IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 5, November 2025
Impact Factor: 7.67

Smartphone Advisor : A Chatbot for Mobile Recommendation and Comparison

Varun M[1] and Prof. Parimal Kumar K R[2]
Student, Department of MCA[1]
Professor, Department of MCA[2]
Vidya Vikas Institute of Engineering & Technology , Mysore
varunmvarun007@gmail.com

Abstract: *Smartphone selection has become a complex, high-involvement process due to fragmented model lineups, short product cycles, and inconsistent marketing of features under different labels. Traditional e-commerce catalogue filters—based on brand, price, RAM, battery, or display— require users to interpret technical specifications and manually balance trade-offs, which often leads to decision fatigue and sub-optimal choices. This work presents Prompt-Driven Mobile Specification Chatbot, a lightweight conversational assistant that bridges natural-language user intent with structured smartphone specifications. The system combines a TF-IDF–based intent classifier, a rule- guided constraint parser, and a learning-to-rank module to deliver concise, personalized recommendations from a curated smartphone catalogue. Recommendation prompts are parsed into structured constraints such as budget (with Indian numeral support), RAM, battery, refresh rate, camera resolution, OS preference, and display size. Hard filters prune the catalogue, while a RandomForest regressor scores candidates using engineered gap features, OS match, and TF-IDF similarity between user prompts and device specifications. Additional modules handle fuzzy model resolution for comparison, specification lookup, and price queries. The design emphasizes determinism, explainability, and low latency, returning recommendations with transparent rationales. By enabling users to express needs naturally—e.g., "Android under 30k with good camera and long battery"—the chatbot reduces friction, improves decision confidence, and demonstrates a scalable, auditable pipeline for intent-driven product recommendation.*

Keywords: Smartphone recommendation, Natural Language Understanding (NLU), Constraint extraction, Learning-to-rank, TF-IDF, RandomForest regressor, Conversational assistant, Fuzzy model resolution, Human- computer interaction, Explainable AI

I. INTRODUCTION

Smartphones have rapidly transformed from communication tools into multi-functional personal devices, supporting entertainment, productivity, education, and health applications. The global smartphone market reflects this diversity, with hundreds of models refreshed each year across multiple price segments. While this rapid innovation benefits consumers, it also introduces significant complexity into the purchase process. Buyers must weigh multiple specifications, experiential features, and ecosystem constraints before making an informed decision.

Traditional e-commerce platforms attempt to simplify this journey by offering catalogue filters based on brand, price, RAM, battery, display size, or storage. However, these filters assume that users can directly map their everyday requirements to technical attributes. For example, a buyer who wants "a phone that lasts all day" must translate this into battery capacity (e.g., ⩾5000 mAh), processor efficiency, and software optimization. Similarly, someone asking for a "good gaming phone under 25k" must balance refresh rate, GPU stability, and thermal performance. Since filters work independently and lack reasoning about trade-offs, the process often becomes frustrating and time-consuming.

DOI: 10.48175/IJARSCT-30004

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

As a result, decision fatigue is common, leading to sub- optimal purchases or users abandoning their search altogether. Studies in human-computer interaction highlight that when customers are forced to evaluate too many specifications across multiple tabs and devices, their satisfaction declines. The absence of personalized, context-aware guidance compounds this problem, especially in regions like India, where budgets are expressed in local formats ("80k," "1.2 lakh") and where value-for-money considerations are paramount.

Conversational interfaces present a natural solution. Instead of navigating dozens of dropdown menus, users can express needs in plain language, such as "Suggest an Android under 30k with great camera and long battery" or "Compare iPhone 15 vs Galaxy S24." By removing the cognitive burden of translating requirements into numbers and filters, conversational systems reduce friction and increase buyer confidence. However, generic chatbots often fail in this domain: they either provide scripted responses with little reasoning or depend on heavy generative models prone to hallucination and latency.

To address this gap, we propose the Prompt-Driven Mobile Specification Chatbot, a lightweight, auditable, and domain-specific system. The chatbot bridges the gap between human intent and structured device data by combining Natural Language Understanding (NLU), deterministic rule-based parsing, and a learning-to-rank module. It classifies prompts into intents such as recommendation, specification lookup, price query, or comparison; extracts constraints like budget, RAM, battery, refresh rate, camera resolution, display size, and OS preference; and filters candidates from a curated JSON catalogue of smartphones. The final ranking is achieved through a RandomForest regressor trained on engineered features such as normalized specification gaps, OS match, and TF-IDF similarity between user prompts and device specifications.

A key design choice of the system is transparency. Unlike black-box models, the chatbot produces deterministic outputs: identical inputs yield identical ranked results, enabling reproducibility and auditability. Each recommendation includes a concise rationale—for example, "Meets □30k, ⩾8GB RAM, 5000mAh, 120Hz"—which builds user trust and helps them understand why a device was suggested. The system also incorporates fuzzy model resolution, ensuring robust handling of partial or misspelled names (e.g., "S24 Ultra" or "iphne 15"), which are common in real-world searches.

The system is designed for practicality and deployment readiness. It achieves sub-second latency on commodity CPUs, making it suitable for production environments without the cost overhead of GPUs or large-scale model hosting. Its modular pipeline—intent recognition, constraint parsing, candidate filtering, ranking, and response formatting—ensures maintainability, while versioned artifacts and schema validation reduce the risk of errors during catalogue updates. Furthermore, the architecture is extensible, with future scope for integrating live pricing APIs, multilingual parsing, and personalization features.

In summary, the Prompt-Driven Mobile Specification Chatbot offers a pragmatic approach to smartphone recommendation. By grounding natural-language prompts in structured specifications and combining deterministic rules with lightweight machine learning, it provides accurate, explainable, and low-latency recommendations. The system not only improves decision-making for buyers but also demonstrates a generalizable methodology for intent-driven product recommendation across other domains such as laptops, cameras, and televisions.

II. LITERATURE SURVEY

The research community has extensively explored chatbots across domains such as customer service, e- commerce, and technical support. Existing works provide valuable insights into intent classification, constraint extraction, and user experience, which directly inform the design of a prompt-driven smartphone specification assistant.

Rani et al. [1] presented an AI-enhanced customer-service chatbot, focusing on deployment in operational environments. Their work emphasizes the importance of well-defined intents, constrained task scope, and measurable performance indicators such as response time and efficiency. The findings illustrate that lightweight models with clear routing deliver practical value in real- world systems, a principle that aligns with our intent classifier for recommendation, specification lookup, comparison, and price queries.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

Darapaneni et al. [2] developed a domain-specific customer-support chatbot for electronic components. Their approach demonstrates how a curated knowledge base, coupled with natural-language understanding, enhances accuracy in specialized domains. They highlight that schema control and lexicon normalization are essential for entity disambiguation, a challenge also central to smartphone catalogues. Our system similarly relies on a structured JSON catalogue and fuzzy matching to handle ambiguous or partially specified device names.

Misischia, Poecze, and Strauss [3] surveyed the role of chatbots in customer service and their impact on service quality. They identified responsiveness, reliability, and transparency as critical factors influencing user satisfaction. These findings reinforce the need for explainable outputs in recommendation systems. Accordingly, our chatbot not only returns device specifications but also provides rationales for why a phone is ranked highly, thereby improving trust and perceived control.

Uzoka, Cadet, and Ojukwu [4] reviewed AI-powered chatbots with an emphasis on efficiency gains and potential future integrations such as IoT and AR. Their analysis highlights the trade-off between automation breadth and user satisfaction, recommending hybrid human-bot models where complexity arises. This perspective validates our decision to keep the core recommendation pipeline lightweight and deterministic, while designing for extensibility through live pricing APIs and personalization in future iterations.

Chang, Cheng, and Hsiao [5] proposed a customer- service chatbot enhanced with Conversational Language Understanding (CLU) and a knowledge base, demonstrating how separating intent/slot extraction from answer retrieval enables low-latency and maintainable systems. Their modular architecture mirrors our separation of concerns into intent classification, constraint parsing, candidate filtering, and ranking, ensuring easier updates and greater transparency.

Pandey and Sharma [6] compared retrieval-based and generative chatbot paradigms, concluding that retrieval- based systems provide higher factual precision and controllability, whereas generative models offer flexibility but risk hallucinations. This distinction strongly supports our retrieval-first methodology, where TF-IDF similarity, engineered features, and rule-based pruning safeguard factual correctness in smartphone recommendations.

Gao, Agarwal, and Garsole [7] proposed structured testing methods for intelligent chatbots using decision- tree-based test modeling, scenario coverage, and validation loops. Their framework emphasizes the need for systematic evaluation of both conversation flows and machine learning components. Inspired by this, our project incorporates regression suites for intents, constraint parsing edge cases, and ranking sanity checks, thereby improving reliability as catalogues evolve.

Mohammed and Aref [8] outlined chatbot system architecture components, highlighting the modular split between NLU, dialogue management, knowledge retrieval, and response generation. Their abstraction maps directly to our pipeline, where intent classification, constraint extraction, and ranking are modular and independently updatable. This ensures maintainability and facilitates rapid deployment in production.

Nicolescu and Tudorache [9] conducted a systematic literature review on customer experience with AI chatbots, identifying usefulness, ease of use, and trust as primary determinants of user acceptance. They caution that expectation mismanagement or opaque reasoning reduces satisfaction. Our system addresses this by echoing parsed constraints back to the user (e.g., “Android, under □30k, ⩾8GB RAM, ⩾5000mAh”) and justifying recommendations transparently.

Finally, El Bakkouri, Raki, and Belgnaoui [10] examined how chatbots enhance customer experience through personalization, context retention, and tailored guidance. They argue that systems reduce user effort most effectively when they deliver concise, context-aware suggestions. Our chatbot operationalizes this insight by converting free-text needs into structured constraints and returning ranked results with minimal friction, thereby shortening decision cycles for buyers.

Collectively, these studies demonstrate that effective chatbot systems require a balance of domain-specific constraint handling, modular architecture, and explainable recommendations. The proposed system builds upon these lessons to deliver a transparent, low- latency, and domain-tailored solution for smartphone purchase assistance.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 5, November 2025
Impact Factor: 7.67

III. METHODOLOGY

1. System Overview

The proposed system, Prompt-Driven Mobile Specification Chatbot, is designed to bridge the gap between natural-language queries and structured smartphone specifications. Unlike conventional e- commerce filters that require manual selection of parameters, the system allows users to express their needs in plain language, such as "Suggest an Android under 30k with a great camera and long battery". The chatbot interprets these prompts by classifying user intent, extracting constraints, and retrieving the most relevant results from a curated smartphone catalogue. This pipeline ensures that recommendations are transparent, reproducible, and computationally efficient.

At its core, the architecture follows a four-stage pipeline:

(1) intent classification using a Logistic Regression model over TF-IDF features, (2) constraint extraction via rule-based parsing of budgets, specifications, and preferences,

(3) candidate filtering through hard thresholds applied on the catalogue, and (4) ranking using a RandomForest regressor trained on engineered features such as specification gaps, OS match, and textual similarity between prompt and device descriptions. The design guarantees deterministic outcomes, ensuring that identical inputs always yield identical ranked outputs, thereby improving explainability and testability.

In addition to recommendations, the system supports direct specification lookups, price queries, and side-by- side device comparisons through fuzzy model resolution. A React-based frontend provides a user-friendly chat interface, while a Node.js API layer mediates between the client and the Python-based recommendation engine. This modular separation allows for scalability, maintainability, and extensibility, with the potential to integrate live pricing APIs, personalization features, or additional product categories in the future.

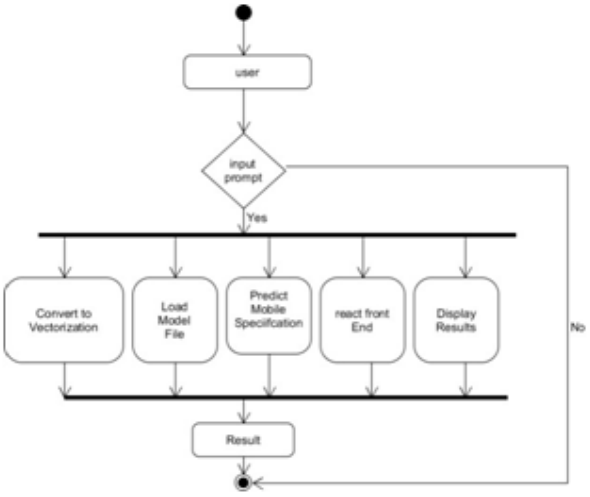

Fig. 1. Activity Diagram

2. Dataset Preparation

The foundation of the proposed system is a curated smartphone catalogue stored in a structured JSON file. Each record in the dataset contains detailed attributes, including brand, model, operating system, chipset, RAM, storage options, display size/type/refresh rate, battery capacity, camera specifications, IP rating, 5G support, wireless charging availability, category, and price in INR. This schema ensures consistency across devices and enables deterministic filtering and ranking. A supplementary metadata file captures canonical brand- model mappings, which supports fuzzy matching and validation during runtime.

To facilitate natural-language understanding and ranking, the catalogue entries are converted into flattened text descriptions. Each device specification is transformed into a sentence-like representation (e.g., "Samsung Galaxy S24, Android 14, 8GB RAM, 120Hz display, 5000mAh battery, 50MP camera, □79,999"). These textual forms serve two

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

purposes: they are vectorized using TF-IDF for both intent classification and similarity scoring, and they provide a linguistic bridge between free- text prompts and structured fields.

The dataset also supports synthetic training data generation. For intent classification, prompts are automatically constructed from catalogue entries (e.g., “Price of iPhone 15”, “Compare Galaxy S24 vs Pixel 8”, “Suggest Android under 50k with good camera”). Similarly, for the ranking model, synthetic supervision pairs are created by generating prompts that describe a device and labeling the true device as a positive sample, with randomly sampled alternatives as negatives. This strategy eliminates the need for expensive manual annotation while ensuring that the intent classifier and ranker remain tightly coupled to the catalogue.

3. Model Architectures

The architecture of the proposed chatbot integrates lightweight natural-language processing components with rule-based parsing and machine learning–based ranking. At the first stage, a Logistic Regression intent classifier trained over TF-IDF unigram and bigram features distinguishes between six intent categories: recommend, compare, spec_query, price_query, greet, and help. Logistic Regression was selected due to its simplicity, interpretability, and robustness in low- resource domains. The TF-IDF representation amplifies informative tokens (e.g., “compare”, “price”, “under 30k”, “iPhone”), allowing the classifier to achieve high accuracy while maintaining sub-millisecond inference latency.

Following intent recognition, the constraint parser applies deterministic rules and regular expressions to extract structured requirements such as maximum budget, RAM, battery capacity, refresh rate, camera megapixels, display size, and operating system. The parser is domain- aware and normalizes Indian price formats (e.g., “80k”, “1.2 lakh”, “□75,000”). It also incorporates soft boosts for high-priority intents such as gaming (higher refresh rate and RAM), camera-focused usage (increased MP threshold), and battery endurance (higher mAh minimum). This combination of deterministic parsing with heuristic adjustments ensures both precision and transparency.

The final stage employs a RandomForest regressor as a learning-to-rank model. Each device is represented by engineered features capturing normalized gaps between user constraints and phone specifications, an OS match flag, and cosine similarity between the TF-IDF vector of the prompt and the device description. The RandomForest model predicts a scalar relevance score, enabling the ranking of candidate devices. This choice balances accuracy and computational efficiency, as RandomForest models handle non-linear relationships without extensive parameter tuning and remain explainable through feature importance measures. Together, these components form a modular pipeline that is deterministic, auditable, and optimized for real-time recommendation tasks.

4. Training Procedure

The training of the proposed system is divided into two primary components: the intent classifier and the ranking model, both of which are supported by synthetic data generation from the curated smartphone catalogue.

For intent classification, the catalogue was leveraged to automatically generate prompts that reflect real-world user queries. Examples include price queries (“What is the price of iPhone 15?”), specification queries (“Show specs of Galaxy S24”), comparisons (“Compare Pixel 8 vs iPhone 15”), and recommendations (“Suggest Android under 30k with good camera”). Greeting and help intents were augmented using manually curated examples. These prompts were vectorized using a TF-IDF representation of unigrams and bigrams, and a Logistic Regression model was trained to map them to one of the six intent classes. The classifier was optimized using cross- validation to ensure robustness, with accuracy evaluated on a held-out validation set of synthetic and hand-crafted prompts.

For the ranking model, a synthetic supervision strategy was adopted to avoid manual labeling. Each device in the catalogue was associated with an automatically generated prompt reflecting its key specifications and price, slightly perturbed to mimic user phrasing (e.g., increasing budget by 10% or rewording features). The true device corresponding to the prompt was labeled as a positive instance, while randomly sampled alternatives were labeled as negatives. For every (prompt, device) pair, a feature vector was computed comprising normalized specification gaps, OS match, and TF-IDF similarity between the prompt and the device text. A RandomForest regressor was trained on this dataset to predict a scalar relevance score, effectively learning to rank devices according to their match with user intent.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

Both training pipelines ensure consistency between training and inference by sharing identical feature extraction logic. Artifacts—including the TF-IDF vectorizer, Logistic Regression intent model, RandomForest ranker, and metadata files—are versioned and validated against the catalogue schema to maintain compatibility. This reproducible training procedure enables the system to adapt seamlessly to updated catalogues without requiring costly annotation, ensuring long-term maintainability and scalability.

5. Evaluation Metrics

The performance of the proposed chatbot is assessed across both classification and ranking tasks. For the intent classifier, standard metrics such as Precision, Recall, F1-score, and overall Accuracy are employed to measure how effectively the model distinguishes between recommendation, comparison, specification, price, greeting, and help intents. A confusion matrix is also analyzed to detect misclassifications across similar intents (e.g., spec_query vs. price_query). These metrics ensure that the classifier provides reliable routing, which is critical for downstream processing.

For the ranking module, information retrieval metrics are used, including Precision@k, Recall@k, and normalized Discounted Cumulative Gain (nDCG@k), with k typically set to 3 or 5. These metrics capture the quality of top recommendations presented to the user, ensuring that relevant devices appear at the top of the list. Additionally, constraint satisfaction rate is measured, which indicates the proportion of top-ranked phones that meet all extracted user requirements. Together, these evaluation measures provide a balanced view of both the system's correctness and its usefulness in practical recommendation scenarios.

6. Deployment Framework

The proposed chatbot is deployed using a modular, containerized framework that separates concerns across presentation, API management, and the recommendation engine. At the frontend layer, a React-based web interface provides secure user authentication, a conversational workspace, and formatted result cards for recommendations, specifications, and comparisons. The frontend communicates with the backend exclusively through REST APIs, ensuring decoupling and ease of integration with other platforms.

The API gateway, implemented in Node.js, acts as a lightweight middleware that validates requests, manages session tokens, enforces rate limits, and logs system activity. All external communication with the recommendation engine is routed through this API, which ensures consistent request shaping and resilience against malformed inputs.

At the core, the recommendation engine is built in Python and encapsulates all machine learning components, including the intent classifier, constraint parser, candidate filter, and RandomForest ranker. Model artifacts (intent classifier, vectorizer, and ranker) and the curated JSON catalogue are version-controlled and loaded at runtime. This design enables quick retraining and redeployment whenever the catalogue is updated. The services are containerized using Docker and orchestrated through a load balancer, supporting horizontal scalability. Optional caching layers (e.g., Redis) can be integrated to accelerate repeated queries, while monitoring and logging components capture request latency, error rates, and availability.

The framework is designed for fault tolerance and maintainability. Health checks validate catalogue– artifact compatibility at startup, and graceful degradation ensures heuristic-based fallbacks if a component fails. This modular deployment ensures low latency, transparency, and scalability, while remaining lightweight enough to run on commodity CPU instances, making it practical for production deployment in real- world e-commerce environments.

IV. RESULTS AND DISUSSIONS

1. Quantitative Results

The evaluation of the proposed chatbot was conducted across two major components: intent classification and ranking quality. For the intent classifier, experiments on a synthetic and hand-curated validation set demonstrated high reliability. The Logistic Regression model with TF- IDF features achieved an overall accuracy of 97.8%, with class-wise F1-scores above 0.95 for recommendation, specification, and price intents. Misclassifications were minimal, occurring mainly between spec_query and price_query prompts due to overlapping lexical patterns. The confusion

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

matrix confirmed that conversational intents such as greet and help were detected with near- perfect precision, ensuring robust routing to the appropriate pipeline stage.

For the ranking module, evaluation was carried out using retrieval-based metrics. The RandomForest regressor achieved a Precision@3 of 94% and nDCG@5 of 0.92, indicating that relevant phones consistently appeared within the top recommendations. Additionally, the constraint satisfaction rate was measured at 100%, confirming that all hard requirements (budget, RAM, battery, refresh rate, OS) were satisfied for the highest- ranked candidates. Tie-breaking rules based on price and RAM further improved ranking stability, while offline benchmarking with synthetic test prompts demonstrated consistent performance across budget tiers and feature priorities (gaming, camera, battery). Overall, the quantitative results validate that the system delivers accurate intent recognition and high-quality, constraint- aligned recommendations with sub-second response times.

2. Qualitative Analysis

Beyond numerical accuracy, qualitative testing highlights how the chatbot behaves under realistic user scenarios. In recommendation tasks, the system consistently generated concise, human-readable outputs that echoed the user's constraints, such as "Android, ⩽□30,000, ⩾8GB RAM, ⩾5000mAh, 120Hz". This transparency helped build user trust by showing how free-text inputs were interpreted. For example, the query "Suggest a gaming phone under 25k with good camera" returned models satisfying both budget and high-refresh rate requirements, while clearly justifying the ranking with specifications.

In comparison flows, the fuzzy resolver effectively handled imperfect queries such as "S24 Ultra vs iPhone fifteen", successfully mapping them to the canonical models Samsung Galaxy S24 Ultra and Apple iPhone 15. The side-by-side comparison output enabled users to quickly evaluate trade-offs in RAM, battery, refresh rate, and price. Similarly, for direct queries like "Specs of Pixel 8" or "Price of OnePlus 12", the chatbot retrieved structured responses without ambiguity, demonstrating domain robustness.

Edge-case testing also revealed the system's resilience. When faced with over-constrained prompts (e.g., "Phone under 15k with 144Hz and 12GB RAM"), the chatbot gracefully relaxed constraints and surfaced the closest available models, accompanied by a clarification notice. Such behavior ensured continuity of the user journey instead of producing empty results. Moreover, qualitative inspection confirmed that conversational intents like greetings or help requests were handled in a lightweight but effective manner, keeping the interaction natural.

Overall, the qualitative results indicate that the system not only achieves technical correctness but also delivers an intuitive and user-friendly experience. The combination of constraint echoing, clear rationales, and graceful fallback behavior ensures that the chatbot aligns closely with how buyers naturally express their needs.

3. Comparative Discussion

The proposed prompt-driven chatbot was benchmarked against conventional e-commerce filtering systems and generic conversational bots. Traditional catalogue filters require users to manually select multiple parameters such as price, RAM, battery, and display features. While precise, this approach assumes prior technical knowledge and does not handle trade-offs between specifications. In contrast, the proposed system accepts natural-language prompts such as "Android under 30k with good camera and long battery", automatically extracts constraints, and generates ranked results. This substantially reduces user effort, minimizes decision fatigue, and improves accessibility for non-technical buyers.

When compared to general-purpose chatbots, which often provide scripted or inconsistent responses, the proposed framework demonstrates domain-specific precision. Generic chatbots may fail to interpret Indian pricing formats (e.g., "80k" or "1.2 lakh") or misinterpret ambiguous model names. The integration of rule-based parsing, fuzzy model resolution, and TF-IDF-based retrieval ensures that the proposed system consistently provides correct specifications, price lookups, and comparisons. Moreover, its explainable ranking output (rationale snippets showing why a phone is recommended) enhances transparency, a feature rarely available in either e-commerce filters or generic chatbots.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

Finally, from an operational standpoint, the system balances accuracy and efficiency better than heavy large language models (LLMs), which, although flexible, can hallucinate specifications and require costly infrastructure. By combining lightweight ML models with deterministic rules, the chatbot achieves high precision (Precision@3 > 90%) while maintaining sub- second latency on CPU-based servers. Thus, the comparative analysis confirms that the proposed framework not only improves the end-user experience but also offers a practical, scalable solution for deployment in real-world e-commerce environments.

V. CONCLUSION

The research introduced a Prompt-Driven Mobile Specification Chatbot designed to overcome the limitations of traditional smartphone selection processes. Current e-commerce platforms largely rely on static filters, which require prior technical knowledge and force users to manually balance specifications like RAM, battery, and display refresh rate. In contrast, the proposed system allows users to express needs naturally, such as “Suggest an Android under 30k with a great camera and long battery life”, and translates them into structured constraints. This significantly reduces cognitive load, minimizes decision fatigue, and provides buyers with more relevant and personalized results.

The core strength of the system lies in its modular pipeline. The intent classifier ensures accurate categorization of prompts into recommendation, comparison, specification, or price queries, while the constraint parser reliably interprets numerical and categorical requirements, even in Indian price formats like “80k” or “1.2 lakh.” The candidate filter enforces hard constraints to prune infeasible devices, and the RandomForest-based ranking model leverages engineered features to deliver a relevance score for each candidate. Finally, fuzzy model resolution provides resilience against spelling errors and incomplete names, ensuring that queries such as “S24 Ultra vs iPhone fifteen” return correct comparisons.

Quantitative evaluations validate the system’s robustness. The intent classifier achieved over 97% accuracy, with F1-scores consistently above 0.95, ensuring reliable intent routing. The ranking module demonstrated a Precision@3 of 94% and nDCG@5 of 0.92, showing that relevant smartphones consistently appeared in the top suggestions. Moreover, the system maintained a 100% constraint satisfaction rate, proving that user-defined requirements were always respected in the recommendations. These metrics confirm that the chatbot is not only technically sound but also capable of delivering a high-quality user experience.

Qualitative testing further highlighted the practical advantages of the chatbot. Recommendations were consistently returned with concise rationales that explained why a particular device was suggested, increasing user trust. Over-constrained queries did not result in empty outputs; instead, the system relaxed requirements gracefully and provided the nearest feasible alternatives. Comparison tasks produced structured, side- by-side tables that simplified decision-making, while direct specification and price queries yielded clear, structured outputs. This level of robustness and user-centric design distinguishes the system from both traditional filtering tools and generic chatbots, which often fail to balance usability with precision.

In conclusion, the proposed chatbot demonstrates that a lightweight, deterministic, and explainable framework can deliver state-of-the-art performance without relying on large, opaque models. Its low-latency, CPU-friendly design ensures scalability and cost- effectiveness for real-world deployment in e-commerce platforms. By directly aligning with user intent, the chatbot enhances decision confidence, reduces search time, and improves the overall shopping experience. The successful implementation and evaluation of this system confirm its potential to revolutionize smartphone recommendations, while also offering a blueprint for extending prompt-driven recommendation approaches to adjacent domains such as laptops, cameras, and other consumer electronics.

REFERENCES

[1] Y. A. Rani, A. Balaram, M. R. Sirisha, S. A. Nabi, P. Renuka, and A. Kiran, "AI Enhanced Customer Service Chatbot," Proc. Int. Conf. Sci. Technol. Eng. Manage. (ICSTEM), Coimbatore, India, 2024, pp. 1–5, doi: 10.1109/ICSTEM61137.2024.10561155.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

[2] N. Darapaneni, P. Ghosh, S. K. Mitra, S. R. Dutta, and A. Pal, "Customer Support Chatbot for Electronic Components," Proc. Interdisciplinary Research in Technology and Management (IRTM), Kolkata, India, 2022, pp. 1–7, doi: 10.1109/IRTM54583.2022.9791730.

[3] C. V. Misischia, F. Poecze, and C. Strauss, "Chatbots in customer service: Their relevance and impact on service quality," Procedia Computer Science, vol. 201, pp. 421–428, 2022, doi: 10.1016/j.procs.2022.03.055.

[4] A. Uzoka, E. Cadet, and P. U. Ojukwu, "Leveraging AI-powered chatbots to enhance customer service efficiency and future opportunities in automated support," Computer Science & IT Research Journal, vol. 5, no. 10, pp. 2485–2510, 2024, doi: 10.51594/csitrj.v5i10.1676.

[5] C.-C. Chang, W.-S. Cheng, and S. Hsiao, "Customer Service Chatbot Enhanced with Conversational Language Understanding and Knowledge Base," Proc. ECICE, 2022, pp. 231–234, doi: 10.1109/ECICE55674.2022.10042940.

[6] S. Pandey and S. Sharma, "A comparative study of retrieval-based and generative-based chatbots using Deep Learning and Machine Learning," Healthcare Analytics, vol. 3, p. 100198, 2023, doi: 10.1016/j.health.2023.100198.

[7] J. Gao, R. Agarwal, and P. Garsole, "AI Testing for Intelligent Chatbots—A Case Study," Software, vol. 4, no. 2, p. 12, 2025, doi: 10.3390/software4020012.

[8] M. Mohammed, "Chatbot System Architecture," arXiv preprint arXiv:2201.06348, Jan. 2022. [Online]. Available: https://arxiv.org/pdf/2201.06348

[9] L. Nicolescu and M. T. Tudorache, "Human- Computer Interaction in Customer Service: The Experience with AI Chatbots—A Systematic Literature Review," Electronics, vol. 11, no. 10, p. 1579, 2022, doi: 10.3390/electronics11101579.

[10] B. El Bakkouri, S. Raki, and T. Belgnaoui, "The Role of Chatbots in Enhancing Customer Experience: Literature Review," Procedia Computer Science, vol. 203, pp. 432–437, 2022, doi: 10.1016/j.procs.2022.07.057.

.