFPB received the complaint. For example, "05/25/2013."
Product	The type of product the consumer identified in the complaint. For example, "Checking or savings account" or "Student loan."
Sub-product	The type of sub-product the consumer identified in the complaint. For example, "Checking account" or "Private student loan."
Issue	The issue the consumer identified in the complaint. For example, "Managing an account" or "Struggling to repay your loan."
Sub-issue	The sub-issue the consumer identified in the complaint. For example, "Deposits and withdrawals" or "Problem lowering your monthly payments."



Consumer complaint narrative

Consumer complaint narrative is the consumer-submitted description of "what happened" from the complaint. Consumers must opt-in to share their narrative. We will not publish the narrative unless the consumer consents, and consumers can opt-out at any time. The CFPB takes reasonable steps to scrub personal information from each complaint that could be used to identify the consumer.

Features Requirements

Key Functional Requirements

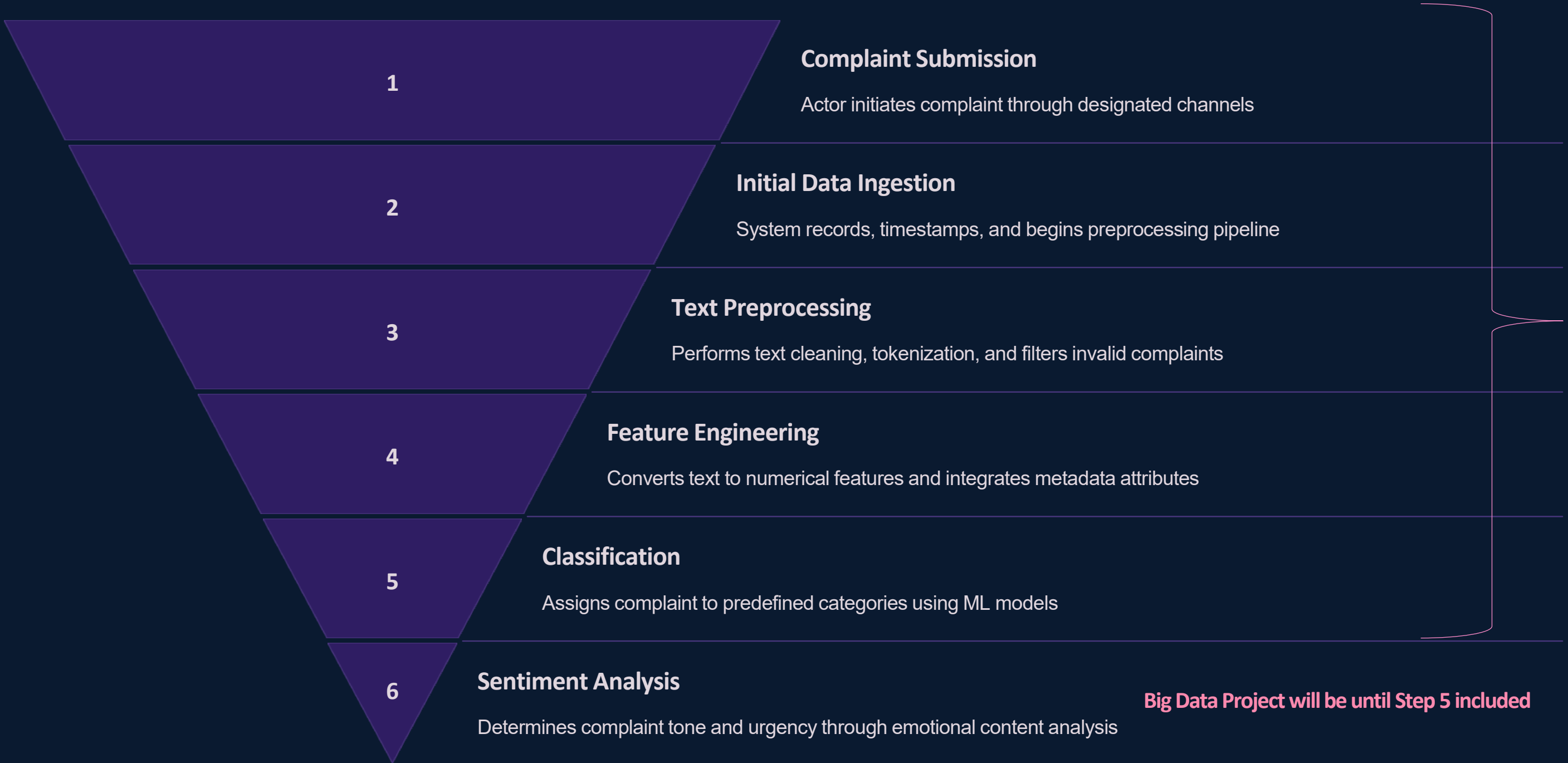
- AI-powered classification and resolution actions
- Real-time tracking and multi-channel notifications
- Analytics for proactive issue identification

Non-Functional Requirements

- Process 90% of complaints within 2 seconds
- Support 10,000 daily complaints

Full Features list if needed

Complaint Processing Flowchart



Big Data Project will be until Step 5 included

for Pseudocode press here

Notebook
Pseudocode

Technology Stack



Frontend

Next.js, TypeScript, Tailwind CSS for performance and maintainability.



Backend

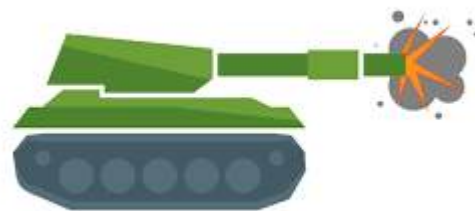
Ingestion & Streaming: Primarily via **Kafka** for real-time data flow.



NLP Integration

Spark for large-scale text processing (tokenization, n-grams, clustering).

deepseek-ai/
DeepSeek-R1



שאלות?

 SoloSolveAI



We Are  SoloSolveAI