AI-Powered Customer Support and Lead Management Assistant: A Survey of Conversational AI, Product Recommendation, and Lead Scoring

Siddhesh Jadhav[1] Anish Bagwan[2] Rupesh Kawade[3] Harshal Chavan[4] Vaishnavi Khalbate[5]
[1,2,3,4,5]Department of Artificial Intelligence and Machine Learning
[1,2,3,4,5]Zeal Polytechnic, Pune, India

Abstract — E-commerce websites provide large product catalogs, but customers may struggle to identify products that match their needs, budget, and preferred features. Traditional customer support often relies on keyword search, fixed menus, or manual assistance, while lead-management teams may lack a consistent method for prioritizing potential buyers. This survey reviews conversational artificial intelligence, retrieval-augmented generation (RAG), product recommendation, and machine-learning-based lead scoring. It proposes an integrated assistant that accepts natural-language shopping queries, identifies intent and constraints, retrieves product and policy information, recommends suitable products, and creates an explainable lead-priority score for an authorized customer relationship management (CRM) workflow. The proposed design uses catalog-grounded responses, hard filters for price and availability, content-based ranking or cosine similarity, and supervised classification only when labelled historical data are available. A final-year prototype can be developed using Python and open-source tools. Evaluation should measure intent recognition, recommendation relevance, factual grounding, response time, and lead-scoring performance. No experimental performance is claimed in this survey; results must be established through testing.

Keywords: Conversational AI, Customer Support, E-Commerce, Lead Management, Lead Scoring, Natural Language Processing, Product Recommendation, Retrieval-Augmented Generation.

I. Introduction

Online shopping gives customers access to many products, but the size of a catalog can make comparison difficult. Customers commonly describe needs in natural language, such as a price limit, intended use, preferred brand, or desired features. Keyword-based search may not understand the full intent behind such requests, and static support menus may require customers to navigate several steps.

A conversational assistant can ask clarifying questions, retrieve relevant product records, and explain why selected products match the stated requirements. For order, delivery, return, and warranty questions, retrieval-augmented generation can ground responses in approved catalog and policy information instead of relying only on a language model's internal knowledge. Amazon Rufus is an example of a shopping assistant that motivates this general project direction; the proposed student prototype is independent and is not affiliated with Amazon or any commercial platform.

The second part of the problem is lead management. A lead is a potential customer who may be interested in a product or service. A lead score estimates the priority for follow-up using permitted interaction signals and, when available, historical conversion labels. This paper surveys the relevant approaches and proposes a unified workflow in which support, product recommendation, and lead prioritization are connected but independently evaluated.

II. Background and Related Work

A. Conversational AI for Customer Support

Conversational AI uses natural-language processing (NLP) to interpret user messages and produce relevant responses. A support assistant may classify intent, identify entities, maintain conversation context, and decide whether to answer, ask a follow-up question, or transfer the issue to a human. A hybrid AI-human design is useful because automation can handle routine requests while staff review complex, sensitive, or low-confidence cases [2].

B. Retrieval-Augmented Generation

Lewis et al. introduced retrieval-augmented generation as a method that combines a generative model with retrieved external knowledge [1]. In a customer-support setting, a retrieval component can search approved FAQs, return policies, product descriptions, and delivery information. The retrieved passages are supplied as context for the response. González Juclà et al. examine retrieval enhancements in a deployed customer-support chatbot and highlight that retrieval quality is important to answer quality [4]. The assistant should cite or identify the product or policy records used and should escalate when reliable evidence is unavailable.

C. Product Recommendation

Recommendation systems reduce a large catalog to a smaller set of items likely to meet a customer's requirements. Content-based methods compare product attributes with the customer's stated constraints or preferences. The proposed assistant can extract product category, budget, brand, size, colour, battery capacity, or intended use. Hard constraints—such as a maximum budget or required category—should be applied before ranking. Eligible items can then be ranked using attribute matching, cosine similarity, or a suitable recommendation method. A recommendation must not override explicit constraints merely because another product has a high similarity score.

D. Lead Scoring and CRM

Lead scoring estimates which potential customers should receive follow-up first. González-Flores et al. study machine-learning-based lead prioritization and provide evidence that lead scoring can be framed as a supervised prediction problem when suitable labelled data exist [3]. For this project, candidate signals may include explicit purchase intent, budget fit, repeated product comparisons, requested contact, and engagement with relevant items. These features must be collected transparently and used for the stated purpose. If historical conversion labels are unavailable, the prototype should use a clearly labelled rule-based score rather than

All rights reserved by www.ijsrd.com

presenting an untrained score as a validated machine-learning prediction.

E. Research Gap and Scope

The reviewed work addresses customer-support automation, retrieval, product recommendation, and lead prioritization as related but distinct tasks. The project-level opportunity is to connect these tasks in one explainable workflow: understand a shopping request, retrieve factual product information, recommend eligible items, answer supported questions, and provide a reasoned lead-priority estimate. This paper proposes a design and evaluation plan; it does not claim that the integrated prototype has already been implemented or experimentally validated.

III. System Requirements and Design Principles

The system accepts a customer's natural-language question and uses a product catalog containing product name, category, price, features, availability, and other relevant attributes. A separate knowledge base stores approved FAQs and support policies. The output includes a short response, matching products, the evidence used, and—where permitted—a lead-priority band with an explanation. The CRM component should only store information necessary for its declared purpose.

The main design principles are: (1) use verified catalog and policy records for factual answers; (2) apply hard filters before product ranking; (3) ask for clarification when essential constraints are missing; (4) display why a product or lead received its ranking; and (5) provide human escalation for unsupported or sensitive requests.

For a customer query q and product p, a simple content-based ranking can use cosine similarity between feature vectors:

$$\text{sim}(q,p) = [E(q) \cdot E(p)] / (\|E(q)\| \, \|E(p)\|) \quad (1)$$

Here E(q) and E(p) represent numerical or embedding-based representations of the customer's requirements and product attributes. Before ranking, the candidate set should satisfy hard constraints such as category and maximum price:

$$P_{\text{eligible}} = \{ p \in P \mid \text{category}(p)=c \text{ and } \text{price}(p) \leq \text{budget} \} \quad (2)$$

The system ranks only eligible products and should return "no suitable match" when no catalog item satisfies the required constraints. Lead scoring is a separate task. When labelled conversion data are available, a classifier may estimate P(convert = 1 | x), where x contains permitted customer-interaction features. The resulting score is a prioritization aid, not a guarantee that a customer will purchase.

IV. Proposed Architecture and Workflow

The proposed architecture separates the catalog and policy preparation stage from the online conversation stage. During preparation, product records are cleaned, numeric fields are validated, product features are standardized, and approved FAQs are divided into searchable passages. The product catalog supports structured filtering, while a text index or vector index can support semantic retrieval. A CRM store keeps only authorized lead records and score explanations.

At query time, the assistant identifies intent and extracts entities such as product category, budget, and desired features. If required details are missing, it asks a short follow-up question. The retrieval layer searches catalog and policy data; the recommendation module filters and ranks eligible products. A response generator summarizes the results using retrieved evidence. If the question is unresolved, confidence is low, or the issue requires staff action, the system escalates to a human. Lead signals are passed to the scoring module only with appropriate notice and authorization.

The workflow has the following main components: customer chat interface; intent and entity extraction; product/policy retrieval; constraint filtering and recommendation ranking; grounded response generation; human escalation; lead scoring; and CRM dashboard. These modules can be implemented incrementally and tested separately before integration.

CUSTOMER QUESTION → NLP: INTENT AND CONSTRAINTS
↓
PRODUCT / POLICY RETRIEVAL → FILTER BY BUDGET AND REQUIREMENTS
↓
RANK SUITABLE PRODUCTS → GROUNDED RESPONSE / CLARIFICATION
↓
HUMAN ESCALATION WHEN NEEDED → PERMITTED LEAD SCORE → CRM

Fig. 1: Text-based workflow of the proposed AI-powered customer support and lead management assistant.

The architecture must not invent product prices, stock status, discounts, delivery dates, or return terms. These values should come from the catalog, approved policy documents, or authorized APIs. Each recommendation should be traceable to product attributes, while support answers should be grounded in retrieved evidence. Lead scores should include a short explanation so staff can understand the factors influencing priority.

The CRM dashboard can show the customer or lead identifier, expressed interest, relevant products, permitted contact details, score band, and reasons for the score. Access should be limited to authorized staff. The prototype should not automatically reject customers, change prices, or make consequential decisions based solely on the score.

V. Methodology and Evaluation Plan

A. Survey Method

This paper uses a focused literature survey covering foundational RAG, customer-support chatbot design, retrieval improvements, and machine-learning-based lead scoring [1]–[4]. The studies are used to identify design patterns and limitations. The proposed project architecture and evaluation plan are derived from this review; they are not reported experimental findings from a completed implementation.

B. Prototype Components

A low-cost prototype can use Python, pandas for catalog processing, scikit-learn for baseline models and similarity calculations, a lightweight web framework for the chat

All rights reserved by www.ijsrd.com

interface, and a CSV or SQLite database for products and leads. A vector-search or embedding component may be added if time and computing resources allow. A simple rule-based intent classifier is acceptable as an initial baseline, with an NLP classifier added only when suitable labelled examples are available.

C. Baseline and Test Scenarios

The baseline can use keyword search and structured product filters. The enhanced version adds intent/entity extraction, similarity-based ranking, policy retrieval, and a lead-score view. Test cases should include clear requests, ambiguous queries, missing budgets, out-of-budget products, unavailable products, unsupported policy questions, repeated product comparisons, and cases requiring human escalation. The same test set should be used for fair comparison.

D. Evaluation Metrics

Intent recognition should be measured using accuracy and macro-F1. Product recommendation should be evaluated with Precision@k, Recall@k, or NDCG@k using labelled relevant products, together with a check that budget and category constraints are respected. Answer grounding should measure the proportion of factual claims supported by catalog or policy records, supported by human review. Support efficiency can be measured using median response time, unresolved-query rate, escalation rate, and user feedback.

Lead-scoring evaluation should use precision, recall, F1-score, ROC-AUC, and a confusion matrix when labelled historical outcomes are available. A held-out or time-aware split is preferable to evaluating on the same records used for training. If the project lacks historical labels, evaluate the rule-based score against a transparent rubric and staff feedback, and state that it is not a validated predictive model.

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (3)$$

Results should be reported only after the prototype is tested. The study should compare the keyword baseline with the enhanced assistant, document errors and unsupported answers, and report dataset size and limitations. Sales growth or business impact should not be claimed without a suitable real-world evaluation.

$$\text{Recommendation Precision@k} = \text{relevant recommended items} / k \quad (4)$$

A recommendation metric should be accompanied by constraint checks: the system must not recommend an item above the stated budget when the budget is a hard requirement. Queries with no eligible products should be evaluated separately so that the assistant is not rewarded for returning irrelevant products.

Lead-score evaluation should also examine calibration and score explanations where the available data permit. Fairness, privacy, and data quality should be reviewed before any use with real customers.

VI. Discussion

An integrated assistant can reduce repeated search effort by combining conversational clarification, structured filtering, and evidence-based responses. RAG can help keep support answers connected to approved sources, while content-based recommendation can make product suggestions more relevant to stated needs. Lead scoring may help staff prioritize follow-up, but its usefulness depends on the quality and representativeness of labelled data. These modules should remain separately testable so that a weak recommendation or score does not undermine the entire support workflow.

VII. Conclusion

This survey reviewed conversational AI, retrieval-augmented generation, product recommendation, and machine-learning lead scoring for an AI-powered customer support and lead management assistant. The proposed design combines intent extraction, catalog and policy retrieval, hard constraint filtering, product ranking, grounded responses, human escalation, and explainable lead prioritization. A small final-year prototype is feasible with open-source software and a carefully prepared dataset. Its performance must be demonstrated through controlled evaluation, with privacy, factual accuracy, and limitations documented clearly.

Acknowledgment

The authors may acknowledge their project guide, department, institute, and any organization that provides authorized data or evaluation support. Replace this sentence with acknowledgments that accurately reflect the project work.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 9459–9474.

[2] E. Mgbeahuruike, C. Ikotun, A. Ojo, E. Oyerinde, O. Famodimu, and T. Olorunoje, "Hybrid AI-Human Architecture for Real-Time Customer Support: Leveraging Retrieval-Augmented Generation," Asian Journal of Research in Computer Science, vol. 18, no. 7, pp. 88–105, 2025, doi: 10.9734/ajrcos/2025/v18i7722.

[3] L. González-Flores, J. Rubiano-Moreno, and G. Sosa-Gómez, "The relevance of lead prioritization: a B2B lead scoring model based on machine learning," Frontiers in Artificial Intelligence, vol. 8, art. 1554325, 2025, doi: 10.3389/frai.2025.1554325.

[4] D. González Juclà, M. Tuteja, M. Esteve Casademunt, K. Unnikrishnan, Y. Usmani, and A. Roshaan, "Retrieval Enhancements for RAG: Insights from a Deployed Customer Support Chatbot," in Proc. 19th Conference of the European Chapter of the Association for Computational Linguistics, vol. 5: Industry Track, 2026, pp. 169–180.

All rights reserved by www.ijsrd.com