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

AN INTELLIGENT PLATFORM FOR STUDENT SKILL-GAP ANALYSIS, INTERNSHIP MATCHING AND PLACEMENT PREDICTION

N. Akshaya Nethra[1], S. Poornima[2]

Dept. of Computer Science and Engineering (Emerging Technologies), SRM Institute of Science and Technology, Vadapalani Campus Chennai, India[1,2]

Abstract: Campus placement depends on a combination of academic, skill and activity factors, yet students usually learn how ready they are only when the placement season has begun. This paper presents PlaceWise, an end-to-end placement-prediction system that selects a classifier by cross-validation, reports a placement probability and turns every prediction into personalised improvement recommendations. Six classifiers—Logistic Regression, K-Nearest Neighbours, Support Vector Machine, Decision Tree, Random Forest and XGBoost—were trained on a public dataset of 10,000 students described by ten features and compared with stratified five-fold cross-validation. Logistic Regression achieved the highest mean F1-score (0.759 ± 0.022) and, on a stratified hold-out set of 2,000 students, reached an accuracy of 80.85%, a precision of 0.770, a recall of 0.776, an F1-score of 0.773 and a ROC-AUC of 0.884, matching or exceeding the tree ensembles on accuracy, F1-score and ROC-AUC. A what-if engine re-scores each student with realistic one-step changes and ranks them by their effect; for a borderline student, the three best changes raised the predicted probability from 28% to 61%. Every prediction is also explained by its main strengths and weak points, measured by re-scoring the student with each feature set to its average, and given a rule-based company tier; for a whole class, the system summarises tiers, readiness, common weak areas and a training plan. The model is deployed in a Flask application with single and bulk prediction, a JSON API and an SQLite history, verified by 21 automated tests.

Keywords: placement prediction, educational data mining, logistic regression, XGBoost, cross-validation, what-if analysis

I. INTRODUCTION

Placement in a company is the main outcome that engineering students, their families and their colleges look for at the end of a degree. Whether a student is placed depends on CGPA and school marks, aptitude, communication skills, internships, projects, certifications, extracurricular activities and placement training. In most colleges these factors are reviewed by hand, typically by sorting students on CGPA, which is slow and ignores most of the information available.

Machine learning can learn the relationship between such attributes and the placement outcome from past records [1]. Several recent studies have applied classifiers to placement data and reported accuracies of up to about 89% [2], [3]. Two problems remain, however. First, models are often compared on a single train/test split, so small accuracy differences may be due to chance [10]. Second, a predicted label tells a student little about what to do next.

This paper presents PlaceWise, which addresses both problems. Its contributions are: (1) a comparison of six classifiers from four families on a 10,000-student dataset with stratified five-fold cross-validation for selection and an untouched hold-out set for reporting; (2) a placement probability with four readiness levels instead of a bare label; (3) a what-if engine that uses the selected model to rank realistic improvements for each student, together with a model-agnostic explanation of the main strengths and weak points behind each prediction and a transparent, rule-based company tier; and (4) a deployable web application with bulk prediction and class-level insights for placement cells, a validated API and stored history.

II. RELATED WORK

Romero and Ventura [1] reviewed educational data mining and identified classification of student outcomes as one of its main tasks, stressing that results must lead to action. Pathak et al. [2] compared Logistic Regression, Decision Tree, Random Forest, XGBoost, K-Nearest Neighbours and Naive Bayes with a soft-voting ensemble on a Kaggle dataset of 2,966 engineering students; XGBoost and the ensemble reached about 89% accuracy. Meshram et al. [3] compared several models for placement prediction and discussed their value for placement cells. Neither study provides per-student guidance.

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

The six algorithms used here are standard: Logistic Regression [4], nearest-neighbour classification [5], support-vector networks [6], decision-tree induction [7], Random Forests [8], which also introduced permutation importance, and gradient-boosted trees in XGBoost [9]. Kohavi [10] showed that k-fold cross-validation gives lower-variance estimates than a single hold-out split for model selection. Grinsztajn et al. [12] found that tree ensembles usually outperform deep networks on medium-sized tabular data; deep learning was therefore not considered. All models were implemented with scikit-learn [11] and XGBoost [9].

Romero and Ventura [1] reviewed educational data mining and identified classification of student outcomes as one of its major research areas. Their study explained how educational data can be analysed to discover useful patterns and support decision-making. Educational data mining helps institutions understand student performance by examining academic records and other learning-related information. Classification techniques can be used to predict student outcomes based on different input attributes. These concepts provide a foundation for developing machine learning-based student placement prediction systems. Such systems can help students understand their expected outcomes and improve their preparation for employment opportunities.

Pathak et al. [2] compared several machine learning algorithms, including Logistic Regression, Decision Tree, Random Forest, XGBoost, K-Nearest Neighbours and Naive Bayes, using a Kaggle dataset containing 2,966 engineering students. Their study also evaluated a soft-voting ensemble approach that combined multiple classification models. XGBoost and the ensemble model achieved approximately 89% accuracy. The comparison demonstrated the effectiveness of machine learning algorithms in predicting student placement outcomes. It also highlighted the importance of selecting appropriate classification techniques for obtaining reliable predictions. This research is closely related to the proposed system because it focuses on predicting the placement status of engineering students using academic and other student-related information. Meshram et al. [3] investigated different machine learning approaches for student placement prediction and discussed their usefulness in supporting college placement activities. Their research highlighted the role of predictive models in analysing student information and identifying placement possibilities. Machine learning algorithms can process multiple student attributes and classify individuals according to their expected employment outcomes. Such predictions can assist placement departments in understanding student placement trends and planning suitable training activities. However, institutional-level prediction and individual student guidance are different requirements. The proposed system focuses on providing an individual prediction through a web-based interface.

Logistic Regression [4] is a widely used supervised machine learning algorithm for binary classification problems. It estimates the probability of an outcome based on the given input features. In student placement prediction, Logistic Regression can analyse attributes such as CGPA, attendance, technical skills, communication skills and internship experience to estimate whether a student is likely to be placed. Its simple structure and ability to generate probability-based predictions make it useful for placement classification. The predicted probability can be converted into two output categories, namely Placed and Not Placed, based on a selected classification threshold.

K-Nearest Neighbours (KNN) [5] is a classification algorithm that predicts the class of a new data point by examining the nearest data points in the training dataset. It calculates the distance between student records and identifies students with similar characteristics. In placement prediction, KNN can compare a student's academic performance and skill-related details with those of other students. The majority class among the nearest neighbours determines the predicted result. This approach is useful for understanding similarities between students with different academic and professional backgrounds. However, its performance depends on the selection of the number of neighbours and the scaling of input features.

Support Vector Machines (SVM) [6] are supervised learning algorithms commonly used for classification and regression tasks. SVM identifies an optimal decision boundary that separates different classes of data. In a student placement prediction system, it can distinguish between placed and not-placed students by analysing different academic and skill-related attributes. The algorithm aims to maximise the margin between the classes, which can help improve classification performance. SVM is particularly useful when the dataset contains multiple features that influence the placement outcome. Selecting an appropriate kernel function can also help the algorithm handle complex relationships between input attributes.

Decision Tree learning [7] is a supervised machine learning technique that represents decisions in the form of a tree structure. It consists of a root node, decision nodes and leaf nodes, where each node represents a condition or prediction. In student placement prediction, the algorithm can evaluate factors such as CGPA, attendance, technical knowledge, communication skills and internship experience to determine the final placement category. Decision Trees are easy to understand because their predictions can be represented through simple decision rules. This interpretability makes them useful for identifying the student attributes that influence placement outcomes.

Random Forests [8] are ensemble learning algorithms that combine multiple decision trees to generate a final prediction. Instead of depending on a single decision tree, Random Forest creates several trees using different samples and feature combinations. The final classification is determined through majority voting among the individual trees. In student placement prediction, Random Forest can analyse multiple academic and skill-related features to classify students into placement categories. It can also provide information about feature importance, helping identify which student attributes contribute more to the prediction. This makes Random Forest a useful approach for handling multiple input variables in placement-related applications.

XGBoost [9] is an advanced gradient-boosting algorithm designed to improve prediction performance by combining multiple decision trees sequentially. Each new tree attempts to correct the errors made by the previous trees. This process gradually improves the overall prediction model. XGBoost is suitable for structured and tabular datasets containing student-related information. In placement prediction, it can analyse complex relationships between academic performance, technical skills, internships and other employability factors. Its boosting approach and regularisation techniques help improve model performance while controlling overfitting. It is therefore considered an important algorithm for comparison with other classification techniques.

Kohavi [10] studied the effectiveness of different model evaluation methods and explained the advantages of k-fold cross-validation over a single hold-out validation approach. In k-fold cross-validation, the available dataset is divided into multiple subsets, and the model is trained and evaluated several times using different combinations of training and testing data. This method provides a more reliable estimate of model performance and reduces the dependence on a single data split. In student placement prediction, cross-validation can be used to compare different classification algorithms and evaluate their consistency. It helps in selecting a suitable model before integrating it into the final prediction application.

Scikit-learn [11] provides a comprehensive Python library for implementing machine learning algorithms and evaluating their performance. It supports classification techniques such as Logistic Regression, K-Nearest Neighbours, Support Vector Machines, Decision Trees and Random Forests. It also provides functions for data preprocessing, feature selection, model training, testing and performance evaluation. These features make it useful for developing machine learning applications without implementing every algorithm from the beginning. In the proposed student placement prediction system, scikit-learn supports the implementation and comparison of classification models using student-related input features.

Grinsztajn et al. [12] investigated the performance of different machine learning approaches on medium-sized tabular datasets. Their findings indicated that tree-based ensemble methods often perform better than deep learning approaches on this type of structured data. This observation is relevant to student placement prediction because student information is generally represented in tabular form, containing numerical and categorical attributes. Features such as CGPA, attendance, technical skills, communication skills, internships and certifications can be processed using tree-based machine learning algorithms. Based on these findings, the proposed system focuses on traditional machine learning and ensemble techniques rather than using complex deep learning architectures.

The proposed system combines the concepts discussed in the above studies to develop a machine learning-based student placement prediction application. It focuses on analysing individual student information and generating a placement outcome through a user-friendly web interface. Students can enter their academic and skill-related details, including CGPA, attendance, technical skills, communication skills, internships, projects and certifications. The system processes the entered information and uses a machine learning model to predict whether the student is Placed or Not Placed. Unlike approaches that mainly concentrate on institutional placement analysis and model comparison, the proposed application focuses on making individual placement prediction accessible through a web-based platform.

[1] Romero and Ventura: This study justifies the use of educational data mining techniques to analyse student information and classify placement outcomes.

[2] Pathak et al.: This study justifies comparing multiple machine learning algorithms to identify a suitable model for student placement prediction.

[3] Meshram et al.: This research supports the development of a placement prediction system that helps evaluate student employability.

[4] Logistic Regression: It is included because it provides a simple and effective baseline for predicting binary placement outcomes.

[5] KNN: It is included because students with similar academic and skill profiles may have similar placement outcomes.

[6] SVM: It is included because it can distinguish between placed and not-placed students using classification boundaries.

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

[7] Decision Tree: It is included because its decision-making structure makes placement predictions easier to understand.
[8] Random Forest: It is included because combining multiple decision trees can improve prediction stability and identify important student attributes.
[9] XGBoost: It is included because gradient boosting can capture complex relationships among student attributes.
[10] Kohavi: This study justifies using k-fold cross-validation to evaluate model performance more reliably.
[11] Scikit-learn: It justifies the selection of a common machine learning library for implementing and evaluating the algorithms.
[12] Grinsztajn et al.: This study supports using traditional machine learning and tree-based models for structured student data.

III. DATASET AND PREPROCESSING

A. Dataset

The Placement Prediction Dataset from Kaggle [13] contains 10,000 student records with ten input features and the label PlacementStatus. There are no missing values or duplicate rows. 4,197 students (41.97%) were placed. The features are CGPA (6.5–9.1), internships (0–2), projects (0–3), workshops/certifications (0–3), aptitude test score (60–90), soft skills rating (3.0–4.8), extracurricular activities (Yes/No), placement training (Yes/No) and 10th (SSC) and 12th (HSC) marks.

B. Exploratory Analysis

Placed students score higher on every feature (Table I). The strongest associations are with the aptitude score (r = 0.52), HSC marks (0.51) and extracurricular activities (0.48), and the two Yes/No flags show the largest gaps in placement rate: 51.6% of students who attended placement training were placed against 15.6% of those who did not, and 62.0% of students with extracurricular activities against 13.7% without. All features are positively correlated with placement (0.26–0.52) and with each other (up to 0.57), so all ten were kept.

TABLE I. MEAN FEATURE VALUES BY CLASS

Feature	Placed	Not placed	r
Aptitude score	84.46	75.83	0.52
HSC marks	79.81	70.67	0.51
Extracurricular (Yes)	86.5%	38.3%	0.48
Projects	2.51	1.68	0.48
SSC marks	74.92	64.99	0.47
Soft skills	4.53	4.17	0.43
CGPA	8.02	7.47	0.42
Workshops	1.40	0.74	0.36
Placement training (Yes)	90.0%	61.0%	0.32
Internships	1.25	0.90	0.26

r = Pearson correlation with the placement label.

C. Preprocessing and Split

The identifier is removed, Yes/No values are encoded as 1/0 and the label as 1 = Placed. The data is split 80:20 with stratification and a fixed seed (42) into 8,000 training students (3,358 placed) and 2,000 hold-out students (839 placed). Each model is a scikit-learn pipeline whose first step is a StandardScaler, so scaling statistics are learned from training folds only. The same preprocessing module is used at training and prediction time.

IV. METHODOLOGY

A. Candidate Models

Six models covering linear, instance-based, kernel and tree families were configured with moderate settings (Table II). All return class probabilities.

TABLE II. MODELS AND SETTINGS

Model	Key settings
Logistic Regression	L2, C = 1.0
k-NN	k = 25, distance weights
SVM	RBF kernel, C = 1.0, Platt scaling
Decision Tree	max depth 5, min leaf 20
Random Forest	300 trees, max depth 8, min leaf 5
XGBoost	300 trees, depth 3, η = 0.05, subsample and colsample 0.8

Logistic Regression models the placement probability of a student with standardised features z as

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta^T z)}} \quad (1)$$

where z is the vector of standardised features and β the learned weights. Gradient boosting adds trees h that fit the negative gradient of the loss, scaled by the learning rate η:

$$F_m(x) = F_{m-1}(x) + \eta\, h_m(x) \quad (2)$$

B. Model Selection Protocol

Each pipeline is evaluated with stratified five-fold cross-validation on the training set. The model with the highest mean F1-score is selected (ROC-AUC breaks ties). All models are then refitted on the full training set and scored once on the hold-out set, which is never used for any choice. A student is predicted Placed when $p \geq t$ with $t = 0.50$; the effect of t is analysed in Section VI-D.

C. Readiness Levels and What-If Recommendations

The probability is mapped to four readiness levels: Low ($p < 0.30$), Moderate ($0.30 \leq p < 0.50$), Good ($0.50 \leq p < 0.75$) and High ($p \geq 0.75$). For recommendations, let x be a student and x′ the same student with feature j improved by one realistic step: +5 aptitude marks, +0.3 CGPA, +0.3 soft-skills rating, one more project, workshop or internship, or switching placement training or extracurricular activities from No to Yes. The gain of change j is

$$\Delta_j = f(x'_j) - f(x) \quad (3)$$

where f is the selected model. A step that would pass the largest value seen in the data is cut back to that value, a feature already at that value is skipped, school marks are never changed and gains below 0.5 percentage points are discarded. Up to three changes with the largest gains are shown, together with the probability obtained when they are applied at once.

D. Explanations and Company Tier

To explain a prediction, each feature j of the student is replaced in turn by its training mean, giving a copy $\tilde{x}$ (with subscript j), and the effect of the feature is measured on the log-odds scale:

$$e_j = logit\, f(x) - logit\, f(\tilde{x}_j) \quad (4)$$

where $\text{logit}(p) = \ln(p / (1 - p))$. The two largest positive effects (≥ 0.1) are reported as strengths and the two largest negative effects as weak points, with a one-line summary. Log-odds keep the explanation meaningful for students far from the threshold, and the method works for any classifier. Because the dataset has no company or salary data, the company tier is not learned but assigned by rules common in Indian campus placements: Super Dream (10+ LPA) requires $p \geq 0.85$, CGPA ≥ 8.0, aptitude ≥ 85 and at least two projects; Dream (6–10 LPA) requires $p \geq 0.70$, CGPA ≥ 7.5 and aptitude ≥ 80; Mass Recruiter (3–6 LPA) requires $p \geq 0.50$; other students are Not Yet Ready. The gap to the next tier is reported, and the tier is labelled as an estimate.

E. Evaluation Metrics

Accuracy, precision (P), recall (R) and the F1-score of the Placed class are computed from true positives (TP), false positives (FP) and false negatives (FN):

$$P = \frac{TP}{TP + FP} \quad (5)$$

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

$$R = \frac{TP}{TP + FN} \quad (6)$$

$$F_1 = \frac{2PR}{P + R} \quad (7)$$

ROC-AUC summarises ranking quality over all thresholds.

V. SYSTEM IMPLEMENTATION

PlaceWise is implemented in Python with a four-layer architecture (Fig. 1). A Flask application serves six pages: Home, Predict (form, probability gauge, readiness badge and ranked tips), Bulk Predict (CSV upload of up to 5,000 students with reasons, tiers, class-level charts, a suggested training plan, filters and a downloadable or printable report), Model Comparison, Data Insights and History. A JSON endpoint, POST /api/predict, returns the prediction, probability, readiness, reasons, company tier and tips. One validation function checks presence, type and range of every field for all routes; the API answers invalid requests with HTTP 422 and a message per field. Predictions made through the Predict form are stored in SQLite with parameterised queries. The trained pipelines are saved with joblib, and 21 pytest tests cover data loading, validation, the saved model, the tips engine, the explanations and tiers, every page and the main routes.

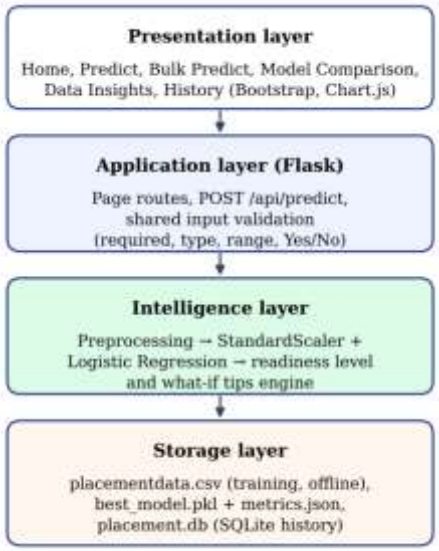


Fig. 1. Layered architecture of PlaceWise.

VI. EXPERIMENTAL RESULTS

A. Setup

Experiments were run with Python 3.13, scikit-learn 1.9 and XGBoost 3.4 on a CPU. Training, cross-validation and chart generation for all six models take under one minute; a single prediction with tips takes about 15 ms.

B. Model Comparison

Table III reports the cross-validated F1-score and the hold-out results. Logistic Regression had the best mean CV F1-score (0.759) and the best hold-out accuracy (0.8085), F1-score (0.773) and ROC-AUC (0.884). The random forest and XGBoost were close (accuracy 0.805 and 0.800). The differences among the top four models are smaller than the fold-to-fold standard deviation (about 0.02), so they are statistically similar; Logistic Regression is preferred because it is equally accurate, faster and interpretable, and its hold-out Brier score (0.136) is the lowest of the six models, level with the random forest. k-NN reached 99.9% training accuracy but 79.5% on the hold-out set, a clear case of over-fitting, whereas Logistic Regression scored 79.8% on training and 80.9% on hold-out data. Fig. 2 shows the ROC curves.

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

TABLE III. CROSS-VALIDATION AND HOLD-OUT RESULTS

Model	CV F1	Acc.	P	R	F1	AUC
Logistic Reg.	**0.759**	**0.809**	**0.770**	**0.776**	**0.773**	**0.884**
k-NN	0.744	0.795	0.761	0.745	0.753	0.876
SVM	0.750	0.800	0.773	0.739	0.756	0.861
Decision Tree	0.742	0.784	0.741	0.745	0.743	0.860
Random Forest	0.751	0.805	0.773	0.758	0.765	0.883
XGBoost	0.753	0.800	0.766	0.752	0.759	0.880

CV F1 = mean of five folds on the training set; other columns on the 2,000-student hold-out set (839 placed). P = precision, R = recall.

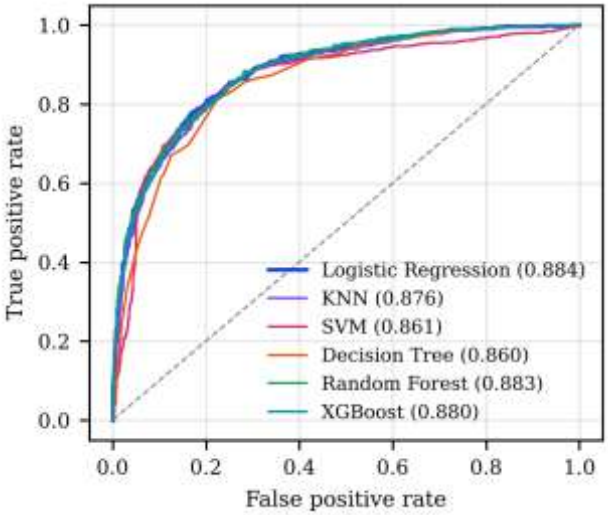


Fig. 2. ROC curves of the six models on the hold-out set (AUC in brackets).

C. Error Analysis and Feature Importance

Fig. 3 shows the confusion matrix of the selected model. Of 839 placed students, 651 were identified and 188 missed; 195 of 1,161 not-placed students were predicted Placed. The two error types are almost balanced, and so are precision and recall.

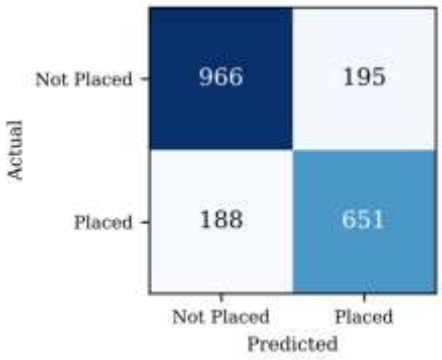


Fig. 3. Confusion matrix of Logistic Regression on the hold-out set (t = 0.50).

Permutation importance on the hold-out set (Fig. 4) ranks the aptitude score first (0.025), followed by placement training (0.010), extracurricular activities (0.009) and soft skills (0.009); internships have no measurable effect. The standardised

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

coefficients agree on the three strongest and the two weakest features (aptitude 0.611, placement training 0.392, extracurricular 0.360, …, workshops 0.095, internships −0.011). The importance values are small because the features are correlated, so shuffling one leaves much of its information in the others.



Fig. 4. Permutation feature importance of Logistic Regression.

D. Threshold Selection

Table IV and Fig. 5 show the trade-off controlled by t. Accuracy peaks at t = 0.50, where precision and recall are balanced. Lowering t to 0.30 finds 88% of placed students but adds 346 false positives; raising it to 0.70 gives a precision of 0.85 but misses 39% of placed students.

TABLE IV. EFFECT OF THE DECISION THRESHOLD

t	Acc.	P	R	F1	Pred. placed
0.30	0.777	0.681	0.881	0.768	1,085
0.40	0.795	0.722	0.830	0.772	964
0.50	**0.809**	**0.770**	**0.776**	**0.773**	**846**
0.60	0.804	0.804	0.703	0.750	734
0.70	0.794	0.854	0.614	0.714	603

E. Cross-Validation Stability

Across the five folds, the F1-score of Logistic Regression ranged from 0.724 to 0.790 (mean 0.759 ± 0.022) and the ROC-AUC from 0.857 to 0.899 (0.877 ± 0.013). The hold-out F1-score of 0.773 lies inside this range, indicating that the reported result does not depend on a favourable split.

F. Application Output

Four profiles were evaluated through the web form. A strong student (CGPA 8.4, aptitude 88, training and activities) received 88% (High). A student with CGPA 7.6, aptitude 78 and no extracurricular activities received 28% (Low); the engine recommended extracurricular activities (28% → 44%), +5 aptitude marks (→ 36%) and one more project (→ 34%), and all three together gave 61%, changing the prediction to Placed. A student at 49% without placement training was told that training alone would raise the chance to 70%. A weak profile received 1%, and its best changes reached only 5%, showing that the engine does not give false hope. The explanations agreed with the tips: for the 49% student the only weak point was the missing placement training (−16 points). The strong student was placed in the Super Dream tier, while the 28% student was Not Yet Ready, as the Mass Recruiter tier needs a probability of at least 50%. In a bulk upload of 30 hold-out students, 23 predictions (76.7%) matched the true labels; the class view showed 3 Super Dream, 1 Dream, 9 Mass Recruiter and 17 Not Yet Ready students and grouped the 17 by their best single change (placement training for 8, extracurricular activities for 7, aptitude and projects for one each).

VII. DISCUSSION AND LIMITATIONS

On this dataset a linear model is sufficient because each feature has a positive, roughly monotonic relationship with placement; the extra capacity of the ensembles brings no gain. About one student in five is still misclassified, so the probability and readiness level, rather than the label alone, should guide decisions. The accuracy reported by Pathak et al. [2] was obtained on a different, smaller dataset with different features and is not directly comparable. The dataset may not represent a particular college and lacks attendance, backlogs and department. The recommendations reflect

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

associations learned by the model, not causal effects, and are presented as estimates. The company tier is a rule-based estimate, not a learned prediction, because the dataset contains no company or salary data. Finally, the prototype has no authentication and stores history locally.

VIII. CONCLUSION AND FUTURE WORK

PlaceWise combines cross-validated model selection, placement probabilities with readiness levels, explanations of each prediction, a transparent company-tier estimate and model-based what-if recommendations in a usable web application. Logistic Regression was selected among six classifiers and achieved 80.85% accuracy, an F1-score of 0.773 and a ROC-AUC of 0.884 on 2,000 unseen students, while the recommendation engine turned predictions into concrete plans that could move borderline students above the threshold. Future work includes retraining on institutional records with additional features, learning the company tier and salary from real offer data, SHAP-based explanations, role-based access, and tracking students' probabilities across a semester.

ACKNOWLEDGMENT

The authors thank S.Poornima, Assistant Professor, and Dr. P. Chitra, Professor and Head, Department of Computer Science and Engineering (Emerging Technologies), SRM Institute of Science and Technology, Vadapalani Campus, for their guidance and support.

REFERENCES

[1] C. Romero and S. Ventura, "Educational data mining: A review of the state of the art," IEEE Trans. Syst., Man, Cybern. C, vol. 40, no. 6, pp. 601–618, Nov. 2010.
[2] A. Pathak, M. Matcha, M. Gopisetti, and S. Joshi, "A machine learning framework for predicting student placement outcomes," Ingénierie des Systèmes d'Information, vol. 30, no. 7, pp. 1715–1721, 2025, doi: 10.18280/isi.300704.
[3] E. Meshram, S. M. Jadhav, S. S. Shirame, S. S. Joshi, and R. S. Golande, "Placement prediction using various machine learning models and their efficiency comparison," Preprints, Jan. 2026, doi: 10.20944/preprints202601.0825.v1.
[4] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.
[5] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Trans. Inf. Theory, vol. 13, no. 1, pp. 21–27, Jan. 1967.
[6] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.
[7] J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, no. 1, pp. 81–106, 1986.
[8] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[9] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[10] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Proc. 14th Int. Joint Conf. Artificial Intelligence (IJCAI), 1995, pp. 1137–1143.
[11] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
[12] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?" in Advances in Neural Information Processing Systems 35, Datasets and Benchmarks Track, 2022.
[13] "Placement Prediction Dataset" (placementdata.csv), Kaggle. Accessed: Oct. 2026. [Online]. Available: https://www.kaggle.com/datasets

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