FinEase

About Architecture Dashboard Report

ML-Powered Platform

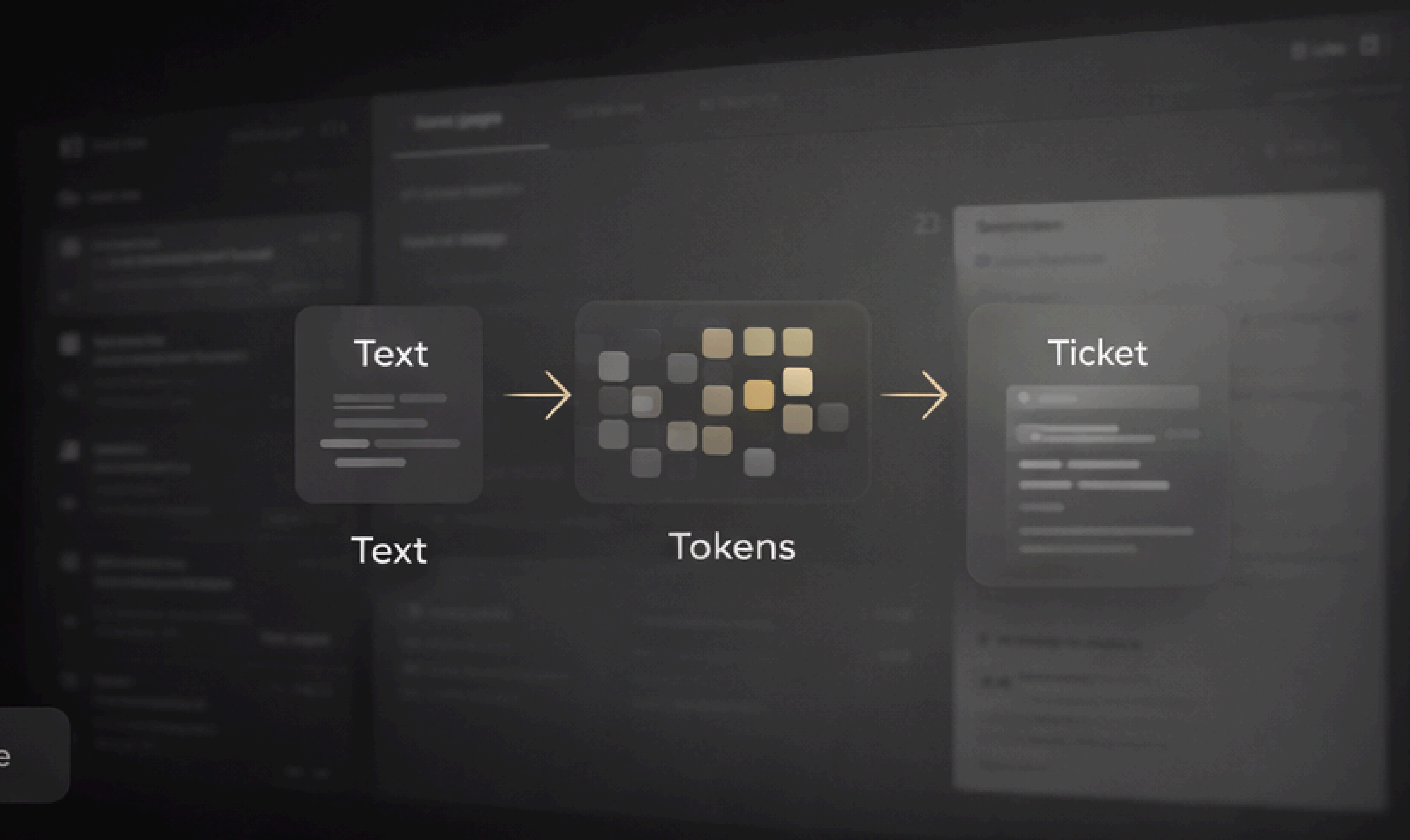
FinEase

Complaint Intelligence for Digital Banking

An end-to-end system that transforms unstructured customer complaints into structured, actionable support tickets using NLP.

View Technical Report

Explore System Architecture



Final Technical & Product Report

Project

FinEase — ML-Powered Complaint Intelligence Platform

Year

2025-26

Author

Shashwat Chauhan

Institution BITS Pilani, Goa Campus

Year 2025-26

Final Technical & Product Report

FinEase

COMPLAINTS PORTAL Architecture

Received	252
Replied	28,945
Drafts	14
Starred	3

Received

16,000 complaints, 252 unread

Filter

Complaints	Department/ Topic
I can not get from chase who services my mortgages, who owns it and who has original loan docs	Mortgages/Loan
The bill amount of my credit card was debited twice. Please look into the matter and resolve at the earliest.	Credit Card or Prepaid Card
I want to open a salary acount at your downtown branch. Please provide me the procedure.	Bank Account Services
Yesterday, I received a fraudulent email regarding renewal of my services.	Bank Account Services
What is the procedure to know my CIBIL score?	Fraud/Theft/Dispute
I can not get from chase who services my mortgages, who owns it ahw has original loan docs	Mortgages/Loan
The bill amount of my credit card was debited twice. Please look into the matter resolve at the earliest.	Credit Card or Prepaid Card
I want to open a salary account at your downtown branch. Please provide me the procedure.	Bank Account Services
Yesterday, I received a fraudulent email regarding renewal of my services.	Bank Account Services
	Fraud/Theft/Dispute

FinEase

ABOUT ARCHITECTURE DASHBOARD FinEase — ML-Powered Complaint Intelligence Platform

Get In Touch

Please fill out the form and we nil be happy to help you

NAME

EMAIL

QUERY

SEND

OR

MORE QUERIES TO YOUR EXPERIENCE

Security & Privacy >

System Architecture >

NLP & Models >

Dashboard Operations >

RECENT QUERIES

There are no recent queries raised in Last 30 days

Show Older Queries

FinEase — ML-Powered Complaint Intelligence Platform

Final Product & Technical Report

Author: Shashwat Chauhan
Institution: BITS Pilani, Goa Campus
Year: 2025–26

Executive Summary

FinEase is a production-oriented **complaint intelligence and routing platform** built for digital banking operations. Its purpose is not to achieve the highest possible classification score in isolation, but to **help banks make better, faster routing decisions when complaint data is noisy, ambiguous, and imperfect**.

In real banking environments, complaints arrive as long, emotional, and loosely structured narratives. Treating these inputs as clean, well-labeled data leads to fragile systems that break in production. FinEase is designed around a different assumption: ambiguity is inevitable and must be managed, not ignored.

The system follows one core principle:

FinEase applies this principle through a structured NLP pipeline that combines unsupervised learning, human validation, and supervised classification, all surfaced through a transparent product interface.

---

1. Product Context & Problem Definition

Banks receive a continuous stream of customer complaints written in free-form natural language. These complaints contain valuable signals about product failures, service gaps, and systemic issues, but they are difficult to operationalize at scale.

From a product perspective, three challenges consistently appear:

- Complaint narratives vary widely in length, tone, and clarity
- Product intent is often implied rather than explicitly stated
- Historical complaint labels are inconsistent and unreliable

As a result, complaint routing is often manual, slow, and error-prone. Misrouted tickets increase resolution time, operational cost, and customer frustration.

Problem framing: The primary goal is not to maximize classification accuracy, but to **reduce uncertainty early enough to enable dependable downstream routing and resolution decisions**.

---

2. Design Philosophy

FinEase is intentionally designed as a **decision-support system**, not a fully autonomous classifier. The platform assists human operations teams by reducing ambiguity and organizing information before final action is taken.

Four principles guide all system decisions:

- Structure is introduced gradually, not all at once
- Every stage remains interpretable and auditable
- Dependence on perfectly labeled data is minimized
- The system aligns with real banking workflows and compliance constraints

These principles directly influence both the modeling strategy and the product experience.

---

3. Canonical System Pipeline

FinEase follows a staged pipeline that introduces complexity only when the previous step has reduced ambiguity. This prevents overfitting early and keeps the system stable under real-world noise.

```
Raw Complaint Text
→ Conservative Text Normalization
→ Linguistic Simplification
→ Sparse, Interpretable Vectorization
→ Topic Discovery (Unsupervised)
→ Human Topic Validation
→ Pseudo-Label Generation
→ Supervised Classification
→ Confidence-Bound Routing
→ Dashboard & Ticket System
```

This pipeline serves as the single source of truth for both backend implementation and product behavior.

---

4. Text Normalization

Objective

Reduce syntactic noise without altering semantic intent.

Rationale

Raw complaint text contains casing inconsistencies, punctuation artifacts, masked identifiers, and numeric tokens that inflate vocabulary size without adding meaning.

Implementation

```
def clean_text(text):
        text = text.lower()
        text = re.sub(r'\[.*\]', '', text)
        text = text.translate(str.maketrans('', '', string.punctuation))
        text = re.sub(r'\S*\d\S*', '', text)
        return text.strip()
```

Outcome

- Reduced vocabulary explosion
- Improved feature stability over time
- Consistent downstream representations

---

5. Linguistic Simplification

Objective

Concentrate semantic signal relevant to routing.

Approach

- Lemmatization to normalize inflected forms
- Selectiveretention of content-bearing tokens (notably nouns)

Implementation

```
def lemmatize(text):
        doc = nlp(text)
        return ' '.join(t.lemma_ for t in doc if not t.is_stop)


def extract_nouns(text):
        doc = nlp(text)
        return ' '.join(t.text for t in doc if t.tag_ == 'NN')
```

Outcome

- Reduced linguistic variance
- Improved topic coherence in unsupervised modeling

This stage deliberately avoids deep contextual embeddings to preserve transparency and auditability.

---

6. Feature Representation

Objective
Transform text into numeric features without sacrificing interpretability.

Choice
TF-IDF was selected over dense embeddings.
Rationale

- Transparent term weighting
- Stable under data drift
- Easyto audit and explain in regulated environments

Implementation

```
tfidf = TfidfVectorizer(
        min_df=2,
        max_df=0.95,
        stop_words='english'
)
X = tfidf.fit_transform(text_processed)
```

---

7. Topic Discovery (Unsupervised Learning)

Objective
Discover latent complaint structure without relying on unreliable labels.

Model
Non-Negative Matrix Factorization (NMF)

Implementation

```
nmf = NMF(n_components=5, random_state=40)
W = nmf.fit_transform(X)
H = nmf.components_
```

Human-Validated Topics

Topic Index	Operational Category
0	Bank Account Services
1	Credit Cards
2	Payments & Billing
3	Fraud & Disputes
4	Mortgages & Loans

Key Insight

NMF components are **intermediate structure**, not final predictions. Human validation converts mathematical patterns into trusted operational categories.

---

8. Pseudo-Label Generation

Once topics are validated, dominant topic assignments are used to generate pseudo-labels:

```
labels = W.argmax(axis=1)
```

These labels are: - Not ground truth - Operationally consistent - Sufficient for supervised routing models

This approach dramatically reduces manual annotation cost while maintaining reliability.

---

9. Supervised Classification

Objective

Enable real-time complaint routing at scale.

Data Preparation

```
X = training_data.complaint_text
y = training_data.Topic

count_vect = CountVectorizer()
X_vect = count_vect.fit_transform(X)

tfidf_transformer = TfidfTransformer()
X_tfidf = tfidf_transformer.fit_transform(X_vect)
```

Models Evaluated

- Multinomial Naive Bayes
- Logistic Regression
- Decision Tree
- Random Forest

Selection Criteria

- Precision stability across categories
- Predictable failure modes
- Low variance across data slices

Logistic Regression and Random Forest demonstrated the most consistent operational behavior.

---

10. Evaluation Strategy

Evaluation prioritizes **robustness over peak metrics**:

- Train–test splits
- Stratified K-fold cross-validation
- GridSearchCV for controlled tuning

No aggressive optimization was pursued to avoid overfitting noisy complaint distributions.

---

11. Inference & Deployment Flow

At inference time:

1. Complaint text is normalized
2. Features are extracted
3. Category is predicted
4. Ticket is routed automatically

The pipeline is deterministic, explainable, and production-safe.

---

12. Product Interface Integration

The UI is a direct manifestation of pipeline stages:

- **Complaints Inbox:** post-classification routing output
- **Department / Topic Column:** supervised predictions
- **Architecture View:** full pipeline transparency
- **Technical Report Access:** auditability and trust
- **Contact / Queries:** human escalation path

The interface reinforces confidence in system decisions rather than obscuring them.

---

13. Financial Modeling Case Study (Supporting Analogy)

A parallel analysis compares OLS regression with regularized methods in financial factor modeling.
Insight

Models that ignore noise, correlation, and regime shifts fail under real-world conditions.

```
betas = np.linalg.inv(X.T @ X) @ X.T @ y
```

Regularization stabilizes estimates:

```
lasso = Lasso(alpha=best_lambda)
lasso.fit(X, y)
```

This mirrors FinEase's philosophy: **constraint produces robustness**.

---

14. Product Impact

Operational Impact

- Reduced manual triaging
- Faster first-touch resolution
- Consistent routing behavior

Strategic Impact

- Visibility into systemic complaint patterns
- Foundation for scalable monitoring and analytics

---

15. Future Extensions

- Transformer-based embeddings with confidence gating
- Explicit uncertainty scoring for escalation
- Multi-label complaint handling
- CRM and ticketing system integration

---

16. Conclusion

FinEase demonstrates how disciplined pipeline design, interpretability, and human-in-the-loop validation enable machine learning systems to operate reliably in noisy, regulated environments.

The system prioritizes **decision quality over metric optimization**, making it suitable for real-world banking operations rather than controlled laboratory benchmarks.