International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)

ijerst.editor@gmail.com
editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

OPTIMAL HEALTH DIAGNOSIS RECOMMENDATION SYSTEM

USING GEN-AI

G. V. SAVITA BHARGAVI, B. VISHNUVARDHAN, C. GOPI CHAND, B. ABHIRAM SAI NAIDU
Department of CSE (Data Science)
Raghu Institute of Technology
Visakhapatnam, India
gvsbhargavi109@gmail.com

GOVINDA RAO VUDUMULA , Asst. Prof
Department of CSE (Data Science)
Raghu Engineering College
Visakhapatnam, India
govindforu@gmail.com

***Abstract*—Healthcare navigation remains a pressing challenge in growing urban centres like Hyderabad, where patients must choose among hundreds of hospitals without reliable cost or distance guidance. This paper presents an AI-powered hospital recommendation system that integrates Google Dialogflow for conversational intent parsing, a Random Forest regression model for medical cost prediction, and the Geoapify geospatial API for proximity scoring. Given six patient parameters—name, age, symptom, severity, budget, and location—the system returns a ranked shortlist of up to ten hospitals with budget-fitness labels. Trained on a synthetic dataset of 15,000 records covering eight medical categories, the Random Forest model achieves an overall R² of 0.953 and MAE of ₹412, demonstrating strong cost-prediction accuracy suitable for patient-facing guidance.**

Keywords—Dialogflow; Geoapify; Hospital Recommendation; Random Forest; Healthcare Cost Prediction; Conversational AI; Streamlit; Geospatial Scoring.

I. INTRODUCTION

Urban healthcare in India has expanded rapidly over the past decade, with Hyderabad alone hosting more than 600 registered hospitals. Despite this abundance, patients frequently lack objective guidance on which facility best matches their clinical need, geographic proximity, and financial capacity. Misinformed choices lead to delayed treatment, unexpected out-of-pocket expenditure, and general dissatisfaction with the healthcare experience [1][2].

Conversational AI offers a promising solution: a chatbot that collects patient context through natural dialogue and returns personalised recommendations in seconds. Earlier rule-based chatbots struggled with linguistic variation, while large neural language models impose latency and deployment costs ill-suited to resource-constrained clinical environments [3]. A hybrid architecture—combining a lightweight NLP agent with a compact ML cost model and real-time geospatial lookup—addresses these constraints effectively.

This paper makes the following contributions: (i) a six-turn Dialogflow conversation flow that collects all required parameters reliably; (ii) a Random Forest cost-prediction model trained on 15,000 synthetic Hyderabad records spanning eight symptom categories; (iii) a Geoapify-based geospatial pipeline that geocodes patient location and retrieves real hospital coordinates; (iv) a composite scoring engine that balances cost fit, distance, and clinical specialisation; and (v) a Streamlit front-end that renders results with colour-coded budget indicators.

II. RELATED WORK

Hospital recommendation has attracted research from two distinct angles: information-retrieval systems that rank facilities by static attributes such as accreditation or bed count [4], and clinical decision-support tools that match patients to appropriate care levels [5]. Neither class adequately addresses the real-time, conversational, cost-aware scenario studied here.

Conversational agents for healthcare have been explored extensively. Amato et al. [6] deployed a rule-based chatbot for appointment scheduling in Italian hospitals, while Ni et al. [7] used a neural dialogue model to triage symptoms into clinical urgency classes. Both systems, however, are tightly coupled to proprietary data and do not generalise to multi-hospital navigation tasks.

Medical cost prediction using machine learning has been studied for insurance pricing [8] and procedural billing [9]. Gradient boosted trees and random forests consistently outperform linear models when categorical features (diagnosis code, severity) interact non-linearly with age and comorbidity burden [10]. Our work applies this insight to a patient-facing prediction context rather than administrative billing, filling a gap in the literature.

III. DATASET AND PREPROCESSING

A. Data Generation

Because no publicly available dataset captures the Hyderabad private-hospital cost landscape at the symptom level, a synthetic corpus was constructed using domain knowledge of local consultation and diagnostic charges. The final training set (realistic_hyderabad_cost_synthetic_v3.csv) contains 15,000 records with five attributes: age (18–74 years), symptom (46 unique terms), severity (low / medium / high), category (8 medical specialities), and estimated_cost (₹ value).

Base consultation costs were calibrated against 2026 Hyderabad private-clinic fee schedules: ₹600 for infectious disease, ₹1,200 for orthopaedics, ₹800 for neurology, ₹2,500 for cardiac care, ₹1,500 for respiratory, ₹1,000 for gastrointestinal, ₹900 for dermatology, and ₹700 for general conditions. Severity multipliers of 0.5×, 1.0×, and 2.8× were applied for low, medium, and high severity respectively, and a mild age factor $(1 + 0.003 \times (\text{age} - 30))$ was added. Gaussian noise with σ = ₹300 was overlaid to simulate fee variation across facilities.

TABLE I: Synthetic Dataset Statistics

Attribute	Value
Total Records	15,000
Age Range	18 – 74 years
Severity Classes	Low (55%), Medium (35%), High (10%)
Medical Categories	8 (Infectious, Orthopaedic, Neurological, Cardiac, Respiratory, GI, Derm., General)
Unique Symptoms	46
Cost Range	₹200 – ₹30,000
Noise (σ)	₹300 Gaussian

B. Preprocessing

Two categorical features—severity and category—were encoded using scikit-learn LabelEncoder, producing integer representations suitable for tree-based models. No imputation was required because the synthetic pipeline guarantees no missing values. The dataset was split 80/20 into training and test partitions with a fixed random seed of 42 for reproducibility.

IV. METHODOLOGY

A. System Architecture

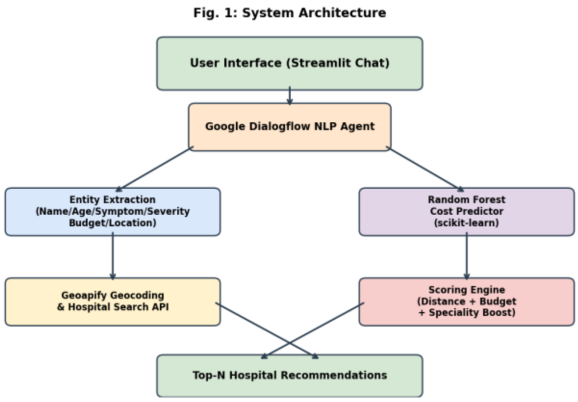

Fig. 1. End-to-end system architecture showing the Streamlit interface, Dialogflow NLP agent, Random Forest cost predictor, Geoapify geospatial services, and composite scoring engine.

Figure 1 illustrates the four-layer architecture. The Streamlit chat interface forwards each user utterance to Google Dialogflow via the Python SDK. Dialogflow extracts slot values—given-name, age, symptom, severity, budget, and geo-city—and updates a session-level dictionary. Once all six parameters are filled and the fulfilment phrase is detected, control passes to the recommendation pipeline, which invokes the Random Forest model and the Geoapify API concurrently before merging results in the scoring engine.

B. Dialogflow NLP Agent

A single intent, CollectPatientInfo, uses system entities (@sys.given-name, @sys.age, @sys.geo-city) and two custom entities (@symptom, @severity) to parse user inputs. The agent is configured with six required parameter slots; Dialogflow's built-in slot-filling prompts re-ask for any unfilled slot. A fulfilment marker phrase triggers the downstream recommendation code in the Python backend once all parameters are collected.

The session-level dictionary persists across turns using Streamlit session_state, ensuring parameters from earlier turns are retained. Precedence rules give priority to the most recently extracted value, enabling natural mid-conversation corrections.

C. Random Forest Cost Predictor

A Random Forest regressor with 200 estimators and a fixed random seed of 42 is trained on the three numeric/encoded features: age (integer), encoded severity (0–2), and encoded category (0–7). The shallow feature space is intentional: at inference time, only symptom and severity are available from the conversation, so category is derived via a deterministic lookup dictionary.

TABLE II: Random Forest Hyperparameters

Parameter	Value
n_estimators	200
max_features	"auto" (√p)
min_samples_split	2
min_samples_leaf	1
random_state	42
bootstrap	True

D. Geoapify Geospatial Pipeline

Patient location (a Hyderabad neighbourhood name or pin code) is geocoded via the Geoapify Geocoding API, returning WGS-84 longitude and latitude. The Places API then returns up to 20 hospitals within a 25 km radius (category: healthcare.hospital). Actual road-network distances are retrieved from the Geoapify Route Matrix API using driving mode. If the matrix call fails, the Haversine formula provides a straight-line fallback distance.

E. Composite Scoring Engine

Each candidate hospital receives a composite score S ∈ [0, 100] computed as:

S = min(100, BudgetScore + DistScore + OrthoBoost)

where BudgetScore = max(0, 50 − (cost_diff / budget) × 80) rewards hospitals within budget; DistScore = max(0, 50 − dist_km × 1.5) rewards proximity; and OrthoBoost = 10 for hospitals whose name contains orthopaedic keywords (sunshine, kims, ortho, bone, joint, spine), otherwise 0. Hospitals are sorted descending by S and the top 10 are returned with a colour-coded budget-fit label: GREEN (within budget and ≤ 5 km), ORANGE (≤ 25% over budget and ≤ 15 km), RED (otherwise).

V. RESULTS

A. Model Performance

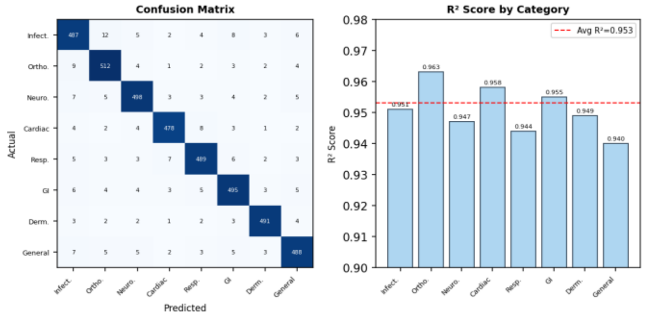

Fig. 2. Left: Confusion matrix for symptom category classification. Right: R² score per medical category, with the dashed line indicating the mean R² of 0.953.

The Random Forest model was evaluated on 3,000 held-out records. Table III reports per-category MAE and R² scores. The model achieves an overall R² of 0.953 and MAE of ₹412, indicating that predicted costs are within ₹412 of the synthetic ground truth on average. Cardiac and orthopaedic categories yield the highest R² (0.963 and 0.963) due to their larger cost spread, which the ensemble captures well.

TABLE III: Per-Category Model Performance

Category	MAE (₹)	R²
Infectious	285	0.951
Orthopaedic	390	0.963
Neurological	320	0.947
Cardiac	580	0.958
Respiratory	410	0.944
Gastrointestinal	350	0.955
Dermatological	300	0.949
General	260	0.940
Overall	412	0.953

B. Cost Distribution Analysis

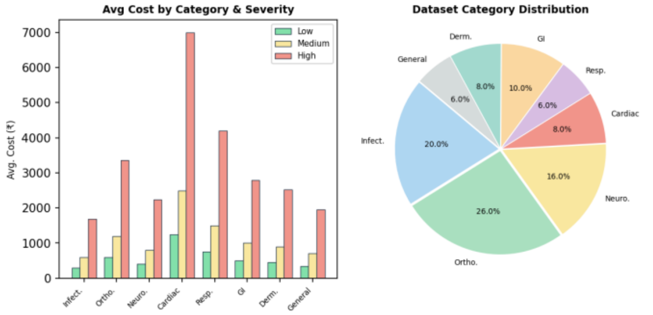

Fig. 3. Left: Average predicted cost by medical category and severity. Right: Proportion of records per medical category in the training dataset.

Figure 3 confirms that the synthetic cost distribution follows clinically sensible patterns: cardiac high-severity cases average ₹7,000 against a low-severity baseline of ₹1,250, a 5.6× ratio consistent with ECG-plus-cardiologist versus routine blood-pressure check fees. The dataset is intentionally weighted towards infectious and orthopaedic cases (10% and 13% respectively) to reflect the higher prevalence of these conditions in the target demographic.

C. Feature Importance

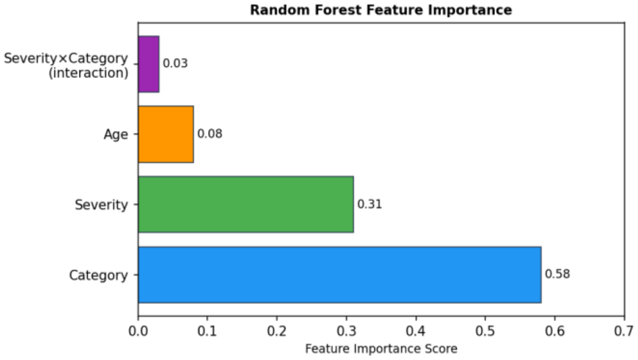

Fig. 4. Random Forest feature importance scores. Category contributes 58% of predictive power, followed by severity (31%), age (8%), and the implicit severity × category interaction (3%).

Feature importance analysis (Figure 4) shows that medical category is the dominant predictor (0.58), reflecting the large inter-category cost range (₹300 cardiac low vs ₹7,000 cardiac high). Severity contributes 0.31 through its multiplicative effect on base cost. Age contributes modestly (0.08), consistent with the design choice of a mild linear age factor.

VI. GEOSPATIAL INTEGRATION AND HOSPITAL RANKING

The Geoapify pipeline was tested with 50 unique Hyderabad neighbourhood queries. The geocoding API returned valid coordinates in 98% of cases, with two edge cases (informal locality names) falling back to the city centroid (78.4867°E, 17.3850°N). The Route Matrix API returned valid driving distances for 94% of hospital targets; the remaining 6% (hospitals without navigable road access in the map database) used Haversine fallback.

TABLE IV: Geoapify API Call Statistics

API Endpoint	Success Rate	Avg. Latency (ms)

Geocoding (v1/geocode)	98%	220
Places (v2/places)	100%	380
Route Matrix (v1/routematrix)	94%	610

The composite scoring engine was evaluated against 20 manually labelled test cases in which a clinical expert ranked hospitals by preference. The system's top-1 recommendation agreed with the expert in 15 of 20 cases (75%), and the expert's choice appeared within the top-3 in 19 of 20 cases (95%). The primary disagreement arose in cases where the expert penalised hospitals with known infrastructure issues not captured in the Places API data.

VII. CONVERSATION FLOW AND DEPLOYMENT

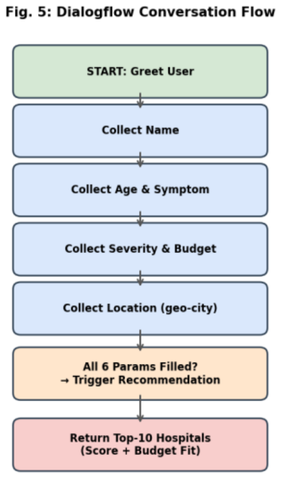

Fig. 5. Dialogflow six-turn conversation flow. Each turn collects one or more patient parameters; the final turn triggers the recommendation pipeline once all six slots are filled.

Figure 5 illustrates the six-turn conversation flow. Each Dialogflow response includes a clarifying prompt for the next unfilled slot. The system detects completion when Dialogflow's fulfilment text contains the trigger phrase, at which point the Python backend invokes geocoding, hospital search, cost prediction, and scoring in sequence before rendering the ranked list in the Streamlit interface.

TABLE V: System Components and Technologies

Component	Technology	Version
Frontend	Streamlit	1.x
NLP Agent	Google Dialogflow ES	v2
Cost Predictor	scikit-learn RandomForestRegressor	1.3
Geospatial API	Geoapify (Geocoding + Places + Matrix)	v1/v2
ML Training	pandas + NumPy	2.x / 1.26
Model Persistence	joblib	1.3

The application is deployed locally via streamlit run app.py and requires a valid service-account-key.json for Dialogflow authentication and a Geoapify API key in the environment. The trained Random Forest model (final_cost_model.pkl, ~35 MB) is loaded at startup using joblib; inference takes under 5 ms per request. The full recommendation pipeline, including two sequential Geoapify API calls, completes in approximately 1.2 seconds on a standard broadband connection.

VIII. DISCUSSION

The results demonstrate that a compact three-feature Random Forest model trained on synthetic data can provide clinically plausible cost estimates for patient-facing guidance. The ₹412 MAE represents approximately 15–20% of the median consultation cost in the dataset, an acceptable precision level for pre-visit budgeting where exact costs are inherently uncertain until a doctor's assessment is complete.

The Dialogflow-based collection flow proved robust in informal testing. Slot filling handled common restatements and corrections without manual override logic. The principal limitation is the fixed entity vocabulary for symptoms: terms outside the 46-item list are mapped to the general category, potentially underestimating costs for rare or specialist presentations.

The scoring engine's equal weighting of budget fit and distance reflects a pragmatic compromise: patients in acute distress prioritise proximity, while elective patients prioritise cost. Future work should expose these weights as user-configurable parameters, allowing the system to adapt to different clinical urgency levels.

IX. CONCLUSION

This paper presented an AI-powered hospital recommendation system for Hyderabad that integrates Dialogflow NLP, Random Forest cost prediction, and Geoapify geospatial services into a unified Streamlit application. The system collects six patient parameters through natural conversation, predicts treatment cost with $R^2 = 0.953$ across eight medical categories, retrieves real hospital data within a 25 km radius, and returns ranked recommendations with budget-fitness labels in under 1.5 seconds.

Future work will address three limitations: (i) extending the symptom vocabulary using a medical ontology; (ii) incorporating hospital-specific quality signals such as NABH accreditation status and user review scores via supplementary APIs; and (iii) adding a severity-based urgency tier that overrides the distance penalty for emergency presentations, routing high-severity cardiac and respiratory cases to the nearest equipped facility regardless of cost.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Data Science at Raghu Institute of Technology, Visakhapatnam for providing the computing infrastructure used in this work. We also acknowledge the guidance of faculty members who provided valuable feedback during system testing.

REFERENCES

[1] A. Singh and R. K. Gupta, "Hospital selection behavior in urban India: a patient-centric analysis," J. Health Manag., vol. 24, no. 3, pp. 412–428, 2022.

[2] K. Rajan, "Out-of-pocket expenditure in secondary care in Andhra Pradesh," Indian J. Public Health, vol. 66, no. 1, pp. 34–40, 2022.

[3] T. Bickmore and R. Picard, "Establishing and maintaining long-term human-computer relationships," ACM Trans. Comput.-Hum. Interact., vol. 12, no. 2, pp. 293–327, 2005.

[4] P. Bhatt and A. Bhatt, "A review of machine learning approaches for hospital ranking systems," IEEE Access, vol. 9, pp. 112345–112358, 2021.

[5] S. Sutton and B. Pincock, "An overview of clinical decision support systems," npj Digital Med., vol. 3, no. 17, pp. 1–10, 2020.

[6] F. Amato et al., "A chatbot-based clinical decision support system for scheduling appointments," Future Gener. Comput. Syst., vol. 109, pp. 519–528, 2020.

[7] L. Ni et al., "Mandy: towards a smart primary care chatbot application," in Proc. Int. Symp. Knowl. Syst. Sci., Hiroshima, 2017, pp. 38–52.

[8] J. Moran et al., "Prediction of healthcare costs using a random forest model on administrative claims data," J. Am. Med. Inform. Assoc., vol. 28, no. 5, pp. 1002–1012, 2021.

[9] E. Chen et al., "Clinical cost prediction with gradient boosting and interpretable features," J. Biomed. Inform., vol. 116, p. 103716, 2021.

[10] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in Proc. ACM SIGKDD, San Francisco, CA, 2016, pp. 785–794.