

Smart Support Ticket Prioritization System

Your Name
ID: Your ID Number

July 15, 2025

1 Introduction

Customer support is vital across industries. As businesses grow, the volume of customer queries—via email, web forms, or chat—increases, making it difficult for support teams to identify and respond to urgent issues quickly. Traditional systems process tickets by arrival order, which can delay responses to critical problems and harm customer satisfaction and retention [4, 5].

This project proposes a **Smart Support Ticket Prioritization System** that uses Natural Language Processing (NLP) and Machine Learning (ML) to analyze incoming support tickets, score their urgency based on message content and customer value, and present prioritized tickets in a dashboard. This helps agents address the most important issues first, improving efficiency and customer experience [8].

Potential Benefits:

- *Academic*: Supports research in NLP/ML for customer service.
- *Technical*: Delivers a prototype combining NLP, ML, and web technologies for real-time ticket triage.
- *Business*: Improves satisfaction and reduces churn by prioritizing urgent issues.
- *Social*: Enhances experience for users with urgent needs and supports fairer service delivery.

2 Problem Statement

Companies receive many support requests daily. Manual triage is slow and error-prone, risking missed or delayed urgent cases. Delayed responses can cause lost productivity, dissatisfaction, and reputational harm. Existing systems often rely on simple rules or manual sorting, lacking analysis of message context or customer value [5].

Why this research is needed: There is a need for systems that use NLP and customer data to prioritize tickets in real time. Addressing this can improve operational efficiency and customer outcomes as expectations for rapid, personalized support rise [4].

3 Research Questions

Main Question: How can NLP and ML be used to prioritize customer support tickets by urgency in a scalable, industry-independent way?

Supporting Questions:

- Which ticket text and customer properties best predict urgency?
- How well do different ML models predict ticket urgency?
- What are the challenges in integrating such a system into existing support workflows?
- How can the system be adapted to different industries and ticket types?

These questions are feasible, interesting, novel, ethical, and relevant.

4 Literature Review

Automated ticket classification is a growing research and industry area. NLP can detect intent and urgency in text, improving triage accuracy [4]. Early methods used keyword searches but lacked context sensitivity [5]. Recent work uses ML to classify tickets based on features from ticket text and customer data [8].

Integrating customer value (e.g., spending history) is important, as not all tickets have equal business impact [8]. Public datasets, such as those on Kaggle, provide real-world data for training and evaluating ML models [9]. Despite progress, few solutions combine NLP, customer metrics, and real-time dashboards for support teams.

Advances in NLP, such as transformer models and multilingual embeddings, allow analysis across languages, critical for global companies. Simpler models like logistic regression and decision trees remain common due to interpretability and efficiency. The choice of model depends on the trade-off between accuracy, speed, and deployment ease.

Urgency scoring is supported by research showing customers expect faster responses to critical issues, and prioritizing high-value or high-impact tickets reduces churn. Risks of bias exist if the system over-prioritizes certain customers or ticket types, which must be managed through careful design and evaluation [11].

5 Methods

5.1 System Overview

- **Frontend:** Web dashboard (React) for agents, displaying prioritized tickets, urgency scores, filters, and action buttons.
- **Backend:** RESTful API (Flask) with an SQL Database to store tickets, customer profiles, and urgency scores.
- **ML Model:** Classifier trained on real-world datasets, using ticket text and customer metrics to predict urgency (score 0–1).

5.2 Data Sources

Two public datasets will be evaluated:

1. **Multilingual Customer Support Tickets:** Ticket text, metadata, and CRM data—suitable for NLP analysis.
2. **Customer Support Ticket Dataset:** Ticket descriptions, categories, urgency labels, and agent replies—ideal for supervised ML.

Datasets will be assessed for quality and completeness. Data augmentation (e.g., synthetic ticket generation or translation) may be used if needed [9].

5.3 Data Processing

- **Preprocessing:** Clean text, remove noise, handle missing values, normalize data.
- **Feature Engineering:** Combine text features (keywords, sentiment, urgency indicators) with customer metrics (spending history, previous tickets).
- **Model Training:** Compare simple models (logistic regression, decision trees) with more complex ones (random forests). Tune hyperparameters for best performance.
- **Evaluation:** Use accuracy, precision, recall, F1-score, and confusion matrix. Apply cross-validation for robustness.
- **Integration:** Urgency scores and suggested actions will be displayed in the dashboard.

5.4 Tools and Technologies

- **Programming:** Python (backend/ML), JavaScript (frontend).
- **Libraries:** scikit-learn, pandas, NLTK/spaCy for NLP; React for UI.

5.5 Rationale

The modular design allows adaptation to different industries and datasets. Open-source tools and public datasets make the project feasible and cost-effective.

6 Ethical Implications

Automated prioritization raises several ethical issues:

- **Fairness:** The system must not disadvantage users based on language, writing style, or customer value. Fairness checks and bias mitigation will be included in model training [12].
- **Transparency:** Agents and customers will see explicit urgency scores and thresholds.

- **Data Privacy:** Customer data will be handled per GDPR and company guidelines. Sensitive data will be anonymized where possible [7].
- **Accountability:** Human agents retain final responsibility; the system is a decision-support tool.

Best practices from leading AI ethics literature will be followed [7].

7 Requirements and Feasibility

7.1 Functional Requirements

- Public ticket submission form (name, email, phone, issue description)
- Agent dashboard with prioritized tickets, urgency scores, suggested actions
- Admin features: ticket search, filters, export
- Integration-ready API

7.2 Non-Functional Requirements

- **Performance:** Real-time scoring (<2 seconds per ticket)
- **Scalability:** Support for hundreds of tickets per day
- **Security:** Secure data storage and access controls
- **Usability:** Simple, intuitive UI for agents
- **Reliability:** System uptime of 99.5% or higher

7.3 User Requirements

- Easy ticket submission for customers, with confirmation
- Actionable dashboard for agents, with urgent ticket indicators
- Admins can review, filter, and export ticket data

7.4 Feasibility

- **Technical:** All technologies are open source and well-documented. Datasets are available.
- **Time:** Scope is manageable within the dissertation period, with a clear plan.
- **Risks:**

- *Data quality*: Mitigated by careful selection, preprocessing, augmentation
- *Model performance*: Fallback to rule-based/manual review if needed
- *Integration*: Modular APIs for phased deployment
- *User adoption*: Addressed through user testing and training

7.5 Risk Management

Risk	Likelihood	Impact	Mitigation
Poor data quality	Medium	High	Use multiple datasets, clean data
Model bias	Medium	High	Fairness checks, retraining
Integration challenges	Low	Medium	Modular APIs, clear documentation
Time overruns	Medium	High	Weekly milestones, reviews
User resistance	Medium	Medium	Training, clear communication

8 Project Plan

Phase	Tasks	Duration	Deliverables
Literature Review	Domain research, review systems	2 weeks	Literature review doc
Data Preparation	Download, clean, analyze datasets	2 weeks	Cleaned datasets
Model Development	Feature engineering, training, evaluation	3 weeks	Trained ML models
Backend Dev	Build API, integrate ML, DB setup	2 weeks	Working API
Frontend Dev	Develop dashboard and ticket form	2 weeks	Web dashboard
Integration	Connect frontend/backend, testing	2 weeks	Integrated system
Evaluation	Performance, usability, fairness tests	2 weeks	Evaluation report
Write-up	Document methodology, results, conclusions	2 weeks	Dissertation draft
Finalization	Revisions, formatting, submission	1 week	Final dissertation

9 Domain Context and Impact

Customer support is a key differentiator for businesses. Rapid, effective support helps retain customers and build reputation. Digital channels have increased both the volume and

complexity of requests, making manual triage unsustainable. NLP and ML can automate prioritization, ensuring urgent issues are addressed promptly and resources used efficiently [4, 8].

The system is relevant to technology, retail, finance, healthcare, and more. Its modular design allows adaptation to various ticket types, languages, and customer profiles.

References

- [1] Wilkinson, A. M. (1991). *The scientist’s handbook for writing papers and dissertations*.
- [2] Wiersma, W. (1995). *Research methods in education* (6th ed.).
- [3] Creswell, J. W. (1994). *Research design qualitative and quantitative approaches*.
- [4] Chatterjee, S., & Saha, D. (2021). Automated customer support using NLP: A survey. *Journal of King Saud University – Computer and Information Sciences*, 33(5), 497-508.
- [5] Zhang, Y., & Wang, S. (2016). A review of automatic customer service ticket classification. *Expert Systems with Applications*, 56, 311-319.
- [6] Pajares, F. (2007). Elements of a proposal.
- [7] Floridi, L., Cowls, J., Beltrametti, M., et al. (2018). AI4People—an ethical framework for a good AI society: Opportunities, risks, principles, and recommendations. *Minds and Machines*, 28, 689–707.
- [8] Kumar, A., & Sharma, A. (2020). Prioritizing support tickets using NLP and machine learning. *Procedia Computer Science*, 167, 2252-2261.
- [9] Kaggle Datasets: Multilingual Customer Support Tickets; Customer Support Ticket Dataset.
- [10] Springer Nature. *Ethics and Information Technology* journal.
- [11] Suresh, H., & Gutttag, J. V. (2019). A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002*.
- [12] Binns, R. (2018). Fairness in machine learning: Lessons from political philosophy. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*.