

An AI-Driven University Recommendation System for Nepalese Study-Abroad Aspirants with Peer Consultation Support

Nirmal Khadka*

Department of Software Engineering
Nepal College of Information Technology
(Pokhara University),
Lalitpur, Nepal
nirmal.201719@gmail.com

Arjun Raut

Department of Software Engineering
Nepal College of Information Technology
(Pokhara University),
Lalitpur, Nepal
arjun.201704@ncit.edu.np

Deepu Shah

Department of Software Engineering
Nepal College of Information Technology
(Pokhara University),
Lalitpur, Nepal
deepu.201707@ncit.edu.np

Pushpa Raj Sangroula

Department of Software Engineering
Nepal College of Information Technology
(Pokhara University),
Lalitpur, Nepal
pushpa.201726@ncit.edu.np

*Corresponding author

Abstract— Every year, over 100,000 Nepalese students pursue higher education abroad, yet university selection remains manual, fragmented, and heavily reliant on commercial consultancies with potential partner-institution bias. This paper presents UNIFINDER, a web platform integrating machine-learning-based university matching with peer-to-peer consultation. Built on the MERN stack with a Python/Flask microservice, the system hosts two regression models trained on scraped data: an XGBoost Regressor for general international applicants, and a second XGBoost Regressor combined with K-Means clustering for Nepalese applicants, ranking institutions by Nepalese student concentration. Data collection used an automated Puppeteer.js and Cheerio.js pipeline; a Zoom API integration (via Ngrok) enabled video consultations with peers abroad. An initial evaluation on a single random split suggested strong accuracy ($R^2 = 0.93$ and 0.81). However, an audit revealed institution-level data leakage. Under corrected, university-grouped cross-validation, both models showed weaker generalization to unseen institutions (mean R^2 of 0.18 and 0.31 across five folds), indicating dataset size limitations (131 records across ≈ 72 institutions). We report this transparently, identifying dataset expansion as key future work. Combining data-driven matching with peer guidance aims to reduce reliance on third-party consultancies, though this benefit remains to be directly evaluated.

Keywords—university recommendation system, XGBoost, gradient boosting, web scraping, MERN stack, peer mentorship, study abroad, machine learning

I. INTRODUCTION

Every year, a growing number of Nepalese students pursue higher education in countries such as Australia, the United States, the United Kingdom, and Canada. Ministry of Education, Science and Technology (MoEST) data indicates that more than 110,000 students received No Objection Certificates (NOCs) for overseas study in fiscal year 2022/23 alone [1], reflecting a steep and continuing rise in international

academic migration. Despite this scale, the process of selecting a university that matches a student's academic profile, budget, and career goals remains manual and fragmented: students must browse multiple websites, reconcile inconsistent admission requirements, and estimate living costs from unverified sources.

Two structural gaps compound this problem. First, centralized and trustworthy information is scarce, so applicants must piece together requirements from many sources. Second, students rely heavily on education consultancies, some of which may be compensated by partner institutions and therefore have an incentive to steer applicants toward specific universities; press reports have documented Nepalese students placed in unaccredited institutions through consultancies [2], [3]. A related, less-discussed problem is post-admission planning: students without prior contacts abroad must make housing and budgeting decisions with limited reliable information. This paper presents UNIFINDER, a platform designed to address these three gaps — matching, trust, and post-admission cost planning — within a single system rather than as separate tools. The specific contributions of this work are:

- A dual machine-learning recommendation engine that separately models eligibility ranking for international applicants and popularity-among-Nepalese-peers for Nepalese applicants, rather than a single one-size-fits-all model.
- An automated, resilient data-collection pipeline that scrapes live university and course data using Puppeteer.js and Cheerio.js, with anti-blocking measures to sustain long-running collection.
- A peer video-consultation module, integrated through the Zoom API and webhooks, that lets applicants speak

directly with Nepalese students already enrolled at a recommended institution.

- An empirical evaluation of the recommendation models using university-grouped cross-validation to avoid institution-level data leakage, together with a comparison against a rule-based baseline and an intra-list diversity analysis, providing an honest assessment of what the current models add beyond simple eligibility filtering.

II. RELATED WORK

Prior work relevant to this problem falls into three strands. The first documents the problem context: press reporting describes Nepalese students relying heavily on education consultancies whose advice is not always reliable, including cases of students placed in unaccredited institutions [2], [3]. The second concerns recommendation methods: hybrid recommender systems that combine content-based and collaborative signals can outperform single-technique approaches [4], [7], which motivates our use of tree-based regressors alongside rule-based filtering; purely rule-based recommenders remain easy to interpret but scale poorly to frequently changing admission data. The third concerns evaluation: when observations are grouped, as courses are within universities, random train/test splits can leak group-level information and overstate performance, and grouped cross-validation is the standard remedy [5], [6]; we use GroupKFold as implemented in scikit-learn [8].

These strands remain largely disconnected in existing commercial systems. The IDP Education Portal [9] connects students to human agents and offers basic filtering, but recommendations are agent-mediated rather than data-driven and the platform has no Nepal-specific context. ApplyBoard [10] applies machine learning to matching but its recommendations skew toward partner institutions and it offers no real-time cost estimation or direct contact with current students. Yocket, a community-oriented platform, allows students to view admission profiles and discuss applications but is primarily built around Indian applicants and lacks scheduled mentorship or financial-planning tools. Table I summarizes these gaps.

TABLE I. COMPARISON OF UNIFINDER AGAINST EXISTING STUDY-ABROAD PLATFORMS

Feature	IDP Education [9]	Apply Board [10]	Yocket	UNI-FINDER
AI-based personalized matching	No	Partial	Partial	Yes
Nepal-specific data / context	No	No	No	Yes
Scheduled peer video consultation	No	No	No	Yes
Real-time cost estimation	No	No	No	Yes
Designed to avoid partner-institution bias*	No	No	Yes	By design

*This row reflects a design goal rather than an experimentally validated outcome; no user study or comparison against consultancy recommendations has yet been conducted.

UNIFINDER is, to our knowledge, the first system to combine a Nepal-specific dual recommendation model, automated real-time data collection, and scheduled peer video consultation within a single platform.

III. SYSTEM DESIGN AND METHODOLOGY

A. Development Process

The system was built using an incremental development model across three increments, each following a requirement-analysis → design → coding → testing cycle. The first increment delivered the core product — authentication and AI-based university matching; the second added the peer-to-peer consultation workflow; the third added tuition and living-cost estimation. This staged approach allowed the AI recommendation engine, which carried the highest technical risk, to be validated early before dependent features were built on top of it.

B. System Architecture

UNIFINDER follows a service-oriented architecture (Fig. 1). A React.js single-page frontend communicates with a Node.js/Express.js backend over a RESTful API. The backend delegates recommendation requests to a Python/Flask microservice that hosts the trained regression models and connects to MongoDB Atlas via PyMongo to read web-scraped university records. The same backend separately manages Zoom meeting scheduling and webhook events for the peer-consultation feature. Separating the ML service from the main backend allowed the data science and web development workstreams to be developed and iterated independently.

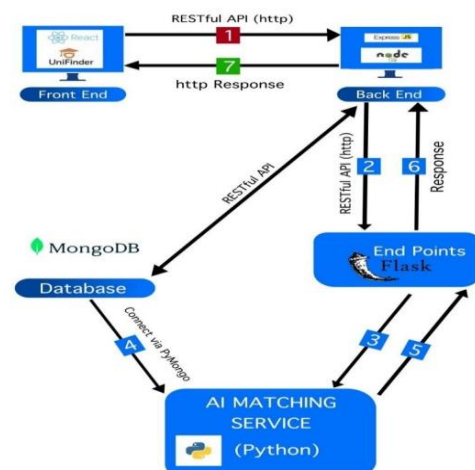


Fig. 1. High-level system architecture of UNIFINDER.

C. Data Collection via Web Scraping

University and course data were collected from a trusted study-abroad portal using a three-phase scraper built on Puppeteer.js (headless-browser automation) and Cheerio.js (server-side HTML parsing). Phase 1 crawled listing pages to extract course titles, university names, rankings, locations, IELTS/TOEFL requirements, and tuition fees. Phase 2 visited individual course pages to extract descriptions, duration, and entry requirements. Phase 3 visited university-specific pages to extract institutional descriptions. To sustain long-running collection without being blocked, the scraper used the puppeteer-extra-plugin-stealth plugin to mask automation signals, randomized delays between requests to mimic human browsing, and the networkidle2 wait strategy to ensure pages were fully rendered before extraction. The scraping logic was modularized across per-source scripts, allowing both vertical scaling (more fields per page) and horizontal scaling (more source pages processed in parallel).

D. Data Preprocessing and Feature Engineering

Raw scraped records were normalized into a tabular format and cleaned before model training. Country was inferred from free-text location strings via keyword mapping; course level was simplified into Undergraduate/Postgraduate categories; university rankings and TOEFL scores, both provided as free text, were extracted with regular expressions and missing rankings were imputed with the column median. IELTS, GPA, and tuition-fee fields were coerced to numeric types and tuition was normalized to USD using fixed exchange rates. Outliers in numeric fields were identified using Z-score and interquartile-range (IQR) methods and either capped or removed. Categorical variables (country, course level) were one-hot encoded, and numeric features were scaled using min-max normalization or standardization depending on their distribution. For the Nepalese-student dataset specifically, additional fields — tuition, cost of living, total cost, and scholarship availability — required currency-symbol stripping and type conversion before use.

TABLE II. FEATURE ENGINEERING SUMMARY (MODEL 1 DATASET)

Feature	Type	Transformation Applied
IELTS score	Numeric	Standardized (z-score)
TOEFL iBT	Numeric	Regex-extracted; missing values imputed; standardized
Minimum GPA	Numeric	Standardized
Tuition fee (USD)	Numeric	Currency-normalized; standardized
University ranking (target)	Numeric	Regex-extracted; median-imputed; standardized
Country	Categorical	One-hot encoded (5 categories)
Course level	Categorical	Simplified to Undergrad./Postgrad.; one-hot encoded

Table II summarizes the transformation applied to each feature; the Nepalese-student dataset (Model 2) followed the same pipeline with additional cost-related fields. As a concrete illustration, a raw record with IELTS = 6.5, TOEFL iBT = 79, GPA = 2.8, and tuition = USD 52,410 (Australia, Bachelor's) is transformed into standardized values of approximately 0.48, 0.70, 0.37, and 0.68 respectively, together with one-hot indicators for country and course level, before being passed to the regression models.

E. AI Recommendation Models

Two separate regression models were trained because the two applicant populations optimize for different outcomes: international applicants need an eligibility ranking, while Nepalese applicants benefit more from knowing where peer communities already exist.

1) *Model 1 — International Students*: Four candidate regressors (Random Forest [11], XGBoost [12], LightGBM [13], and Gradient Boosting [14]) were tuned via RandomizedSearchCV with 10-fold cross-validation on an initial random 80/20 train/test split ($n = 131$ records). XGBoost achieved the best held-out performance and was initially selected for production. A subsequent audit identified two issues: (i) courses from the same university could appear in both training and test partitions, letting the model learn institution-specific patterns; and (ii) hyperparameter search was run on the full dataset before splitting in several scripts, leaking test information into model selection. All four algorithms were therefore re-evaluated using GroupKFold cross-validation (five outer folds, three inner folds for tuning)

grouped by university, with zero train/test institution overlap verified per fold. Random Forest achieved the highest corrected mean R^2 (Section IV), but XGBoost remains deployed in the production Flask microservice, predicting eligibility ranking from GPA, IELTS/TOEFL scores, tuition budget, country, and course level (Sections IV and V).

2) *Model 2 — Nepalese Students*: Five candidate regressors (Random Forest [11], XGBoost [12], LightGBM [13], Gradient Boosting [14], and CatBoost [15]) were tuned via GridSearchCV with 10-fold cross-validation on an initial random 80/20 split, and XGBoost was selected for production. This evaluation was later found to suffer from the same leakage as Model 1, so a corrected university-grouped cross-validation was run for Random Forest, XGBoost, and LightGBM (CatBoost was not re-evaluated; Section IV).

Rather than predicting institutional ranking, this model predicts the count of currently enrolled Nepalese students per course, since peer density is a stronger signal of community support and cultural fit for this population — a framing we make explicit, since a higher concentration of Nepalese students does not necessarily indicate a university's academic suitability or quality. Candidate universities are first filtered by country, course level, subject-keyword match, and the applicant's score/GPA/budget thresholds; scholarship eligibility is flagged where the applicant's GPA meets the listed threshold. The filtered set is then partitioned using K-Means clustering ($k = 5$) [16] over academic and cost features, the cluster with the highest average Nepalese-student count is identified, and the top five universities within that cluster — ranked by a weighted combination of the model's predicted Nepalese-student count (70%) and the historically observed Nepalese-student count (30%) — are returned to the applicant.

Both models are served through a Flask REST API that accepts an applicant's profile as JSON, applies the same scaling/encoding pipeline used in training, and returns the top five matching universities.

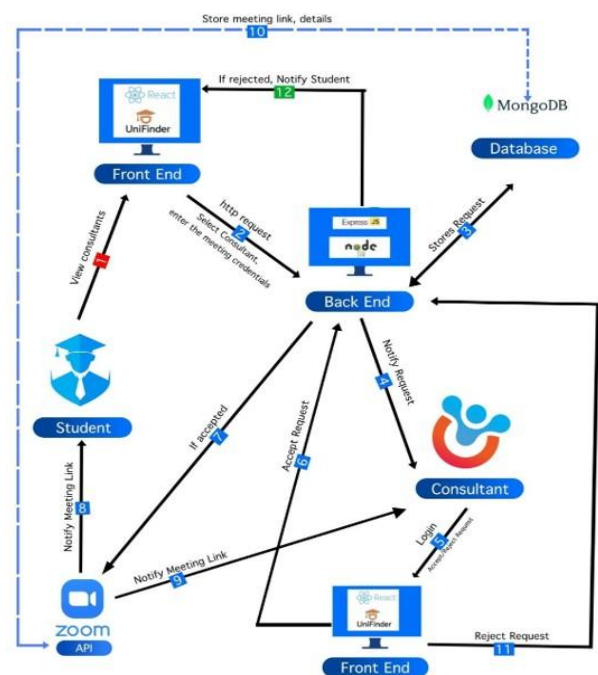


Fig. 2. Peer video-consultation workflow: request creation, consultant accept/reject, Zoom meeting-link generation, and notification of both parties.

F. Peer Consultation Module

To let applicants speak with Nepalese students already studying at a recommended university, UNIFINDER integrates the Zoom API (Fig. 2). A Zoom Marketplace app was registered under OAuth, and four webhook events — meeting.started, meeting.ended, participant.joined, and participant.left — were subscribed to for session tracking.

Because development occurred on localhost, Ngrok was used to expose a temporary public HTTPS endpoint that forwarded Zoom webhook traffic to the local Express.js server; webhook requests were validated with Zoom verification tokens to prevent spoofing. In the consultation workflow, a student selects an available consultant and proposes a meeting time; the consultant may accept or reject the request; on acceptance, the backend calls the Zoom API to generate a meeting link, which is stored and surfaced to both parties.

IV. RESULTS

Model 1 was initially evaluated on a single random 80/20 split ($n = 27$ held-out records), where XGBoost outperformed Random Forest ($R^2 = 0.93$ vs. 0.75) and was selected for production (Table III). This evaluation was later found to suffer from institution-level and search-before-split leakage (Section III-E), so Table III also reports the corrected five-fold GroupKFold cross-validation grouped by university, with tuning restricted to each training partition and zero train/test institution overlap.

Under this protocol, Random Forest achieved the highest mean R^2 (0.18 ± 0.47), followed by Gradient Boosting (0.15 ± 0.45), while XGBoost (-0.02 ± 0.73) and LightGBM (-0.07 ± 0.63) performed near or below a mean-prediction baseline. XGBoost remains deployed, but its apparent advantage did not hold once leakage was removed (Section V). Fig. 3 shows the corrected mean R^2 , RMSE, and MAE with standard-deviation error bars.

TABLE III. MODEL 1 ALGORITHM COMPARISON
ORIGINAL EVALUATION (SINGLE RANDOM 80/20 SPLIT, $N = 27$):

Algorithm	R^2	RMSE	MAE
Random Forest	0.75	0.42	0.23
XGBoost (deployed)	0.93	0.23	0.15

CORRECTED EVALUATION (5-FOLD GROUPKFOLD CROSS-VALIDATION
GROUPED BY UNIVERSITY; MEAN $\pm$ STD ACROSS FOLDS):

Algorithm	R^2	RMSE	MAE
Random Forest	0.18 ± 0.47	0.78 ± 0.23	0.49 ± 0.15
Gradient Boosting	0.15 ± 0.45	0.80 ± 0.24	0.50 ± 0.16
XGBoost (deployed)	-0.02 ± 0.73	0.85 ± 0.24	0.52 ± 0.17
LightGBM	-0.07 ± 0.63	0.89 ± 0.22	0.65 ± 0.17

The same methodology was applied to Model 2. Under the original single split ($n = 27$), XGBoost achieved the best R^2 , RMSE, and MAE (Table IV) and was selected for production. Under corrected 5-fold GroupKFold cross-validation (Random Forest, XGBoost, and LightGBM only; CatBoost was not re-evaluated), mean R^2 fell to 0.31 ± 0.62 for XGBoost, 0.25 ± 0.53 for Random Forest, and -0.30 ± 1.27 for LightGBM, with high fold variance (Fig. 4). XGBoost retains the highest mean R^2 of the three and remains deployed, though with much weaker generalization than the original estimate suggested.

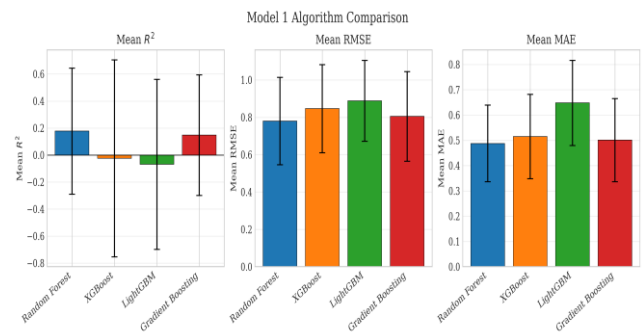


Fig. 3. Model 1 algorithm comparison under corrected 5-fold grouped cross-validation: mean R^2 , RMSE, and MAE with standard-deviation error bars across Random Forest, XGBoost, LightGBM, and Gradient Boosting.

TABLE IV. MODEL 2 ALGORITHM COMPARISON
ORIGINAL EVALUATION (SINGLE RANDOM 80/20 SPLIT, $N = 27$):

Algorithm	R^2	RMSE	MAE
Random Forest	0.795	189.83	108.14
XGBoost (deployed)	0.812	181.61	83.96
LightGBM	0.727	218.94	154.11
Gradient Boosting	0.774	199.24	102.95
CatBoost	0.789	192.40	109.20

CORRECTED EVALUATION (5-FOLD GROUPKFOLD CROSS-VALIDATION
GROUPED BY UNIVERSITY; MEAN $\pm$ STD ACROSS FOLDS; CATBOOST NOT RE-EVALUATED):

Algorithm	R^2	RMSE	MAE
XGBoost (deployed)	0.31 ± 0.62	217.71 ± 74.73	170.76 ± 50.39
Random Forest	0.25 ± 0.53	245.37 ± 89.66	193.72 ± 63.68
LightGBM	-0.30 ± 1.27	292.58 ± 66.85	226.11 ± 51.08

^a. Note: R^2 , RMSE, and MAE for Model 2 are computed on the raw enrolled-student-count target rather than the standardized ranking scale used for Model 1 (Table III), since the two models predict different quantities; RMSE/MAE values are therefore not directly comparable across Tables III and IV.

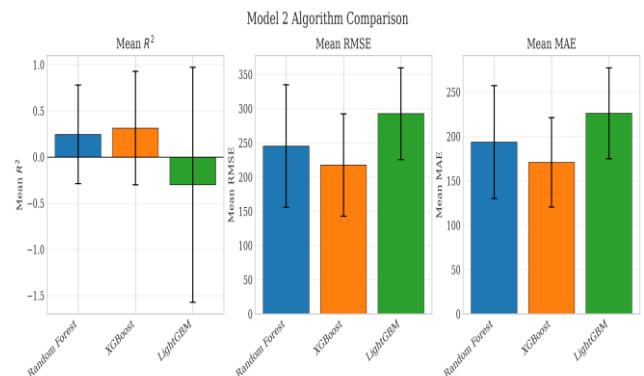


Fig. 4. Model 2 algorithm comparison under corrected 5-fold grouped cross-validation: mean R^2 , RMSE, and MAE with standard-deviation error bars across Random Forest, XGBoost, and LightGBM.

Table V summarizes both models under both protocols. The original R^2 values (0.93 and 0.81) came from a leaky single split; the corrected values (0.18 for Model 1, 0.31 for Model 2) are substantially lower, with wide fold-to-fold variance (Fig. 9, Fig. 10), indicating that with 131 records across approximately 72 institutions neither model generalizes reliably to unseen universities. Regularized linear models (Ridge, Lasso) under the same grouped protocol did not outperform the best tree-based model (Model 1: Ridge $R^2 = 0.07$, Lasso $R^2 = 0.05$ vs. Random Forest 0.18 ; Model 2: Ridge $R^2 = -0.65$, Lasso $R^2 = -0.86$ vs. XGBoost 0.31), suggesting the limiting factor is data quantity and diversity rather than model complexity.

TABLE V. FINAL TEST-SET PERFORMANCE OF THE TWO RECOMMENDATION MODELS

Model	Algorithm	Target Variable	Original R ² (single split)*	Corrected Mean R ² (grouped CV $\pm$ std)
Model 1 (Int'l students)	XGBoost Regressor (deployed)	University ranking (eligibility)	0.93	-0.02 $\pm$ 0.73
Model 1 (Int'l students)	Random Forest (highest corrected R ²)	University ranking (eligibility)	0.75	0.18 $\pm$ 0.47
Model 2 (Nepalese students)	XGBoost Regressor + K-Means (deployed)	Count of enrolled Nepalese students	0.81	0.31 $\pm$ 0.62

b. *The "Original R²" is from a single 80/20 split ($n = 27$) later found to be leaky (Section III-E); "Corrected" is 5-fold GroupKFold CV grouped by university (mean $\pm$ std). XGBoost is deployed for both tasks; Random Forest is shown for Model 1 as the best corrected model.

Data cleaning changed the correlation structure of both datasets. In the international-student data, the IELTS–tuition correlation rose from 0.35 to 0.41 and TOEFL–tuition from -0.031 to 0.18 after imputation and normalization (Fig. 5); we report this descriptively, not as proof that cleaning introduced no bias. In the Nepalese-student data, the IELTS–TOEFL correlation rose from 0.57 to 0.64, while cost of living–total cost fell from 0.73 to 0.68 (Fig. 6). Cost of living showed the strongest correlation with the Model 2 target ($r = 0.37$), followed by minimum GPA ($r = 0.34$) and IELTS ($r = 0.33$).

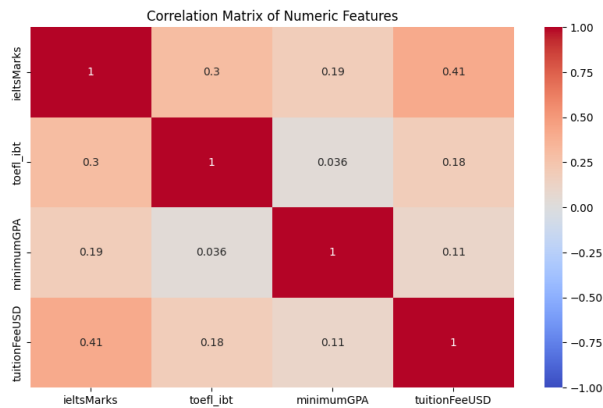


Fig. 5. Correlation matrix of numeric features for the international-student dataset (after cleaning).

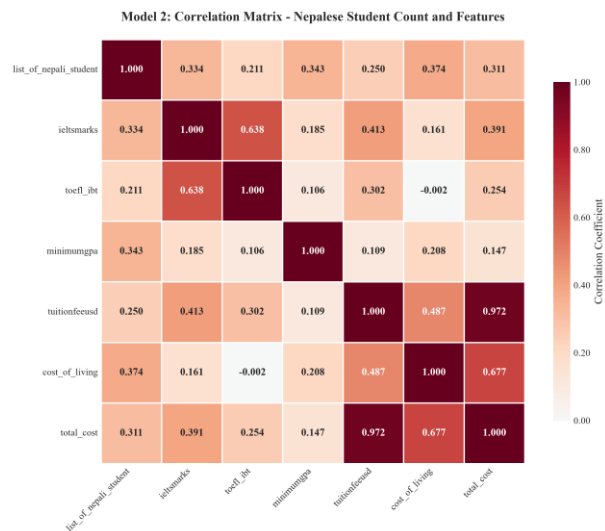


Fig. 6. Correlation matrix of numeric features for the Nepalese-student dataset (after cleaning).

Fig. 7 and Fig. 8 plot predicted against actual values on a held-out, university-grouped test fold; both show substantial scatter around the identity line, consistent with the modest corrected R² in Table V. Fig. 9 and Fig. 10 show fold-level R² per algorithm: values range from approximately +0.65 to -1.18 for Model 1's XGBoost and vary similarly for Model 2, depending on which institutions fall in the held-out fold. This instability is consistent with a training set too small to learn a stable, institution-independent pattern.

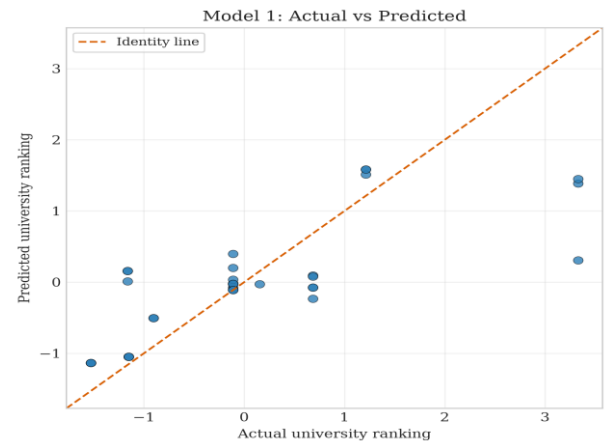


Fig. 7. Actual vs. predicted university-ranking scores on a held-out, university-grouped test fold, Model 1 (XGBoost).

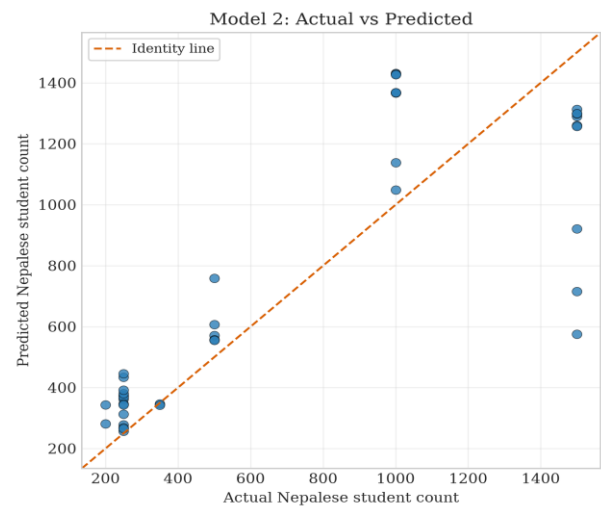
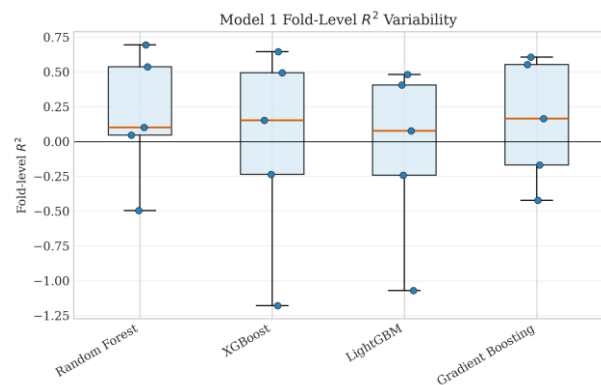


Fig. 8. Actual vs. predicted values on a held-out, university-grouped test fold, Model 2 (XGBoost).

Fig. 9. Fold-level R² across five university-grouped cross-validation folds, Model 1 (Random Forest, XGBoost, LightGBM, Gradient Boosting).

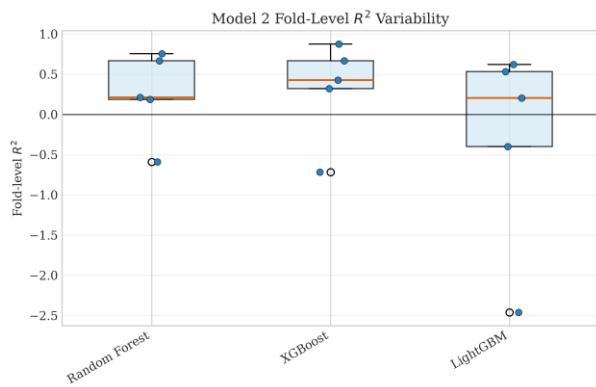


Fig. 10. Fold-level R^2 across five university-grouped cross-validation folds, Model 2 (Random Forest, XGBoost, LightGBM).

We also compared ML-ranked recommendations with a rule-based baseline using the same eligibility filters and a non-ML tiebreaker (ascending tuition for Model 1; historical Nepalese-student count for Model 2). The top-5 lists overlapped heavily (mean Jaccard 0.85 for Model 1, $n = 27$; 0.91 for Model 2, $n = 40$), with small average rank shifts (1.12 and 0.17 positions), indicating that eligibility filtering, rather than learned ranking, accounts for most of the recommendation quality. Intra-list diversity (Gower distance; only queries returning a full five-item list: 18 of 27 and 30 of 40) was lower for Model 1's ML lists than for the baseline (0.032 vs. 0.047) and comparable for Model 2 (0.036 vs. 0.036) (Fig. 11). These findings are consistent with the grouped cross-validation results: a model that has not learned a stable pattern from 131 records should not be expected to outperform simple rule-based filtering.

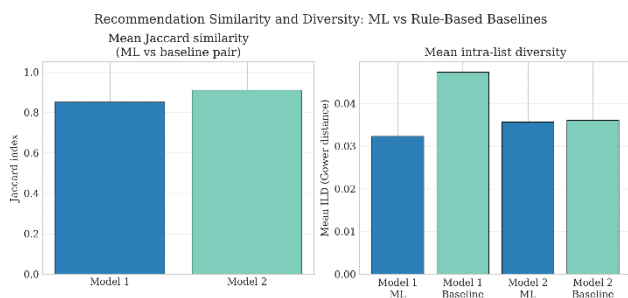


Fig. 11. Fig. 11. Mean Jaccard similarity (ML vs. rule-based baseline) and mean intra-list diversity (ML vs. baseline), Model 1 and Model 2.

Functional testing covered eight core workflows (registration/login, university matching, meeting request/accept/reject/history, password reset, admin management, and both prediction endpoints). Two initial defects, consultant accept/reject handling and the password-reset flow, were corrected and retested; all workflows now pass.

V. DISCUSSION

The corrected, university-grouped cross-validation supports more cautious conclusions than the original single split. None of the evaluated algorithms generalizes reliably to unseen universities: best corrected mean R^2 (0.18 for Model 1, 0.31 for Model 2) is far below the original estimates (0.93 and 0.81), with wide fold-to-fold variance (Fig. 9, Fig. 10). Regularized linear models ruled out excessive model complexity, pointing to dataset size and institutional diversity (131 records, approximately 72 institutions) as the binding constraint. The baseline comparison further shows that most of the recommendation value currently comes from rule-based

eligibility filtering rather than learned ranking. We report this negative result openly because it identifies dataset expansion, rather than further model tuning, as the priority. Splitting the problem into two population-specific models remains reasonable on conceptual grounds, since peer density and eligibility ranking are distinct notions of a "good recommendation," but this choice is not validated by the current accuracy results.

These findings have a practical implication for how the current system should be used. Given that most of the recommendation value derives from eligibility filtering, UNIFINDER is best understood at present as a structured filtering and peer-consultation tool, with the learned ranking serving as a secondary signal rather than an authoritative ranking. The peer video-consultation module complements this by giving applicants first-hand, experience-based information that no predictive model trained on the current dataset can supply. As the dataset grows, the contribution of the learned component can be re-evaluated using the same university-grouped protocol reported here.

Development also surfaced practical challenges. Bridging the Flask ML service and the Node.js backend required resolving JSON schema mismatches and CORS configuration; the scraper needed continual adjustment as anti-bot defenses evolved; and the Ngrok tunnel, which generates a new URL on restart, must be replaced by a fixed endpoint in production. The two functional defects also indicated that state-transition logic in asynchronous, multi-actor workflows was more failure-prone than the synchronous recommendation pipeline, motivating broader test coverage.

Data collection and privacy also present limitations. Data were scraped from a single public portal without a formal review of its terms of service or robots.txt, limiting reproducibility and coverage. Applicant data (GPA, test scores, contact information) relies on MongoDB Atlas at-rest encryption without application-level field encryption, and no data-retention policy exists. These should be addressed through scraping-consent review, source diversification, and a retention policy before production deployment. Recommendation bias may also arise from uneven country and institution representation in the scraped dataset, which has not been audited (Section IV).

Several threats to validity should also be noted. The held-out evaluation sets are small ($n = 27$ for the original splits, and 27 and 40 queries for the baseline comparison), and with approximately 72 institutions split across five folds, each test fold contains few universities, which partly explains the wide fold-to-fold variance. Both target variables (university ranking and enrolled Nepalese-student count) are proxies for recommendation quality rather than direct measures of applicant outcomes, and the Model 2 target reflects peer density rather than academic suitability. Finally, CatBoost was not re-evaluated under the corrected protocol, so its comparison rests only on the original single split. These factors reinforce our view that the reported results should be read as preliminary.

VI. CONCLUSION AND FUTURE WORK

This paper presented UNIFINDER, a dual-population recommendation system for Nepalese students planning to study abroad, combining tree-based regression with peer video consultation. Its central contribution is twofold: treating international and Nepalese applicants as requiring structurally different prediction targets (eligibility ranking versus peer

density), and conducting a transparent, corrected evaluation that identified and addressed institution-level data leakage. Under university-grouped cross-validation, both models show modest and variable generalization to unseen institutions (best corrected mean R^2 of 0.18 and 0.31), well below the original single-split estimates, and current recommendations derive most of their value from rule-based eligibility filtering rather than learned ranking. We present these results as an honest characterization of the system's current capabilities, with the population-specific framing remaining a reasonable direction whose predictive value must be established at larger scale.

This work has limitations. The dataset comes from a single source portal over a limited time window (131 records, approximately 72 institutions), so generalization to other sources, periods, or populations is untested, and corrected cross-validation shows this size is insufficient for reliable generalization to unseen institutions. The evaluation measures predictive accuracy and structural properties of the recommendation lists, but does not include ranking-quality metrics such as Precision@K or NDCG, expert assessment, or real admission outcomes, which require ground-truth labels and user participation not yet available. The platform's potential to reduce dependence on biased consultancies (Table I) remains a design goal rather than a demonstrated outcome, and representational bias in the scraped data has not been audited. Future work should prioritize: (i) expanding the dataset across additional portals, countries, and time periods; (ii) logging real user outcomes (applications, admissions, feedback) to enable Precision@K and NDCG; (iii) a user study comparing UNIFINDER against consultancy advice; and (iv) resolving the privacy and scraping-governance limitations in Section V before production deployment. Secondary directions include a native mobile client, multi-language support, and a secure payment gateway for premium consultations.

ACKNOWLEDGMENT

AI-assisted tools were used to help structure and refine the writing; all technical work, data, and conclusions are the authors' own.

REFERENCES

- [1] Ministry of Education, Science and Technology (MoEST), "Educational NOC Data FY 2022/23," Kathmandu, Nepal. [Online]. Available: <https://moest.gov.np>.
- [2] "150 consultancies sent over 700 Nepalis to an Australian institute without accreditation," The Kathmandu Post, Feb. 25, 2019. [Online]. Available: <https://kathmandupost.com/national/2019/02/25/150-consultancies-sent-over-700-nepalis-to-an-australian-institute-without-accreditation>
- [3] "As study abroad numbers rise, Nepali students land in trouble," University World News, Mar. 1, 2019. [Online]. Available: <https://www.universityworldnews.com/post.php?story=20190301101826555>
- [4] R. Burke, "Hybrid recommender systems: Survey and experiments," User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, 2002.
- [5] D. R. Roberts et al., "Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure," Ecography, vol. 40, no. 8, pp. 913–929, 2017.
- [6] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," Patterns, vol. 4, no. 9, Art. no. 100804, 2023.
- [7] F. Ricci, L. Rokach, and B. Shapira, Eds., Recommender Systems Handbook, 2nd ed. New York, NY, USA: Springer, 2015.
- [8] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
- [9] IDP Education, "Study Abroad Services – Apply Online," [Online]. Available: <https://www.idp.com/>. Accessed: Apr. 2025.

- [10] ApplyBoard, "University Matching for Global Students," [Online]. Available: <https://www.applyboard.com/>. Accessed: Apr. 2025.
- [11] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [12] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, USA, Aug. 2016, pp. 785–794.
- [13] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.
- [14] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
- [15] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," in Advances in Neural Information Processing Systems (NeurIPS), vol. 31, 2018.
- [16] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in Proc. 5th Berkeley Symp. Mathematical Statistics and Probability, vol. 1, Berkeley, CA, USA: Univ. of California Press, 1967, pp. 281–297.