

END TO END CAMPUS PLACEMENT PREDICTION SYSTEM USING ENSEMBLE LEARNING

¹ L.Mallika, ² Yadagiri Omkar, ³ Gangula Abhilash, ⁴ Vanga Ashwith Reddy, ⁵ Jatothu Nithin

¹Assistant Professor, ^{2,3,4,5}Students

Department of ECE

Siddhartha Institute of Technology & Sciences, Narapally

mallika_ece@siddhartha.co.in, 24tq1a0441@siddhartha.co.in, 24tq1a0415@siddhartha.co.in,

24tq1a0440@siddhartha.co.in, 24tq1a0419@siddhartha.co.in

ABSTRACT

Campus placement is one of the most important outcomes of an undergraduate engineering programme, both for the students who hope to begin their careers and for the institution whose reputation depends on its placement record. Training and placement cells spend a great deal of effort preparing students for recruitment drives, yet they usually learn which students are at risk of remaining unplaced only after the recruitment season has ended. This paper presents an end to end campus placement prediction system that uses ensemble learning to estimate, well before the drives begin, how likely each student is to be placed.

The system is built on academic and profile data that institutions already hold. The attributes considered include secondary and higher secondary marks, semester grade point averages, the number of active and cleared backlogs, branch of study, internship experience, certifications, participation in technical events, aptitude test scores, communication assessment scores, and the results of mock interviews. These records are collected from the examination section and the placement cell, cleaned, encoded, and combined into a single structured dataset in which every row describes one student.

Several individual classifiers are trained on this dataset, including logistic regression, a decision tree, k-nearest neighbours, and a support vector machine, so that a baseline can be established. Ensemble methods are then applied, namely a random forest, gradient boosting, an extreme gradient boosting model, and a soft-voting ensemble that combines the strongest individual learners. The models are compared using accuracy, precision, recall, F1 score, and the area under the ROC curve under stratified cross-validation, and the ensemble that gives the most balanced performance is selected for deployment.

In addition to a yes or no prediction, the system estimates the expected salary band for students who are predicted to be placed and reports the factors that most influence each prediction. Feature importance and per-student explanations show, for example, whether a student's weak point lies in aptitude, in communication, or in the lack of practical exposure. This information is presented through a web dashboard in which the placement officer can view the overall readiness of a batch, filter students by department, and open an individual profile to see the recommended areas of improvement.

The complete pipeline, from data ingestion to the dashboard, is implemented in Python with commonly used open-source libraries, so that it can be maintained by the

institution without specialised infrastructure. The objective is not to label students but to help the placement cell direct its training resources towards those who need them most, early enough for the intervention to make a difference. Future extensions may include company-specific eligibility matching, integration with learning management systems, and periodic retraining as new placement seasons add fresh data.

I. INTRODUCTION

Engineering colleges in India admit large numbers of students every year, and a considerable portion of them expect to secure employment through on-campus recruitment. The placement percentage of an institution influences its accreditation, its ranking, and the choices of future applicants. For the students themselves, the placement season in the final year is often the single most stressful period of the programme, because the result of a few days of tests and interviews can shape the start of a career.

Placement cells usually prepare students through aptitude classes, soft-skill sessions, and mock interviews, but these activities are generally delivered to the whole batch in the same way. The staff do not have a reliable method for identifying, a semester or two in advance, which students are likely to struggle. Decisions about who needs extra coaching are based on impressions formed during training sessions or on a single attribute such as the aggregate percentage. As a result, students who are academically average but weak in communication, or strong in communication but carrying backlogs, are often not noticed until they have already failed several interviews.

Machine learning offers a practical way to address this gap. Historical placement records contain the outcomes of hundreds of students together with the attributes that were available about them before recruitment. A classifier trained on these records can learn the combinations of factors that are associated with success and can apply that knowledge to the current batch. Ensemble learning is particularly suitable for this kind of tabular data, because combining many trees or many different learners tends to reduce variance, handle mixed feature types well, and give more stable predictions than a single model.

The system described in this paper takes an end to end view of the problem. It covers the collection of student data from existing institutional records, preprocessing and feature engineering, the training and comparison of individual and ensemble classifiers, the explanation of predictions, and a dashboard through which placement staff and students can use the results. The intention is to give the placement cell an early and evidence-based picture of batch readiness, so that training effort can be planned for each student rather than applied uniformly to everyone.

II. LITERATURE SURVEY

Educational data mining has been an active field of research for more than two decades, and a large portion of the published work concerns the prediction of student performance. Early studies applied decision trees, naive Bayes classifiers, and rule-

based systems to examination records in order to forecast final grades or to identify students at risk of dropping out. These studies showed that prior academic performance is a strong indicator of later outcomes, but they also reported that attendance, family background, and engagement in extracurricular activities add useful information. Researchers noted that institutional data is often incomplete and inconsistently formatted across departments, so a substantial part of the effort in such projects goes into cleaning and integrating records. The general conclusion of this body of work is that routinely collected academic data can support early intervention if it is organised properly and analysed with suitable models, and that the benefit is greatest when predictions are made early enough for staff to act on them.

Placement prediction emerged as a specific application within this field as engineering and management institutions began to maintain digital records of recruitment outcomes. Published studies have used attributes such as secondary and higher secondary percentages, degree aggregate, number of backlogs, specialisation, work experience, and aptitude scores to predict whether a student will be placed. Logistic regression and decision trees were among the first methods used, largely because their outputs are easy to interpret for non-technical staff. Later studies compared these with support vector machines and k-nearest neighbours and generally reported moderate accuracy, often in the range of seventy to eighty five percent depending on the dataset. Several authors pointed out that most placement datasets are small and specific to a single institution, which limits how far the results can be generalised, and that the proportion of placed and unplaced students can be uneven enough to distort a simple accuracy figure.

Ensemble learning has been studied widely as a way to improve the reliability of predictions on structured data. Bagging methods such as random forest train many decision trees on bootstrap samples and average their outputs, which reduces variance and makes the model less sensitive to noise in individual records. Boosting methods such as AdaBoost, gradient boosting, and extreme gradient boosting build trees sequentially so that each new tree concentrates on the errors of the previous ones, and they frequently achieve the highest accuracy on tabular benchmarks. Stacking and voting ensembles combine models of different types, relying on the fact that different learners make different mistakes. Studies that applied these methods to student and placement data have consistently reported gains of a few percentage points over single classifiers, together with more stable results across cross-validation folds, which is valuable when the dataset is small.

Interpretability has become an important concern as predictive models are introduced into educational decision making. Staff and students are unlikely to trust a prediction that cannot be explained, and an unexplained label may discourage a student rather than motivate improvement. Research in this area has used global feature importance from tree ensembles, permutation importance, partial dependence plots, and local explanation techniques based on Shapley values to show which attributes drive a given prediction. Studies on placement data frequently find that aptitude scores, communication assessments, internship experience, and the number of backlogs carry more weight than raw percentages alone. Authors have argued that presenting such explanations alongside predictions turns the model into a guidance tool, because it

indicates not only who is at risk but also which area of preparation deserves attention, and it allows staff to check that the model is relying on sensible factors.

More recent work has moved from isolated experiments towards complete systems that can be used by placement cells in practice. These systems typically include a data collection interface, a preprocessing pipeline that handles missing values and categorical encoding, a model training and selection stage, and a web interface built with frameworks such as Flask or Streamlit. Some studies also add salary prediction as a regression task and company recommendation based on eligibility criteria. Reported deployments emphasise the need for periodic retraining as recruitment patterns change from year to year, careful handling of personal data, and simple presentation of results for staff who are not familiar with machine learning. The system proposed in this paper follows this end to end direction, combining an ensemble classifier with explanation, salary band estimation, and a dashboard designed around the working routine of a placement cell.

III. SYSTEM ANALYSIS

The campus placement prediction system takes student records from the examination section and the placement cell and turns them into a readiness estimate for every student in the current batch. Academic attributes such as tenth and twelfth marks, semester grade point averages, and backlog counts are combined with profile attributes such as branch, internships, certifications, technical event participation, aptitude scores, communication scores, and mock interview ratings. The data is checked for missing and inconsistent values, categorical fields are encoded, and numerical fields are scaled where the chosen model requires it. Historical records with known placement outcomes are used for training, and the class balance between placed and unplaced students is examined so that the evaluation is not misleading. Several individual and ensemble classifiers are trained and compared under stratified cross-validation, and the best model is stored with its preprocessing steps. For each current student the system reports a placement probability, an expected salary band, and the attributes that most influence the prediction, and presents these through a dashboard used by the placement officer.

Existing System

In most institutions the assessment of placement readiness is carried out informally. The placement cell conducts aptitude and soft-skill training for the entire final-year batch, and trainers form impressions of individual students during these sessions. Academic records are held by the examination section in one format, while training attendance and mock test scores are kept by the placement cell in separate spreadsheets. When shortlists are prepared, staff mainly filter students by aggregate percentage and backlog count, because these are the fields that recruiters specify. There is no systematic method that combines all available attributes to estimate the likelihood of placement for each student in advance. Students who are at risk are therefore usually identified only after they have been rejected in several drives, which leaves very little time for targeted preparation. Where simple spreadsheet formulas or single decision trees have been tried, they tend to overfit the small datasets involved

and give unstable results from one year to the next. Students also receive little specific feedback about which aspect of their profile needs improvement.

Disadvantages of Existing System

- Placement readiness is judged informally from trainer impressions rather than from data.
- Academic and training records are kept in separate formats by different offices.
- Shortlisting relies mainly on aggregate percentage and backlog count alone.
- Students at risk are identified only after they have failed several recruitment drives.
- Training programmes are delivered uniformly to the whole batch regardless of individual need.
- Single-model approaches overfit small institutional datasets and give unstable results.
- No estimate of expected salary range is available to students or staff.
- Students receive little specific feedback on which area of their profile is weak.
- Historical placement data is not reused to improve future preparation.
- Batch-level readiness cannot be monitored conveniently by the placement officer.

Proposed System

The proposed system is an end to end placement prediction platform that uses ensemble learning on institutional student data. Academic records and placement cell data are imported from spreadsheets or a database and merged into a single dataset keyed by roll number. Preprocessing handles missing values, encodes branch and other categorical fields, and creates derived features such as average grade trend, total certifications, and a composite skill score from aptitude, communication, and mock interview results. Logistic regression, a decision tree, k-nearest neighbours, and a support vector machine are trained as baselines, followed by random forest, gradient boosting, extreme gradient boosting, and a soft-voting ensemble of the strongest learners. The models are compared on accuracy, precision, recall, F1 score, and ROC-AUC under stratified five-fold cross-validation, and the best is saved as a complete pipeline. A separate regression model estimates the salary band for students predicted to be placed. Shapley-value explanations identify the main factors behind each prediction, and a web dashboard presents batch readiness, individual profiles, and recommended areas of improvement.

Advantages of Proposed System

- Combines academic, skill, and experience attributes into a single readiness estimate.
- Identifies students at risk early enough for targeted training to be arranged.
- Ensemble models give more accurate and more stable predictions than single classifiers.
- Evaluation with precision, recall, F1, and ROC-AUC avoids misleading accuracy figures.

- Explains each prediction by showing the attributes that influence it most.
- Provides an expected salary band for students predicted to be placed.
- Allows placement staff to plan training according to individual weaknesses.
- Gives the placement officer a clear view of readiness across departments and batches.
- Uses data that the institution already collects, so no new data sources are needed.
- Can be retrained each year as new placement outcomes become available.

IV. METHODOLOGY

The methodology begins with the collection of historical student records together with their final placement outcomes and salary offers. Data from the examination section and the placement cell is merged by roll number, duplicate entries are removed, and missing values are filled with medians for numerical fields and the most frequent category for categorical fields. Branch, gender, and internship type are one-hot encoded, and derived features such as grade trend across semesters and a composite skill score are constructed. Exploratory analysis with histograms, correlation heatmaps, and class distribution plots guides the choice of features. The dataset is split into training and test portions with stratification, and class imbalance is handled with class weights or synthetic oversampling applied only to the training folds. Baseline and ensemble classifiers are trained, and hyperparameters are tuned through grid or randomised search with five-fold cross-validation. The selected model is evaluated on the held-out test set using accuracy, precision, recall, F1 score, ROC-AUC, and a confusion matrix, explanations are generated, and the full pipeline is serialised and served to the dashboard.

SYSTEM ARCHITECTURE

End to end campus placement prediction system using ensemble learning

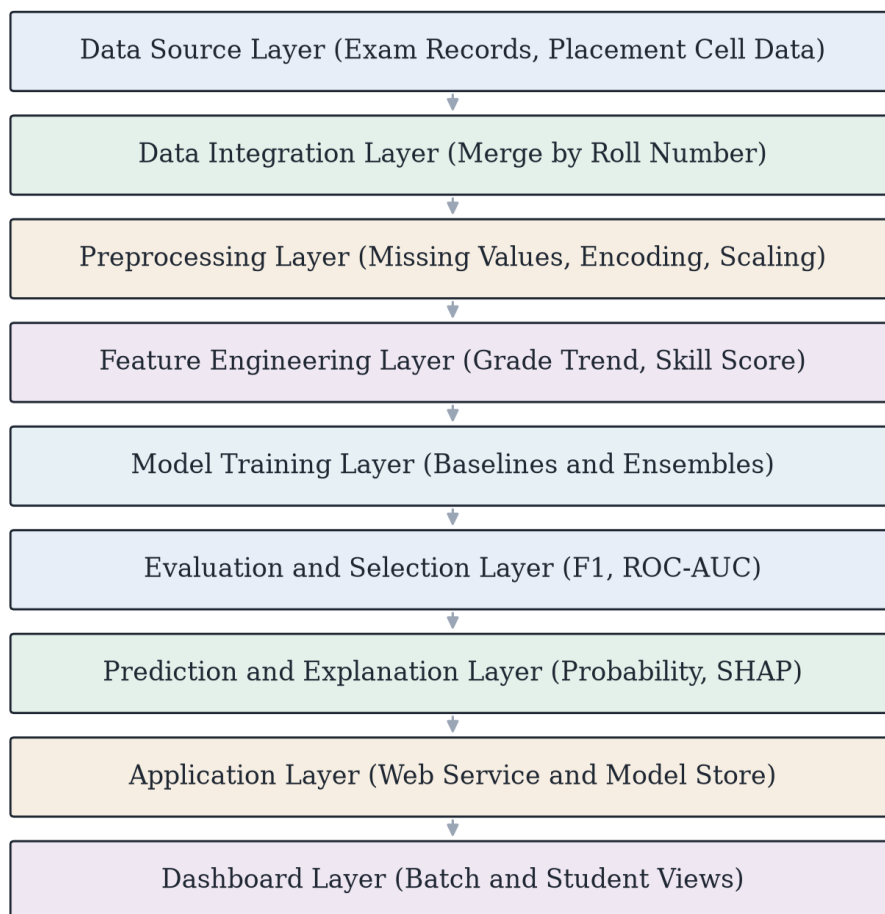


Fig. 1 System Architecture

The architecture of the placement prediction system is arranged in layers that carry student data from institutional sources to decisions made by the placement cell. The Data Source Layer holds examination records, placement cell spreadsheets, and historical recruitment outcomes. The Data Integration Layer merges these sources by roll number and resolves duplicates and format differences. The Preprocessing Layer handles missing values, encoding of categorical attributes, and scaling of numerical attributes. The Feature Engineering Layer derives grade trends, skill composites, and experience indicators and selects the most useful features. The Model Training Layer trains baseline classifiers and ensemble models and tunes their hyperparameters under cross-validation. The Evaluation and Selection Layer compares models on accuracy, precision, recall, F1, and ROC-AUC and stores the chosen pipeline. The Prediction and Explanation Layer produces placement probabilities, salary bands, and per-student explanations for the current batch. The Application Layer exposes these results through a web service, and the Dashboard Layer presents batch summaries, individual profiles, and improvement suggestions to staff and students.

V. RESULT AND OUTPUT



VI. CONCLUSION

This paper has presented an end to end campus placement prediction system that applies ensemble learning to data that institutions already maintain. By bringing together academic records, skill assessments, and experience indicators, the system gives the placement cell an early and consistent estimate of each student's likelihood of being placed. This replaces the informal judgement that is currently used and allows attention to be directed to students who need it before the recruitment season begins.

The comparison of classifiers showed that ensemble methods such as random forest, gradient boosting, and a soft-voting combination provide more accurate and more stable predictions than single models on small institutional datasets. Evaluating the models with precision, recall, F1 score, and ROC-AUC rather than accuracy alone ensures that students at risk are not overlooked simply because they form a minority of the batch.

The explanation module and the dashboard are central to the practical value of the system. Each prediction is accompanied by the factors that most influence it, so that a student can see whether aptitude, communication, practical exposure, or backlogs are holding back their profile. The salary band estimate gives students a realistic expectation, and the batch view helps the placement officer plan training sessions department by department. The system is intended to guide preparation, and final decisions about training and shortlisting remain with the placement staff.

Future work can extend the system with company-specific eligibility matching, so that students are guided towards recruiters whose criteria they meet. Integration with the learning management system would allow continuous updating of skill scores during the semester, and retraining at the end of every placement season would keep the model aligned with changing recruitment patterns. Data from several institutions, handled with appropriate privacy safeguards, could further improve the generality of the predictions.

REFERENCES

- [1] Kumar, R. D., Prudhviraj, G., Vijay, K., Kumar, P. S., & Plugmann, P. (2024). Exploring COVID-19 through intensive investigation with supervised machine learning algorithm. In *Handbook of Artificial Intelligence and Wearables* (pp. 145-158). CRC Press.
- [2] Swathi, B., Vijay, K., Sushanth Babu, M., & Dinesh Kumar, R. (2024, November). Machine Learning Techniques in Cloud Based Intrusion Detection. In *The International Conference on Artificial Intelligence and Smart Environment* (pp. 557-564). Cham: Springer Nature Switzerland.
- [3] Sv satykrishna, shirisha rang, bhargavi nalacheruve. (2024) Prospective investigation on colorectal cancer with SMOTE on machine learning Algorithm
- [4] Dr.G.Vishnu Murthy, BhargaviNalacheruve 1Professor, Department of computer Science & engineering, Anurag University, TS, India. 2Student, Department of computer Science & engineering, Anurag University, TS, India.
- [5] V. N. S. Manaswini, K. K, C. Nigam, S. S. Ali, R. Niranjana, and Suman, "Real-Time Object Detection in Drone Surveillance Using YOLOv5," in *Proc. 2025 3rd Int. Conf. IoT, Communication and Automation Technology (ICICAT)*, Gorakhpur, India, 2025, pp. 1–6, doi: 10.1109/ICICAT68430.2025.11414670.
- [6] B. Soundarya, V. N. S. Manaswini, M. Ayyakrishnan, R. D. Kumar, "Contextual Analysis of Big Data Analytics in Intelligent Transportation Frameworks," in *Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologies: From Concepts to Applications in Smart Environment*, Lecture Notes in Networks and Systems, vol. 1353, Cham: Springer, 2025, doi: 10.1007/978-3-031-88304-0_79.
- [7] R. D. Kumar, V. N. S. Manaswini, "Applications of blockchain in smart cities: detecting fake documents from land records using blockchain technology," in *Blockchain for Smart Cities*, Elsevier, 2021, pp. 105–117, doi: 10.1016/B978-0-12-824446-3.00017-X.
- [8] Tejavath Veeramma, Badarla Anil, Guguloth Ravinder, "An advanced movie recommender using collaborative filtering and sentiment analysis," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 7, no. 7, July 2025, doi: 10.56726/IRJMETS81618.
- [9] Ravi Kumar Banoth, Ramana Murthy B V, "Automatic crop recommendation system using LightGBM and decision tree machine learning models," *Journal of Machine and Computing*, vol. 5, no. 1, pp. 343, Jan. 2025, doi: 10.53759/7669/jmc202505026.
- [10] Ravi Kumar Banoth, Dr. B.V. Ramana Murthy, "Smart agriculture through IoT and machine learning for analyzing carbon footprints," in *Proc. Int. Conf. Computer Science and Communication Engineering (ICCSCE)*, Apr. 2025.

[11] Ravi Kumar Banoth, B. V. Ramana Murthy, "Soil image classification using transfer learning approach: MobileNetV2 with CNN," SN Computer Science, vol. 5, art. no. 199, 2024, doi: 10.1007/s42979-023-02500-x.