Placement Prediction System Using Machine Learning and Ensemble Learning Techniques

Prof. Manasi Ghamande, Kaustubh Ghamande, Piyush Mishra, Aum Mishra, Vedant Mehar, Mayur Chikhale, and Harshal Marathe

Department of Information Technology, Vishwakarma Institute Of Technology, Pune , India

manasi.ghamande@vit.edu , kaustubhghamande@gmail.com, piyush.mishra24@vit.edu, aum.mishra24@vit.edu, vedant.mehar24@vit.edu, mayur.chikhale24@vit.edu, harshal.marathe24@vit.edu

Abstract

This project offers an end-to-end placement prediction and preparation system based on machine learning and AI. Employing ensemble models for one datasets, the system makes accurate student placement predictions of up to 98.75%. It offers major features such as a customized recommendation module, an AI chatbot, real-time code analysis, and self-assessment of speaking skills. Through studying academic, technical, and communication profiles, the platform delivers customized learning pathways and real-time feedback, enabling students to improve employability. Through data-driven insights, the system strengthens institutions and assists students with end-to-end career readiness solutions. The platform also offers interactive learning materials, and its real-time analytics enable students to monitor progress while enhancing crucial skills such as communication and coding. Through ongoing feedback, the system promotes proactive career development, keeping both students and institutions in step with industry requirements.

Introduction

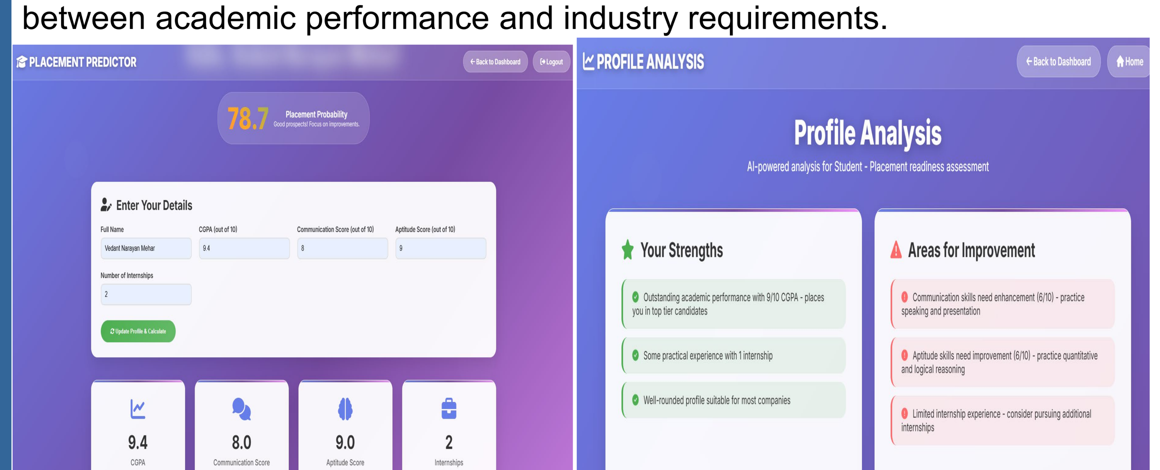
The project is an AI and machine learning combined placement prediction and preparation system. On a single dataset, it predicts student placement outcomes with 98.75% accuracy using ensemble models. The system includes a personalized recommendation module, AI chatbot, live code review, and automated speaking skill assessment. Using academic, technical, and communications profile analysis, the platform provides tailored learning streams and real-time feedback, developing the employability of students. The system provides data-driven insights for institutions and career readiness end-to-end solutions for students.

Project Objectives

To develop an AI-powered web application that accurately predicts student placement outcomes using machine learning techniques and provides personalized recommendations, coding practice, and communication skill evaluation to enhance career readiness and support institutional decision-making.

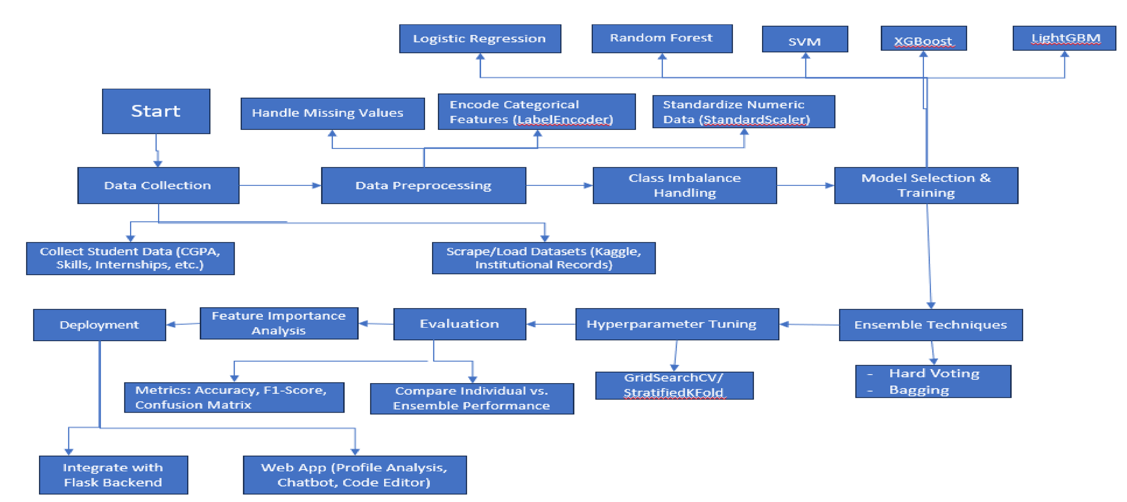
Problem Statement

Students are still not clear about placement because of the lack of one-on-one counseling and instantaneous measurement of performance. The colleges also struggle to determine readiness and plan training sessions in an optimal manner. There is still a necessity for a data-driven model that can predict placement outcomes, provide one-on-one means of improvement, and provide interactive tools to fill the knowledge gap between academic performance and industry requirements.



Process Layout

•**1. Data Foundation**
•The workflow begins with aggregating student academic records (CGPA, skills, internships) and filling with institutional/Kaggle datasets. Thorough preprocessing addresses missing values, encodes categories, and normalizes features to prepare data for modeling.
•**Model Building**
•Several algorithms (Logistic Regression, Random Forest, SVM, XGBoost, LightGBM) are trained and blended with ensemble techniques. Hyperparameter tuning with GridSearchCV and cross-validation optimizes predictive accuracy for placement results.
•**3. Performance Validation**
•Models are rigorously tested using accuracy, F1-score, and confusion matrices. Relative comparison of individual and ensemble models determines the most dependable method for student placement predictions.
•**4. System Implementation**
•The chosen model is implemented through a Flask web application with profile analysis resources, an advisory chatbot, and coding evaluation capabilities - forming a useful platform for career counseling and placement prediction.
•**5. Sustainable Evolution**
•Built-in feedback mechanisms and periodic retraining ensure the system adapts to changing job markets and educational trends, maintaining long-term relevance through continuous improvement cycles.



Methodology

Supervised machine learning pipeline with sophisticated preprocessing and model training, was performed in conjunction with ensemble learning to build placement prediction model. The following steps outlines the methodology:

1. Data Acquisition & Feature Selection

The student records data set was structured and had features like:

- **Academic Performance**: CGPA, 10th and 12th marks
- **Demographics & Background**: Gender, Stream, Hostel status
- **Skills & Experience**: Communication skill rating, Aptitude score, Technical course completion, Internship experience
- **Other Factors**: History of academic backlogs

The binary target variable classified students as either placed or unplaced.

2. Data Preprocessing

An aggressive preprocessing approach was employed to guarantee model dependability:

- **Missing Values**: Checked and handled to maintain data integrity
- **Categorical Encoding**: LabelEncoder was employed to convert categorical data (e.g., gender, stream) into numerical format
- **Feature Scaling**: All numeric features were standardized using `StandardScaler` to improve model convergence
- **Train-Test Split**: The dataset was split into 80% training and 20% testing sets to evaluate generalization

3. Model Development

Several machine learning models were executed using Python's Scikit-learn library. Training involved:

- **Baseline Models**: Logistic Regression, Support Vector Machine (RBF Kernel), K-Nearest Neighbors, Decision Tree, Naive Bayes
- **Advanced Models**: Random Forest, Gradient Boosting (tuned), Bagging Classifier

•**Ensemble Techniques**: A **Hard Voting Classifier** was constructed by combining predictions from four high-performing classifiers to improve predictive robustness

Hyperparameter tuning was performed using `GridSearchCV` to optimize each model's performance.

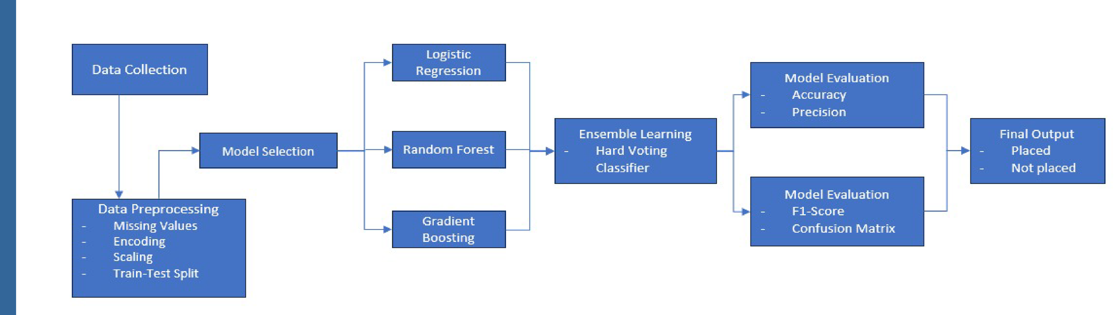


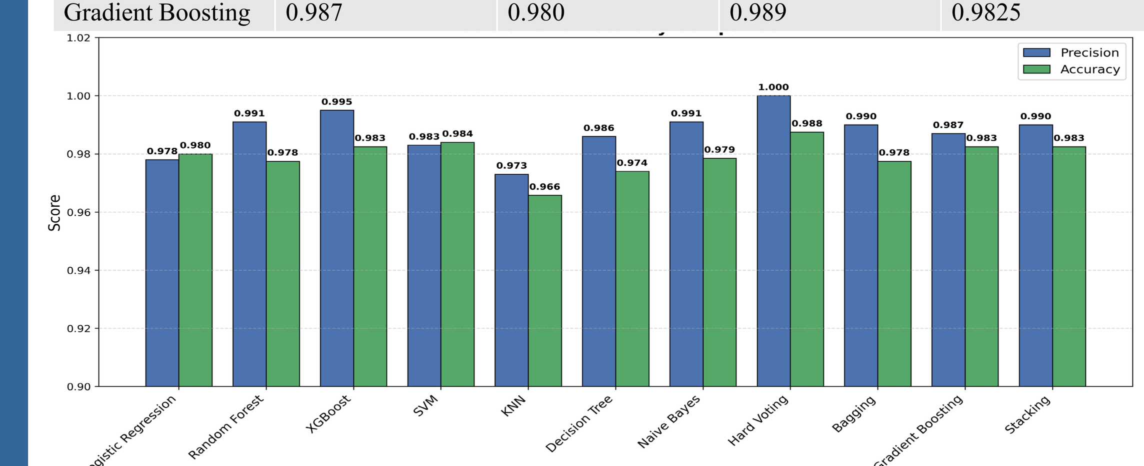
Figure 1. Placement Prediction System Workflow Using Ensemble Machine Learning Models

Model Evaluation Summary

We calculated metrics like precision, accuracy, recall and f1 score along with confusion matrix to evaluate and compare various machine learning models for student placement prediction. These metrics offer a comprehensive picture of the ability of each model to identity placement outcomes. Here, we have visual comparisons and discussion.

Table 1: Evaluation Metrics Summary

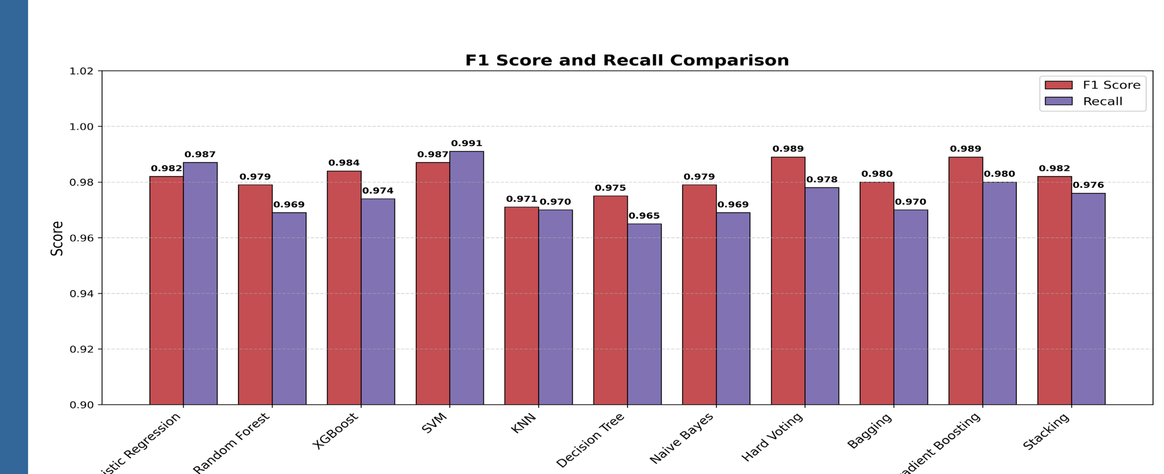
Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	0.978	0.987	0.982	0.980
Random Forest	0.991	0.969	0.979	0.9775
XGBoost	0.995	0.974	0.984	0.9825
SVM (RBF Kernel)	0.983	0.991	0.987	0.984
KNN	0.973	0.970	0.971	0.96575
Decision Tree	0.986	0.965	0.975	0.974
Naive Bayes	0.991	0.969	0.979	0.9785
Hard Voting Classifier	1.000	0.978	0.989	0.9875
Gradient Boosting	0.987	0.980	0.989	0.9825



The bar graph compares the **Precision** and **Accuracy** values across 11 ML models.

- The Hard Voting Classifier was the best performer achieving **100% precision** and **98.75% accuracy** meaning it made no false positive predictions and classified almost all students correctly.
- The precision of **Gradient Boosting** and XGBoost were respectively **0.987** and **0.995**, while their accuracy scores were **98.25% and 98.25%.**
- The **least accurate** algorithm was KNN with **96.57% accuracy**, indicating its sensitivity to feature mapping and noise.

The graph highlights **Hard Voting** as the best in **ensemble-based models** for maintaining essential precision in student placement predictions.



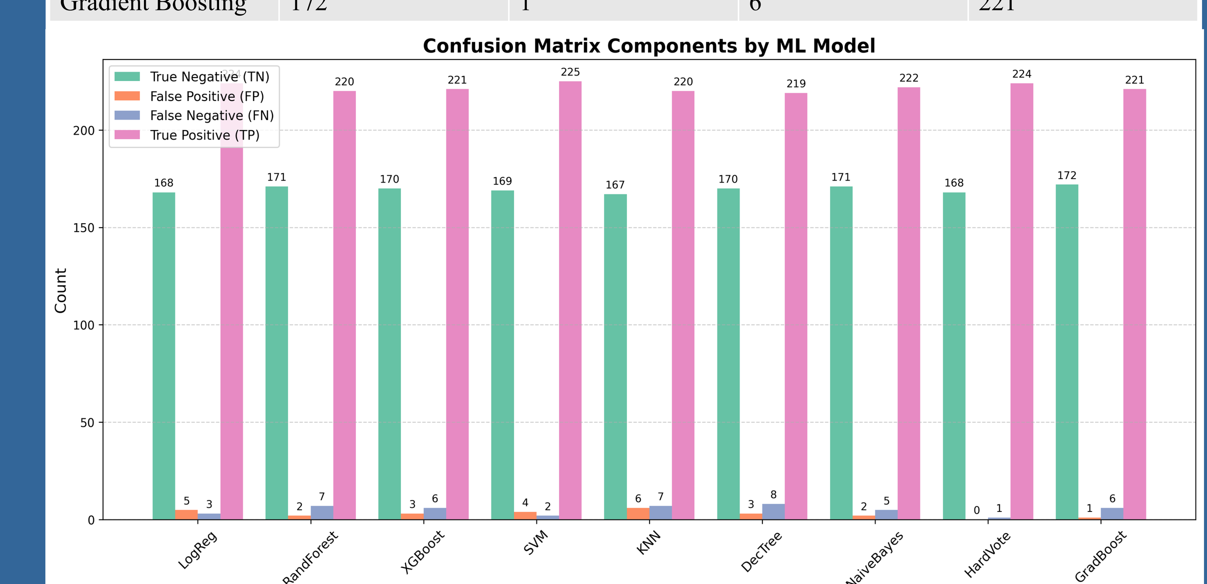
This graph depicts the **Recall** and **F1-Score**, which are essential for knowing the model's performance in identifying placed students.

- The SVM model had the **highest recall (0.991)** and **F1-score (0.987)** indicating that it almost correctly classified all placed students with good precision.
- The **F1-score** for Gradient Boosting was **0.989** and the **Recall** was **0.980**.
- In contrast, **Decision Tree** and **KNN** showed slightly lower recall (0.965 and 0.970), suggesting they missed more actual placements (false negatives).

According to these findings, **SVM and Gradient Boosting** types appropriately balance both false positives and false negatives and are therefore suitable for applications that require precision and coverage.

Confusion Matrix Comparison

Model	True Negative (TN)	False Positive (FP)	False Negative (FN)	True Positive (TP)
Logistic Regression	168	5	3	224
Random Forest	171	2	7	220
XGBoost	170	3	6	221
SVM (RBF Kernel)	169	4	2	225
KNN	167	6	7	220
Decision Tree	170	3	8	219
Naive Bayes	171	2	5	222
Hard Voting Classifier	168	0	1	224
Gradient Boosting	172	1	6	221



The grid of the confusion matrix visualizes the **True Positives (TP)**, **True Negatives (TN)**, **False Positives (FP)** and **False Negatives (FN)** of every model:.

- The **Hard Voting Classifier** predicted **224 placed students (TP)** and **168 non-placed (TN)**. It also generated **0 FP** and only **1 FN.**
- SVM identified **225 true positives (TP)**, **169 true negatives (TN)**, and only **2** false negatives (FN). Hence, SVM shows consistency in identifying the real placements.
- However, KNN has **7 FN** and **6 FP.** This means it has an inability to discriminate between the two classes in our dataset.

Conclusion

•In summary, the implementation of a data-driven approach using machine learning models has yielded significant improvements in forecasting campus placement outcomes through the Placement Predictor platform.

•By integrating advanced algorithms and ensemble techniques within a secure and interactive coding environment, we have enhanced both technical learning and placement preparedness among students in a measurable and impactful way.

•Beyond delivering accurate and personalized predictions, this initiative fosters a culture of continuous learning and self-improvement, positioning the institution to better meet evolving industry demands and empower students toward greater career success.

References

Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5–32.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.