Predict Job Placement Success for Graduates Using Skill and Internship Data

Vennapusa Hashavardhan Reddy
Undergraduate Student
Indian Institute of Information Technology Sonepat
Roll no: 12311094
harshavennapusa2006@gmail.com

Nevuri Varun
Undergraduate Student
Indian Institute of Information Technology Sonepat
Roll no: 12311077
varunnevuri198@gmail.com

Abstract—In today's competitive job market, predicting a graduate's employability has become a key challenge for educational institutions. This study presents a machine learning-based framework to predict job placement success for graduates using skill, academic, and internship data. The dataset, sourced from Kaggle's Placement Prediction Dataset (10,000 entries, 12 attributes), includes features such as CGPA, projects, internships, and aptitude test scores. Multiple models—including XGBoost, LightGBM, CatBoost, and Random Forest—were trained and evaluated. An initial analysis identified mild overfitting across complex ensemble models. After applying regularization and balancing the data with SMOTE, a final comparison was performed. The best-performing and most generalizable model, Logistic Regression, achieved a test accuracy of 80.33% and a ROC AUC of 88.17%. The model demonstrated excellent generalization (a -0.67% train-test gap), proving its reliability.

I. Introduction

How well grads find work is key to judging colleges. Good grades, internships, and skill-building programs really matter when students look for jobs. For an educational institution, understanding the drivers of student employability is a core objective. In the past, placement offices and academic advisors often relied on simple heuristics or minimum grade cutoffs to guide students. This manual approach is not only time-consuming but also fails to capture the complex, multi-dimensional nature of a student's profile.

By sifting through information, machine learning (ML) reveals what influences where people land jobs—giving clues ahead of time. The goal here? To build tools which guess if a student will find employment, considering what they know alongside grades. An accurate predictive model can act as a powerful early-warning system. It can help universities identify students who may be at risk of not getting placed, allowing for timely interventions like specialized training or counseling.

II. Literature Overview

We began our research by reviewing existing work on employability prediction to understand the established methods and identify areas for improvement.

A. Traditional Classification Models

Much of the initial research in this domain focused on traditional, interpretable models. Many studies have successfully applied models like Logistic Regression and Decision Trees to academic datasets. While these models provide a clear, easy-to-understand set of rules, they often lack the predictive power to handle complex, non-linear interactions.

B. Advanced Ensemble Methods

More recent work, recognizing the limitations of simple models, has shifted towards ensemble learning. Ensemble methods, which combine the predictions of multiple "weaker" models, are now standard in high-stakes prediction tasks. Models like Random Forest, XGBoost, LightGBM, and CatBoost are highly optimized, powerful variations that are famous for winning data science competitions [1], [2].

C. The Case for Regularization and Generalization

While ensemble models are powerful, they often suffer from overfitting, where they learn the training data too well and fail to generalize to unseen data. Our initial analysis confirmed this, with most ensembles showing a 2-5% "Train-Test Gap." This study investigates if a well-regularized, simpler model like Logistic Regression can achieve comparable accuracy while providing superior generalization, especially after addressing class imbalance with techniques like SMOTE.

III. Methodology

Our project was built on a clear, data-driven methodology, primarily using the scikit-learn library for modeling and starting with a high-quality public dataset.

A. Dataset Description

We used the "Placement Prediction Dataset" available on Kaggle. This dataset was ideal for our study, containing 10,000 records of graduates with 12 relevant attributes [4]. The features provided a holistic view of a student's profile, including:

- Academic: CGPA, 10th-grade marks (SSC Marks), 12th-grade marks (HSC Marks).
- Practical: Internships, Projects, Workshops/Certifications.
- Skills: AptitudeTestScore, SoftSkillsRating.
- Other: Extracurricular Activities, Placement Training.

The target variable was Placement Status (Placed / Not Placed). The dataset was clean, with no missing values.

B. Data Pre-processing

Before modeling, we performed four key steps:

1) Encoding: Categorical features (like ExtracurricularActivities, PlacementTraining, and PlacementStatus) were label-encoded into numerical values (e.g., Yes=1, No=0; Placed=1, NotPlaced=0).
2) Feature Engineering: We created 12 new engineered features to capture potential interactions, such as an 'AcademicScore' (combining CGPA, SSC, and HSC marks) and 'PracticalScore' (combining projects and internships), resulting in 22 total predictive features.
3) Train-Test Split: We split the data into an 85% training set (8,500 samples) and a 15% test set (1,500 samples). We used a stratified split to ensure the proportion of ”Placed” and ”Not Placed” students was the same in both sets.
4) Balancing (SMOTE): To correct the class imbalance in the training data (58% 'Not Placed' vs 42% 'Placed'), we applied the Synthetic Minority Over-sampling Technique (SMOTE). This created a balanced training set of 9,866 samples.

C. Model Selection and Training

Our core hypothesis was that advanced models would provide the best performance. We first trained a suite of leading classifiers: XGBoost, LightGBM, CatBoost, Random Forest, Extra Trees, and Gradient Boosting.

After identifying mild overfitting in all of them, we implemented a second, ”Anti-Overfitting” stage. In this stage, we applied SMOTE and trained regularized versions of the tree ensembles, alongside a Logistic Regression model, to find the model with the best generalization.

IV. Experimental Setup

All experiments were performed in a Python environment, primarily using the scikit-learn, pandas, xgboost, lightgbm, and catboost libraries. Our evaluation focused on four key metrics:

- Accuracy: The overall percentage of correct predictions.
- Precision: Of those predicted as ”Placed,” how many were correct?
- Recall: Of all students who actually ”Placed,” how many did we find?
- F1-Score: The harmonic mean of Precision and Recall.

Crucially, we also analyzed the Train-Test Gap to measure overfitting and the ROC AUC score to evaluate model discrimination.

V. Result and Analysis

The systematic comparison of models after regularization yielded a clear and decisive winner.

A. Model Performance Comparison

While initial ensemble models (Stacking, Gradient Boosting) achieved 80.00% accuracy, they all showed mild overfitting (a 2-5% gap between training and test scores). The ”Anti-Overfitting” training stage, which used SMOTE and regularization, produced more reliable models. The Logistic Regression model emerged as the top performer, balancing high accuracy with perfect generalization. The results are summarized in Table I.

TABLE I
Performance Comparison of Regularized Models

Model	Test Acc. (%)	CV Mean (%)	Train-Test Gap (%)
Logistic Regression	80.33	79.68	-0.67
CatBoost (Reg.)	79.73	80.15	0.55
LightGBM (Reg.)	79.67	80.28	1.00
XGBoost (Reg.)	79.47	80.07	1.22
Gradient Boosting (Reg.)	79.13	80.22	1.41
Random Forest (Reg.)	78.73	79.24	1.35
Extra Trees (Original)	79.13	79.29	1.51

B. Analysis of Best Model (Logistic Regression)

The Logistic Regression model achieved the highest test accuracy at 80.33%. More importantly, its train-test gap was -0.67% (Train Acc: 79.66%), indicating it is exceptionally well-regularized and not overfitted to the training data. Its high ROC AUC score of 88.17% shows it is very good at discriminating between the 'Placed' and 'Not Placed' classes. A Brier Score of 0.1395 also demonstrated good calibration.

TABLE II
Detailed Metrics for Logistic Regression

Metric	Score (Class: Placed)	Score (Overall)
Accuracy	-	80.33%
ROC AUC	-	88.17%
Precision	74.24%	80.79% (Weighted)
Recall	81.43%	80.33% (Weighted)
F1-Score	77.67%	80.43% (Weighted)

C. Analysis of Confusion Matrix

To better understand the 80.33% accuracy, we plotted a confusion matrix for the Logistic Regression model, as shown in the placeholder Fig. 1.

The matrix analysis from the test set (1,500 samples) revealed:

- True Negatives (TN): 692 (Correctly predicted 'Not Placed')

Image Placeholder
Your Plotted Confusion Matrix goes here

Fig. 1. Plotted Confusion Matrix for the Logistic Regression Model. This visualizes the True Positives, True Negatives, False Positives, and False Negatives.

- True Positives (TP): 513 (Correctly predicted 'Placed')
- False Positives (FP): 178 (Incorrectly predicted 'Placed')
- False Negatives (FN): 117 (Incorrectly predicted 'Not Placed')

The model demonstrated a strong ability to identify students who will be placed (Recall of 81.43%), which is a critical goal for an early-warning system.

VI. Discussion

Our study successfully produced a model that can predict student placement with high, reliable accuracy. The 80.33% accuracy achieved by the Logistic Regression model is a strong result, demonstrating it is a highly effective tool for this problem.

A. Model Choice and Justification

The most compelling finding was the superiority of a well-regularized Logistic Regression model after applying SMOTE. While complex ensemble models like XGBoost and Stacking are popular, our analysis showed they were prone to mild overfitting on this dataset.

The simpler, interpretable Logistic Regression model, when trained on balanced data, achieved a higher test accuracy and, critically, a near-zero generalization gap. This suggests that for this dataset, controlling overfitting and addressing class imbalance were more important than modeling complex non-linear interactions. The model's interpretability is an added benefit, allowing administrators to understand which features contribute most to the prediction.

B. Limitations

This study, while successful, has some limitations.

- Dataset Source: The Kaggle dataset is anonymized and simulated. We do not know the specifics of the university or the companies.
- "Soft Skills" Feature: The SoftSkillsRating feature is a single, subjective score. A more granular dataset would be more predictive.
- No NLP: The Projects feature is just a number. A model that could analyze the text descriptions of those projects using NLP would be far more powerful.

Image Placeholder
Your Feature Importance plot goes here

Fig. 2. Feature Importance Plot. (e.g., from Logistic Regression coefficients or a tree-based model).

VII. Conclusion

This project successfully demonstrated that machine learning can be a powerful tool for predicting graduate employability. We systematically compared multiple ensemble models, identified overfitting, and applied regularization and balancing techniques. We identified the Logistic Regression classifier as the most effective and robust, achieving a final test accuracy of 80.33% with a ROC AUC of 88.17%.

Our analysis provided a key insight: a simpler, interpretable model, when properly regularized and trained on balanced data, can outperform more complex "black-box" models by offering superior generalization. Our work provides a strong foundation for an early-warning system that could help educational institutions support their students and improve placement outcomes.

VIII. Future Scope

While our model is a strong proof-of-concept, there are many exciting ways this work could be extended.

- Real-World Data: The next logical step would be to apply this methodology to a real (non-simulated) dataset from our own university's placement cell.
- Explainable AI (XAI): We could use tools like SHAP or LIME to explain why the model made a specific prediction for an individual student.

- Model Deployment: The ultimate goal would be to deploy this model as a simple web tool for students and counselors to use.
- NLP Integration: We could enhance the model by incorporating NLP to analyze text from student resumes, such as project descriptions and skill summaries.

References

[1] T. Chen and C. Guestrin, ”XGBoost: A Scalable Tree Boosting System,” in Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 2016, pp. 785-794.

[2] G. Ke et al., ”LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” in Proc. Advances in Neural Information Processing Systems 30 (NIPS), 2017.

[3] P. Geurts, D. Ernst, and L. Wehenkel, ”Extremely Randomized Trees,” Machine Learning, vol. 63, no. 1, pp. 3-42, 2006.

[4] R. Kumbhar, ”Placement Prediction Dataset,” Kaggle, 2024. [Online]. Available: https://www.kaggle.com/datasets/ruchikakumbhar/placement-prediction-dataset

[5] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, ”CatBoost: unbiased boosting with categorical features,” in Proc. Advances in Neural Information Processing Systems 31 (NIPS), 2018.

[6] L. Breiman, ”Random forests,” Machine learning, vol. 45, no. 1, pp. 5-32, 2001.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, ”SMOTE: Synthetic Minority Over-sampling Technique,” Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

[8] D. W. Hosmer Jr., S. A. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed. John Wiley & Sons, 2013.

[9] S. Ahuja and S. S. Panigrahi, ”A comprehensive review of machine learning techniques for student placement prediction,” Applied System Innovation, vol. 6, no. 1, 2023.

[10] P. K. Deheri and R. K. Bhatt, ”A comparative study of machine learning algorithms for predicting student employability,” in Proc. International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2021.

[11] M. S. R. Basha and S. S. S. N. Varma, ”Student placement prediction using logistic regression and decision trees,” in Proc. 3rd International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020.