IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151123

CounselAI: A Multi-Agent Conversational Counsellor with Calibrated Cutoff Forecasting for MHT-CET Engineering Admissions

Dr. Nitin Shinde[1], Dnyaneshwari Ganpatsingh Girase[2], Dhiren Sanjay Thakur[3], Om Ujjwal Patil[4], Vivek Narendra Patil[5]

Dept. of Artificial Intelligence and Machine Learning, R. C. Patel Institute of Technology, Shirpur, Maharashtra, India[1-5]

Abstract: Every year, students who clear the Maharashtra Common Entrance Test (MHT-CET) have to choose and order their engineering college preferences for the Centralized Admission Process (CAP), usually with very little personal guidance. The college predictors available today mostly filter a list of past cutoffs, work only in English and say nothing about strategy. In this paper we present CounselAI, a chat-based counselling system for these students. A large language model (LLM), coordinated through LangGraph, works with six agents that handle the student profile, eligibility, ranking, CAP-round strategy, document checklists and the final explanation. The LLM never calculates any number itself: cutoffs and chances come from a statistical model, and a grounding check rejects any reply that contains a number the tools did not produce. We built a dataset of 92,135 official closing-percentile records from nine Directorate of Technical Education (DTE) cutoff documents for 2022 and 2023, covering 338 colleges, 1,921 college–branch programs, 81 seat-category codes and three CAP rounds. On colleges that were kept out of training, our gradient-boosted model predicts the next year's cutoff with a mean absolute error (MAE) of 8.10 percentile points, 18.0% lower than simply reusing last year's cutoff (9.88), and the error drops to 1.53 points for programs with cutoffs in the 90–100 band. The admission chances derived from it are well calibrated (Brier score 0.058, AUC 0.976, decision accuracy 91.97%), and students labelled "Safe" were admitted 96.4% of the time. For deciding whether to accept a seat or wait for the next round, a classifier trained on 2022 and tested on 2023 reached 70.9% accuracy and 64.5% precision, compared with 66.4% and 54.7% for a simple rule based on the previous year. The system replies in English or Marathi. On 16 held-out test messages in English, Hindi-English and Marathi, Gemini read the profile correctly for 97.5% of fields, all 31 explanations it wrote passed the grounding check, and the median reply time was about 4 s. A rule-based fallback (98.7%) keeps the system working when the LLM is unavailable. CounselAI is deployed as a public web application. Our results suggest that keeping an LLM counsellor tied to a calibrated statistical model makes its advice something that can be checked rather than taken on trust.

Keywords: admission counselling, multi-agent systems, large language models, tool-grounded generation, cutoff prediction, gradient boosting, probability calibration, MHT-CET, Marathi language.

I. INTRODUCTION

Admission to undergraduate engineering in Maharashtra is handled through the Centralized Admission Process (CAP), run by the State Common Entrance Test Cell and the Directorate of Technical Education (DTE) [1]. Students are ranked by their MHT-CET percentile, and seats are allotted over several CAP rounds based on merit, the order of the preferences each student submits, and the state's seat-reservation rules. After every round the DTE publishes the closing percentile for each college, branch and seat category. A student who gets an allotment then has to decide whether to accept it or stay in the process and hope for something better in the next round.

Hundreds of thousands of students appear for the examination every year, and a typical student is eligible for far more programs than anyone can compare by hand. Our own dataset has 1,921 college–branch programs offered by 338 institutions. On top of this there are 81 different seat-category codes, which mix gender, social group, home-university status and special quotas. Cutoffs also move from one year to the next, and most of the guidance available to students is in English. Portals such as CollegeDekho and Careers360, and the various college predictors, let students filter past cutoffs, but they do not show how uncertain a prediction is, they do not explain whether waiting for a later round is worth it, and they do not talk to the student in a regional language. Private counselling helps, but it is expensive and not available to everyone.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151123

Large language models (LLMs) make it possible to offer this kind of guidance as a conversation [12]. The difficulty is that an LLM working on its own can state things that sound right but are not supported by any data, a problem usually called hallucination [18]. For admissions this is serious: a counsellor that makes up a cutoff or an admission chance can do real harm. We therefore keep language and numbers apart. The LLM understands what the student says and explains the advice, while every cutoff, probability and ranking is produced by code whose accuracy we can measure.

The main contributions of this work are:

- **A cleaned dataset** of 92,135 official CAP closing-percentile records (2022–2023, three rounds, 338 colleges), together with the steps we used to parse the source files, standardise branch names and seat-category codes, and merge duplicate entries.
- **An analysis of how cutoffs move** between CAP rounds and between years. It shows that cutoffs do not reliably fall in later rounds and that the year-to-year change depends heavily on how much demand a program has.
- **A prediction model with calibrated chances.** A gradient-boosted model predicts next year's cutoffs, and its past errors are turned into admission probabilities. Both are tested on colleges the model never saw during training.
- **CounselAI**, a working, publicly deployed chat system built on these parts. It adds CAP-round strategy, category-wise document checklists, replies in English and Marathi, and a grounding check that stops the LLM from showing any number the tools did not calculate.

The rest of the paper is arranged as follows. Section II covers related work and Section III describes the CAP process and our dataset. Section IV explains the system design and Section V the prediction method. Section VI gives the results, Section VII discusses what they mean and where they fall short, and Section VIII concludes.

II. RELATED WORK

A. Centralised Admissions and Matching

How to allocate seats centrally is a well-studied problem in market design. Gale and Shapley introduced the deferred-acceptance algorithm for college admissions [2], and Abdulkadiroğlu and Sönmez studied school-choice mechanisms that use priorities and preferences [3]. Closer to our setting, Baswana *et al.* describe how the joint seat allocation for India's national engineering institutes was designed and run, including reservation categories and multiple rounds [4]. These papers are about the allocation system itself. Our focus is the other side of the problem: the individual student who has to choose and order preferences without knowing where the cutoffs will land.

B. College Predictors and Recommender Systems

Recommender systems rank items using what a user prefers and what each item offers [5], [6]. A college predictor is a special kind of recommender, because the most important attribute, the chance of getting admission, is not known for certain. Most public predictors simply compare a student's score with last year's closing value, which is the persistence baseline we test in Section VI. Gradient-boosted decision trees [7], [8] work well on tabular data of this kind, and when probabilities are shown to users they need to be calibrated [10], [11]. As far as we know, no published MHT-CET predictor reports calibrated admission probabilities tested on colleges held out from training.

C. LLM Agents and Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) reduces unsupported answers by giving the LLM relevant documents to work from [13], [14]. The ReAct approach mixes reasoning steps with tool calls [15], and multi-agent frameworks such as AutoGen split a task between cooperating agents [16]; recent surveys describe the wider range of LLM-based agents [17]. CounselAI builds on these ideas but gives the LLM a narrow role: it understands the student and explains the results, and all calculations are left to tools. The agents run as a stateful graph in LangGraph [24], and the language model is from the Gemini Flash family [23]. Rather than retrieving documents, we tie the LLM to the output of our own tools and then check its text against that output.

D. Educational Chatbots and Indian-Language Technology

Chatbots are already common in education, mostly for answering questions about institutions and supporting students [19], [20], but most of them are rule-based or FAQ-based and work only in English. Recent progress in Indian-language NLP, such as IndicNLPSuite [22] and the IndicTrans2 models for all 22 scheduled Indian languages [21], makes it realistic to give guidance in Marathi, which is the first language of most students in the state.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151123

III. ADMISSION PROCESS AND DATASET

A. The CAP Process and Seat Categories

In each CAP round, a student submits an ordered list of program preferences and is allotted the highest preference for which their merit is enough within a seat category they are eligible for. The closing percentile published for a (college, branch, category) combination is the percentile of the last student admitted to it in that round. Each category code is made of four parts: a gender prefix (G for seats open to all, L for seats reserved for women); a social group (OPEN, OBC, SC, ST, VJ, NT1–NT3, and EWS for economically weaker sections); a seat-level suffix (H for Home-University seats, O for Other-than-Home-University seats, S for state-level seats); and special quotas for defence (DEF) wards, persons with disabilities (PWD), tuition-fee waiver (TFWS), minority (MI) and orphan candidates. Table II shows how the dataset is spread across these parts.

B. Data Collection and Parsing

Our data come from the official CAP cutoff documents that the DTE publishes as PDF files [1]. We parsed nine documents, covering CAP Rounds 1–3 of the 2022 and 2023 admission years, using a table-extraction script based on pdfplumber. Every record holds the year, the CAP round, the four-digit DTE college code, the college name, the branch name, the category code and the closing percentile. The parser produced 92,135 records, 42,904 for 2022 and 49,231 for 2023 (Table I). The first digit of a college code tells us the DTE region (Amravati, Chhatrapati Sambhajinagar, Mumbai, Nagpur, Nashik or Pune), and we use it as a regional feature.

C. Cleaning and Consolidation

The same branch is often spelled differently in different documents, for example “Artificial Intelligence and Data Science” and “Artificial Intelligence (AI) and Data Science”. We convert names to lower case, expand short forms such as “Engg” and remove punctuation, which brings 105 raw names down to 102. Each name is then placed in one of eight branch families (CS/IT, AI/DS, ECE, EE, ME, CE, Chemical/Technology, Other), and each category code is split into the four parts described in Section III-A. We call the combination (college, normalised branch, category) a *key*, within a given year and round. Fifteen rows were exact duplicates, and 2,747 of the 89,019 keys (3.1%) appeared more than once, because some programs are listed under more than one choice code or stage. Since the choice code is not kept in our data, we merge such entries by keeping the minimum closing percentile, that is, the lowest percentile at which a seat under that key was given. Section VI-E checks how much this choice affects the results.

TABLE I
Dataset Summary

Attribute	Value
Source documents	**9 DTE CAP cutoff PDFs**
Records (2022 / 2023)	**42,904 / 49,231**
Records by round (R1 / R2 / R3)	**45,209 / 28,924 / 18,002**
Total records	**92,135**
Colleges (2022 / 2023 / total)	**326 / 337 / 338**
College–branch programs	**1,921**
Branch names (raw / normalised)	**105 / 102**
Seat-category codes	**81**
Consolidated keys (year × round)	**89,019**
Closing percentile: median (IQR)	**61.67 (35.17–80.93)**

TABLE II
Composition by Facet (Records)

Facet	Distribution
Branch family	**CS/IT 35,477; ECE 15,092; ME 13,571; AI/DS 10,097; EE 7,463; CE 7,054; Chem/Tech 3,055; Other 326**
Region (colleges)	**Pune 39,513 (126); Mumbai 15,913 (70); Nashik 12,773 (48); Nagpur 10,405 (42); Amravati 6,929 (23); Chh. Sambhajinagar 6,602 (29)**
Social group	**OPEN 32,386; OBC 16,177; SC 11,714; EWS 6,484; NT2 5,194; TFWS 5,102; VJ 4,059; NT1 3,748; NT3 3,086; ST 2,945; MI 1,194; Orphan 46**
Seat level	**Home Univ. 36,580; Other-than-HU 25,642; State 29,913**
Gender prefix	**General (G) 62,816; Ladies (L) 29,319**

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151123

IV. SYSTEM ARCHITECTURE

Fig. 1 shows how CounselAI is built. Students use a React web page (built with Vite and Tailwind CSS) where they can ask questions in English or Marathi, and the shortlist appears as college cards with a chance bar. Behind it, a Python FastAPI service offers four endpoints: /chat for the conversation, /eligible for every eligible program with its predicted cutoff and chance, /predict for a single program, and /documents for the checklists. For /chat, LangGraph runs the six agents described below as a graph with a shared conversation state [24], so a student can answer a follow-up question without repeating what they said earlier. The cleaned cutoff table is stored in SQLite, the trained models are loaded once when the server starts, and the document requirements for each category are kept in a JSON file. The language model, gemini-3.5-flash-lite, is used only to read the profile and to write the explanation, including Marathi replies.

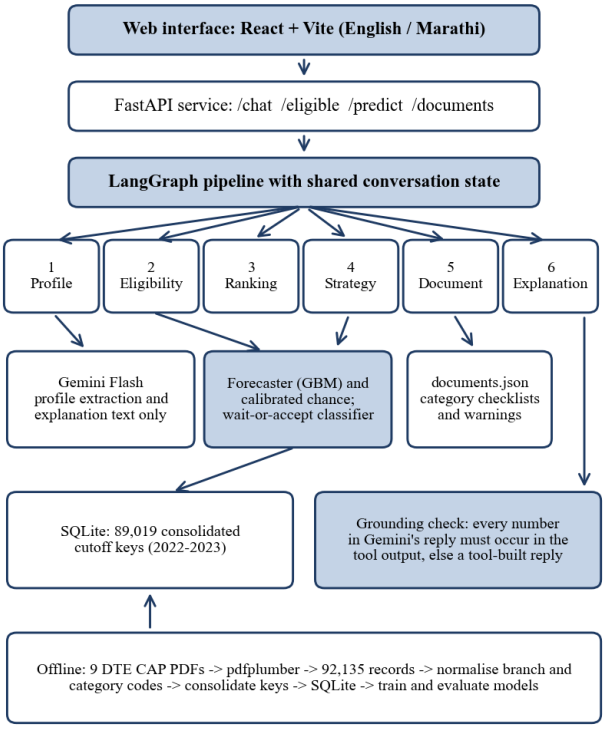


Fig. 1. Architecture of CounselAI as implemented. Gemini only extracts the profile and writes the explanation; cutoffs, chances and rankings come from deterministic tools, and a grounding check verifies every number in the reply.

A. Agents

Profile Agent. Turns a free-text message such as "87 percentile, OBC, male, Nashik, want CS or IT" into a structured profile with the percentile, category, gender, home region and preferred branches. Gemini does the extraction, and a rule-based reader that knows English, Hindi-English and Marathi words (for example मुलगी, नागपूर) acts as a backup and a cross-check. Details from earlier messages in the same conversation are kept unless the student changes them. If the percentile or the category is missing, the agent asks one short question instead of guessing.

Eligibility Agent. Works out which seat-category codes the student can compete for (for example GOBCH and GOPENH for a male OBC home-university student, plus the L codes for a female student), looks up the predicted cutoff of every matching program, and turns it into an admission probability using the method in Section V-C.

Ranking Agent. Splits the eligible programs by chance into *Reach* ($0.2 \le \pi < 0.5$), *Target* ($0.5 \le \pi < 0.8$) and *Safe* ($\pi \ge 0.8$), and sorts each group by a simple measure of program quality

$$qk = median\ \{\ ck,g1 : g \in general\ OPEN\ seats\ \} \tag{1}$$

where $c_{k,g}{}^{1}$ is the 2023 Round-1 closing percentile of program k for general (G-prefixed) OPEN seat type g. We use the median over general seats instead of the highest OPEN cutoff, because one small seat type, such as a single ladies-only

DOI: 10.17148/IJARCCE.2026.151123

seat, can otherwise make a weak program look strong. At most four programs are shown in each group, and only one per college, so the shortlist covers several institutions. We did not collect placement, accreditation or fee data for this study; adding them to the ranking is left for future work.

Strategy Agent. For each Reach and Target program, estimates how likely it is that the cutoff will fall by more than 2 points in the next round, using the classifier from Section V-D and the program's own seat category. If this probability is above 0.5, the program is marked as worth waiting for; otherwise the student is advised to freeze a seat they are happy with.

Document Agent. Gives the checklist of documents for the student's category (for example caste-validity and non-creamy-layer certificates for reserved categories, and domicile and nationality certificates for everyone) for OPEN, OBC, SC, ST, VJ/NT and EWS students, along with warnings about documents that take weeks to get.

Explanation Agent. Sends the results of the other agents (profile, Reach/Target/Safe lists, strategy and documents) to Gemini, which writes a short explanation in the student's chosen language. This reply is then checked as described next.

B. Grounding Check

The LLM is never asked to calculate a cutoff, a probability or a score. Once Gemini has written its explanation, we take every number in the reply (after converting any Devanagari digits) and look for it in the tool output that was given to the model; the only exception is list numbering from 1 to 9. If even one number is missing from the tool output, or if the Gemini call fails, we throw the reply away and send a reply built directly from the tool output instead. This means a student never sees a number that the prediction model did not produce, and the system still answers when the LLM service is down.

C. Deployment

We package the whole application as one Docker image. A Node.js stage builds the website, and the Python stage installs fixed versions of the libraries, rebuilds the SQLite database from the original CSV files and retrains the models during the build, so no generated files need to be stored in the code repository. A single FastAPI process serves the website at the root path and the API under /api. The Gemini API key is given to the host as a secret and is never placed in the image or the repository. The public version runs on the free tier of Render, which allows 512 MB of memory; on our development machine the server used about 280 MB at its peak while answering a chat request. Since all visitors share one LLM quota, /chat accepts at most 600 characters per message and at most six requests per minute and 100 per day from one visitor, and it remembers at most 5,000 conversations. The live system can be accessed at https://counselai-awyv.onrender.com.

V. FORECASTING AND CHANCE ESTIMATION

A. Problem Formulation

Let $c_{k,r}^{t}$ be the closing percentile of key k in CAP round r of year t. When a student is choosing colleges for year t, only earlier years are known, so the task is to estimate $c_{k,r}^{t}$ from information available before year t. We use all 29,928 (key, round) pairs that appear in both 2022 and 2023, from 324 colleges (16,723 in Round 1, 8,837 in Round 2 and 4,368 in Round 3), and predict the 2023 values from the 2022 data.

B. Features and Models

For each pair we use the following features: last year's cutoff for the same key and round; last year's Round-1 cutoff for the key; a demand measure, defined as the highest OPEN-category cutoff of the same program last year; the CAP round; and six categorical attributes (branch family, region, gender prefix, social group, seat level and special quota). We compare four models. **Persistence** simply predicts last year's value, which is what most public predictors do. **Persistence+shift** adds the median year-on-year change for each round, estimated on the training folds. **Ridge** regression [26] uses one-hot encoded categorical features. **GBM** is the histogram-based gradient-boosted tree model from scikit-learn [7], [8], [9], with 400 iterations, a learning rate of 0.05, at most 31 leaves, at least 40 samples per leaf, L2 regularisation of 1.0 and native handling of categorical features. All predictions are clipped to [0, 100].

C. Calibrated Admission Probability

A single predicted cutoff does not tell a student how much to trust it. We therefore look at the prediction error $e = c - \hat{c}$ and estimate its distribution separately for six bands b of the predicted cutoff (with edges 0, 25, 50, 70, 85, 95, 100),

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151123

because errors are much larger for programs with low demand. A student with percentile p is admitted under key k if $p \geq c$, so the admission probability is

$$\pi(p) = Pr(e \leq p - \hat{c}) = (1/|Rb|) \Sigma e' \in Rb \; 1[e' \leq p - \hat{c}] \quad (2)$$

where R_b is the set of errors (residuals) in band b. To keep the estimates honest, the residuals used for a test fold come only from predictions made on the other folds. The probability is then shown to the student as one of four labels: *Safe* ($\pi \geq 0.8$), *Moderate* (0.5–0.8), *Reach* (0.2–0.5) and *Unlikely* (< 0.2).

D. CAP-Round Strategy Model

Whether a student should accept a seat or wait depends on whether the cutoff of a preferred program is likely to drop noticeably in the next round. We label a key as $y = 1$ if $c_{k,1} - c_{k,2} > 2$ percentile points within the same year. A gradient-boosted classifier is trained on the 11,925 keys that appear in both Round 1 and Round 2 of 2022, and tested on the 14,096 such keys of 2023, so the test year comes strictly after the training year. Its features are the Round-1 cutoff, the demand measure, the gap between the two, and the six categorical attributes. As a baseline we use the rule from our original project design: predict a drop if the same key dropped by more than 2 points in the previous year.

E. Evaluation Protocol and Metrics

We test the prediction models with five-fold cross-validation grouped by college, so every test college is one the model has not seen. This matches the real situation of a college with little history. We report the MAE, the root-mean-square error (RMSE), the coefficient of determination R^2, and the share of predictions that fall within ±2 and ±5 points. Confidence intervals come from a cluster bootstrap over colleges with 1,000 resamples [25]. To test the admission decisions, we simulate 20 applicants for every test key and count an applicant as admitted if their percentile is at least the actual 2023 closing percentile. In *Protocol A* the applicant percentiles are spread uniformly over [0, 100]; in the harder *Protocol B* they are spread uniformly within ±10 points of the actual cutoff, where decisions are the most difficult. Each protocol gives 598,560 decisions. We judge the probabilities with the Brier score [10], the area under the ROC curve (AUC) and reliability diagrams [11].

VI. RESULTS

A. Cutoff Dynamics Across Rounds and Years

Table III shows how closing percentiles change from one round to the next for keys present in both rounds. Between Round 1 and Round 2 the median change is small and negative (−0.43 in 2022 and −0.74 in 2023). Cutoffs fall for 58.0% and 62.3% of keys, but only 35.8% and 39.3% fall by more than 2 points, and around 40% actually rise. Between Round 2 and Round 3 the picture flips: the median change is positive (+1.39 and +0.10), and more keys rise than fall. Fig. 2 shows that most changes are close to zero, with long tails on both sides. In other words, waiting for a later round does not always help, and advice about waiting has to be specific to each program and category.

TABLE III
Change in Closing Percentile Between Rounds

Year	Transition	Keys	Median	% fall	% fall >2	% rise
2022	R1→R2	11,925	−0.43	58.0	35.8	40.3
2022	R2→R3	6,290	+1.39	39.4	22.1	58.7
2023	R1→R2	14,096	−0.74	62.3	39.3	36.4
2023	R2→R3	7,769	+0.10	46.4	28.3	51.2

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151123

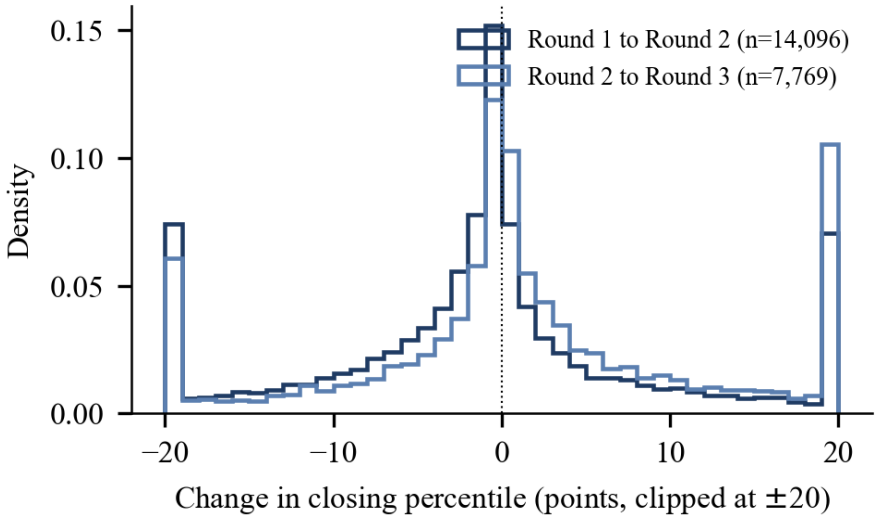


Fig. 2. Distribution of the change in closing percentile for the same key between consecutive CAP rounds in 2023. Values beyond ±20 points are accumulated in the end bins.

From one year to the next, programs keep roughly the same order but not the same values. For Round 1, the Pearson and Spearman correlations between the 2022 and 2023 cutoffs of the same key are 0.867 and 0.915 (16,723 keys), yet the average absolute change is 9.96 points and only 46.9% of keys moved by 5 points or less. The median key went up by 3.02 points. Most of this rise was in the core branches with lower demand: for general OPEN seats, the median Round-1 cutoff went up by 9.2 points in Civil, 8.9 in Mechanical and 8.7 in Electrical Engineering, against 3.1 in AI/DS and 1.4 in CS/IT. As a result, a predictor that just repeats last year's value keeps predicting cutoffs that are too low; on our test set its average signed error is −4.06 points.

B. Cutoff Forecasting

Table IV compares the models on colleges that were not used in training. The GBM gives the lowest error on every measure, with an MAE of 8.10 points, an RMSE of 12.48 and an R^2 of 0.784, against 9.88, 15.33 and 0.674 for persistence. The 95% cluster-bootstrap interval of the MAE is [7.75, 8.47] for the GBM and [9.49, 10.28] for persistence, and the paired improvement of 1.78 points has an interval of [1.63, 1.93], so the gain is not a matter of chance. The average signed error of the GBM is −0.21 points, which means it removes almost all of the downward bias seen in persistence.

TABLE IV
Cutoff Forecasting on Unseen Colleges (2023)

Model	MAE	RMSE	R^2	≤±2 (%)	≤±5 (%)
Persistence	**9.877**	**15.328**	**0.674**	**26.2**	**48.0**
Persistence+shift	**9.298**	**14.828**	**0.695**	**28.7**	**52.0**
Ridge	**8.332**	**12.754**	**0.774**	**25.6**	**52.0**
GBM	**8.097**	**12.480**	**0.784**	**31.5**	**53.0**

Five-fold cross-validation grouped by college; n = 29,928 keys from 324 colleges. Errors in percentile points.

How accurate the prediction is depends mostly on how much demand a program has (Fig. 3). For keys whose actual cutoff is in the 90–100 band, the GBM MAE is 1.53 points (persistence: 2.08). In the 75–90 band it is 4.65 (5.99), and it grows to 8.89 in the 50–75 band and 13.85 below 50. Low cutoffs are naturally unstable, since only a few seats and a thin spread of students decide the closing value. Among branch families, the MAE is lowest for AI/DS (6.23) and CS/IT (6.33) and highest for Mechanical (12.01) and Civil (11.94). The GBM does better than persistence in every branch family, every region and every social group; the largest remaining errors are for the small minority (15.82) and ST (13.55) groups. Permutation importance puts last year's Round-1 cutoff first (the MAE rises by 7.86 points when it is shuffled), followed by last year's same-round cutoff (3.73) and the demand measure (2.89). Each categorical attribute adds less than 0.5 point.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151123

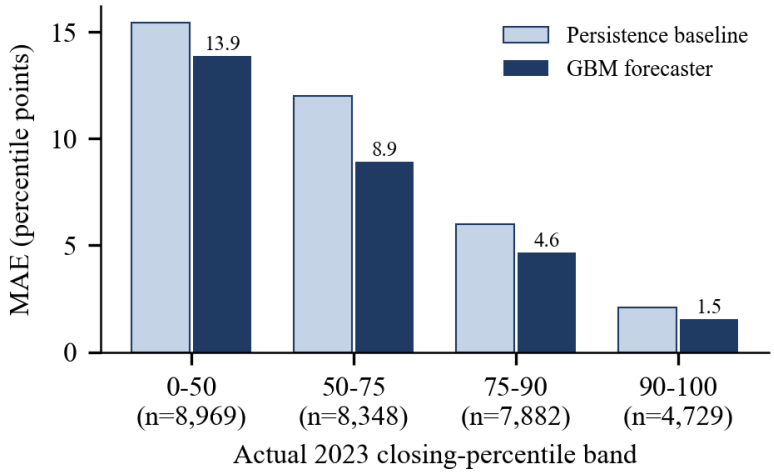


Fig. 3. Forecast MAE by the actual 2023 closing-percentile band. Highly competitive programs are predicted to within about 1.5 points.

C. Calibrated Admission Probability

Under Protocol A, the probabilities based on the GBM reach a Brier score of 0.058, compared with 0.234 for a constant guess and 0.064 when the persistence prediction is used instead, and an AUC of 0.976. Using 0.5 as the cut-off gives 91.97% accuracy, 90.44% precision, 87.77% recall and an F1 score of 89.08%. In the reliability diagram (Fig. 4) the points lie on the diagonal: in every probability decile, the actual admission rate is within 1 percentage point of the average predicted probability. The 80% prediction intervals built from the same residuals contain the actual cutoff in 79.8% of cases. Table V shows that the labels mean what they say. Applicants labelled *Safe* were admitted 96.4% of the time, and those labelled *Unlikely* only 2.9% of the time.

TABLE V
Admission Labels Shown to Students (Protocol A, GBM)

Label	Share (%)	Mean predicted (%)	Observed admit (%)
Safe ($\pi \geq 0.8$)	**29.2**	**96.5**	**96.4**
Moderate (0.5–0.8)	**7.1**	**65.7**	**65.7**
Reach (0.2–0.5)	**8.7**	**33.7**	**34.1**
Unlikely (< 0.2)	**55.0**	**2.8**	**2.9**

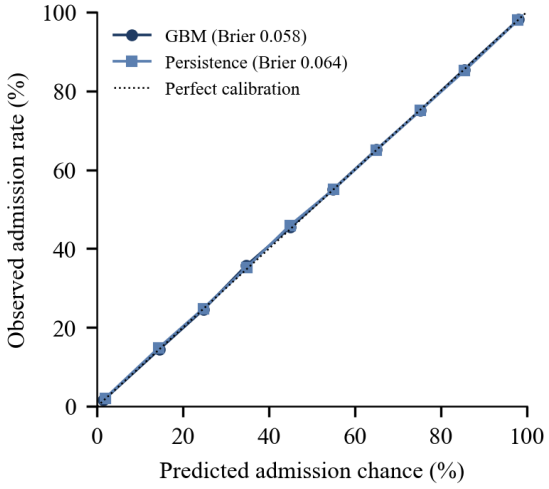


Fig. 4. Reliability diagram of admission probabilities on unseen colleges (Protocol A). Points show mean predicted chance against observed admission rate per decile.

DOI: 10.17148/IJARCCE.2026.151123

Protocol B only uses applicants within ±10 points of the true cutoff, which is hard for any predictor. Here the GBM reaches 74.09% accuracy and 72.66% precision, against 71.65% and 67.23% for persistence (F1 of 74.84% and 74.82%). Persistence has a higher recall (84.34% against 77.15%) only because it predicts cutoffs that are too low and so marks more applicants as eligible. That also means more wrong "eligible" messages, which is the more harmful mistake, since a student may then leave out safer options.

D. CAP-Round Strategy

Table VI gives the results of the accept-or-wait classifier on the 2023 test year. The previous-year rule can only be used for the 59.5% of keys that also had Rounds 1 and 2 in 2022. On these keys the classifier raises accuracy from 66.43% to 70.91% and precision from 54.73% to 64.51%, while always advising students not to wait would give 63.25%; the F1 scores are close (53.92% and 52.32%). On all 14,096 keys the classifier reaches 69.00% accuracy and an AUC of 0.757. Precision matters most here, because wrongly telling a student to wait can cost them a seat they already had. For the related task of predicting the Round-2 cutoff from Round 1, the GBM brings the MAE down to 7.89 points, compared with 8.68 when we assume no change.

TABLE VI
Accept-or-Wait Prediction (Train 2022, Test 2023)

Method	Keys	Acc.	Prec.	Rec.	F1	AUC
Majority (never wait)	**8,394**	**63.25**	**–**	**0.00**	**0.00**	**–**
Previous-year rule	**8,394**	**66.43**	**54.73**	**50.11**	**52.32**	**–**
GBM classifier	**8,394**	**70.91**	**64.51**	**46.32**	**53.92**	**0.774**
GBM classifier (all)	**14,096**	**69.00**	**64.83**	**46.28**	**54.01**	**0.757**

Target: Round-2 cutoff more than 2 points below Round 1. The first three rows use keys with previous-year history.

E. Sensitivity to Data Consolidation

If we remove every key that had more than one source entry (Section III-C), 26,996 pairs remain. On this subset the MAE is 7.35 for the GBM and 9.12 for persistence, so the conclusions stay the same and the merging step does not make the results look better than they are. The prediction model as rebuilt inside the deployed system gives the same results (MAE 8.06 for the GBM and 9.88 for persistence under the same protocol; Brier score 0.058).

F. System-Level Checks

To test the chat as a whole, we wrote 40 messages in English, Hindi-English and Marathi with known profiles, including some that leave out the percentile or the category. We used 24 of them during development to find and fix mistakes. One of these was serious: a reservation sub-group such as NT2 was being read as the wider VJ/NT group, which would have counted seats the student is not eligible for. The other 16 were written afterwards and were not used to tune anything. Every message went through the full pipeline twice, once with Gemini (gemini-3.5-flash-lite, with requests spaced out to stay within the free-tier rate limit) and once with the rule-based reader alone. Table VII shows the results.

TABLE VII
End-to-End Checks on 40 Test Messages

Path / set	Fields correct	Asks when needed	Grounded replies	Median time
Gemini, development	**116/117 (99.1%)**	**24/24**	**18/18**	**4.8 s**
Gemini, held-out	**77/79 (97.5%)**	**15/16**	**13/13**	**4.1 s**
Rule-based, development	**117/117 (100%)**	**24/24**	**–**	**<0.1 s**
Rule-based, held-out	**78/79 (98.7%)**	**16/16**	**–**	**<0.1 s**

Fields: percentile, category, gender (scored only when stated), region and branch families. Grounded replies: Gemini explanations whose every number occurs in the tool output. Rule-based replies are built from the tool output and need no check. Development results are after fixes made on that set.

All 31 explanations written by Gemini passed the grounding check, so no reply had to be replaced in this test. The check itself is covered by an automated test in which a fake LLM adds an invented cutoff to its reply. On the held-out messages Gemini made two kinds of mistake. It returned a branch as a phrase ("AI and ML") that our branch mapper did not split, and, more seriously, it filled in "OPEN" for a student who had not given a category, so the system answered when it should have asked. Since a guessed category shows the student the wrong seats, the Profile Agent now accepts a category from the LLM only if the message itself names one, and branch phrases are split before mapping. Both fixes are covered

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151123

by tests, but we have not yet measured them on new messages. We chose the lighter model because, on the free API tier, the larger gemini-3.8-flash model allowed only 20 requests per day, which is about ten conversations. In addition, 28 automated tests cover the data pipeline, the models, the API and the agents.

VII. DISCUSSION

A. Implications for Counselling

A few of our findings matter directly for students. First, cutoffs change a lot from year to year, and in 2023 they rose most for the core branches with lower demand, so a student who relies on last year's list will tend to overrate their chances. Second, the uncertainty is very uneven: for competitive programs the prediction is accurate to about 1.5 points, but for low-demand programs it can be off by more than ten. Showing a calibrated probability, rather than a plain "eligible" or "not eligible", makes this clear to the student. Third, later rounds do not always bring cutoffs down, so the decision to wait should depend on the specific program and not on a general rule.

B. Limitations

Our study has some limitations. (i) We only had two admission years, so the model learns from a single year-to-year change; with 2024 and later data we could add multi-year features and test on a truly future year. (ii) We treat admission as percentile ≥ closing percentile, which ignores the student's own preference order and tie-breaking, so the probability is the chance of being above the cutoff, not of getting a particular seat. (iii) Merging duplicate entries by the minimum is a cautious choice that we had to make because the parsed data do not include choice codes. (iv) The chat was tested on 40 messages that we wrote ourselves, and students have not yet judged how clear and useful the English and Marathi explanations are in a formal user study. (v) The ranking uses a quality measure based on cutoffs, because we did not collect placement, accreditation or fee data. (vi) The public version runs on a free hosting tier that puts the service to sleep when it is not used, so the first request after a quiet period can take about a minute; it also has no user accounts, and conversations are kept in the server's memory and are lost when it restarts.

C. Ethical Considerations

All the data we used are public, aggregate cutoffs and contain no personal information. The system shows probabilities with clear labels and tells students that its advice is not a guarantee of admission. Because the prediction error is different for different social groups, mainly because the number of seats per category differs, the system shows uncertainty for each category instead of one overall confidence figure. The final decision always stays with the student and their family.

VIII. CONCLUSION AND FUTURE WORK

We have presented CounselAI, a chat-based counsellor for MHT-CET engineering admissions in which the LLM handles the conversation and a tested statistical model handles every number. On 92,135 official cutoff records, our gradient-boosted model cuts the prediction error by 18.0% compared with reusing last year's cutoff on unseen colleges, and the admission chances based on it are well calibrated, with 96.4% of "Safe" recommendations turning out right. A classifier tested on a later year than it was trained on improves the precision of accept-or-wait advice by about ten percentage points over a simple rule. In future we plan to add the 2024 and 2025 cutoffs; include placement, NAAC and fee data in the ranking; answer descriptive questions about hostels and placements using retrieval; add user accounts and a production database; build a post-admission guidance agent; compare Gemini's Marathi with IndicTrans2 [21]; run a user study with MHT-CET students; and support direct second-year (DSE) admissions.

DATA AND CODE AVAILABILITY

The cutoff data come from public DTE documents [1]. The cleaned dataset, the scripts that reproduce every number, table and figure in this paper (analysis, chance estimation and figures, with a fixed random seed), and the source code of the CounselAI application with its 28 automated tests are available from the authors on request. A live version of the system is available at https://counselai-awyv.onrender.com.

ACKNOWLEDGMENT

This work was carried out as a final-year project under the guidance of Dr. Nitin Shinde, Department of Artificial Intelligence and Machine Learning, R. C. Patel Institute of Technology, Shirpur. An AI writing assistant was used to help draft and format portions of this manuscript; the authors reviewed and verified all content and take full responsibility for it.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151123

REFERENCES

[1] State Common Entrance Test Cell and Directorate of Technical Education, Government of Maharashtra, "CAP cut-off lists for first-year engineering admissions, 2022–23 and 2023–24." [Online]. Available: https://cetcell.mahacet.org and https://dte.maharashtra.gov.in
[2] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Monthly*, vol. 69, no. 1, pp. 9–15, 1962.
[3] A. Abdulkadiroğlu and T. Sönmez, "School choice: A mechanism design approach," *Amer. Econ. Rev.*, vol. 93, no. 3, pp. 729–747, 2003.
[4] S. Baswana, P. P. Chakrabarti, S. Chandran, Y. Kanoria, and U. Patange, "Centralized admissions for engineering colleges in India," *INFORMS J. Appl. Anal.*, vol. 49, no. 5, pp. 338–354, 2019.
[5] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.
[6] F. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*, 3rd ed. New York, NY, USA: Springer, 2022.
[7] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.
[8] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 3146–3154.
[9] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[10] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Rev.*, vol. 78, no. 1, pp. 1–3, 1950.
[11] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1321–1330.
[12] T. B. Brown *et al.*, "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.
[13] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 9459–9474.
[14] Y. Gao *et al.*, "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2023.
[15] S. Yao *et al.*, "ReAct: Synergizing reasoning and acting in language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023.
[16] Q. Wu *et al.*, "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," arXiv:2308.08155, 2023.
[17] L. Wang *et al.*, "A survey on large language model based autonomous agents," *Front. Comput. Sci.*, vol. 18, no. 6, Art. no. 186345, 2024.
[18] Z. Ji *et al.*, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, Art. no. 248, 2023.
[19] C. W. Okonkwo and A. Ade-Ibijola, "Chatbots applications in education: A systematic review," *Comput. Educ.: Artif. Intell.*, vol. 2, Art. no. 100033, 2021.
[20] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, "Chatbot for university related FAQs," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, 2017, pp. 1525–1530.
[21] J. Gala *et al.*, "IndicTrans2: Towards high-quality and accessible machine translation models for all 22 scheduled Indian languages," *Trans. Mach. Learn. Res.*, 2023.
[22] D. Kakwani *et al.*, "IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," in *Findings Assoc. Comput. Linguist.: EMNLP*, 2020, pp. 4948–4961.
[23] Gemini Team, Google, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," arXiv:2403.05530, 2024.
[24] LangChain, Inc., "LangGraph: Build resilient language agents as graphs," 2024. [Online]. Available: https://github.com/langchain-ai/langgraph
[25] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York, NY, USA: Chapman & Hall, 1993.
[26] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

 This work is licensed under a Creative Commons Attribution 4.0 International License