Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Next Hire: A Data-Driven Student Employability Prediction System

Lathika. R
Vel tech high tech Dr.Rangarajan
Dr. Sakunthala Engineering College

Madhubala. M
Vel tech high tech Dr. Rangarajan
Dr. Sakunthala Engineering College

Lokesh. P
Vel tech high tech Dr.Rangarajan
Dr. Sakunthala Engineering College

Christopher Joseph. L
Assistant professor
Department of Information Technology
Vel tech high tech Dr.Rangarajan
Dr. Sakunthala Engineering College

Abstract - ***Nowadays, one of the biggest dilemmas faced by institutions of higher learning is how to improve the placement rate of students. The selection rate and popularity of a higher institution essentially depend on its placement results; hence, improving the placement department has become a keynecessity. Landing a job offer prior to completing a program in a higher institution is one of a student's key aims.***

Educational institutions require effective technological solutions that may assist the decision-making process when developing placement solution systems. These may facilitate students to assess their current position in terms of improvements to be made for effective placement transfer.

In the project, Next Hire, a model is created to predict the student's placement by using the Random Forest machine learning algorithm. This predicts the probability of placement of the students. Therefore, this model will help the placement cell of the institution in selecting the candidates on whom they need to concentrate and train. The parameters related to academics, technology, and soft skills help the institution guide the student accordingly to increase the performance of the placement.

Keywords- Machine Learning, Data science, Hiring Probability, Candidate evaluation, Predictive analysis, decision support system

1.INTRODUCTION

The “Next Hire” project is a data science-based system that predicts the potential for hiring according to specific measurable attributes possessed by applicants. The ever-changing fast-paced environment within the recruitment industry generates a substantial number of applications for a given advert, creating a trend where the hiring process is becoming more time-consuming and subjective for HR professionals within an organization.

Next Hire tackles this issue by relying on algorithms and machine learning, where it evaluates candidate profiles and predicts their selection possibilities based on their coding abilities, communication abilities, academic record (CGPA), and other activities, which are prominent determining factors regarding their employability. The data is also preprocessed to remove any inconsistencies and unknown values, resulting in accuracy and reliability in their results.

This method not only removes any bias on the part of humans but also reduces time and resource usage when recruiting. Another benefit it has for applicants is identifying areas for improvement to increase their chances of employment. In sum, it can be said that “Next Hire” proves how data science and machine learning can be practically employed for managing human resources through the effective processing of recruitment data to support smarter, more balanced, and more effective recruitment practices.

The objective of the work is data collection, analysis, and prediction whether. In this way, it is ensured that students get placed or not, making the entire process efficient and easy. The selection procedure is of utmost importance to the success of productivity in an organization. But the conventional recruitment processes undertaken in organizations tend to be inefficient, time-consuming, and subjective. Moreover, the number of individuals seeking employment is escalating consistently.

Traditional Recruitment Process in Organizations factors A continues to rise, particularly within the technological and corporate worlds, for the identification of the candidate quickly and accurately has become increasingly difficult. The issue that the "Next Hire" resolves have immense relevance because it deals with in undertaking data science and machine learning to tackle a real-world human resource issue: recruiting the best possible candidates. In this regard, by adopting measurable parameters such as programming skills, communication skills, CGPA, and co-curricular activities, this project brings in a data-driven solution which reduces human bias and improves the accuracy of decision-making. Additionally, this problem is of considerable not only in

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

academic as well as real-life situations because there are very rarely any projects which can professional settings. The text shows how analytics can change a traditional HR process into intelligent, automated systems.

An integral part of the recruitment process is decision-making. This concept is both academically and practically applicable. However, a lot of times, academic projects fail to address the requirements of the professional workplace. Bridging this gap is where predictive analytics has added value by modernizing typical human resource processes. The primary aim of this initiative is to formulate a mechanism to assess the readiness for employment of graduates by employing machine learning. Educational institutions would appreciate this system in that it would permit the improvement of placement training and career counseling services. The initiative, however, endeavors to determine students who may be deficient in certain critical skills relating to employability. The system is designed to assist educational institutions identify training and support deficiencies in graduates to address issues relating to high levels of unemployment.

LITERATURE REVIEW:

This part analyses existing studies concerning data-driven recruitment and hiring predictive systems. The aim of this analysis is to comprehend the existing contributions in this domain, elucidate the existing research gap, and establish a groundwork for the anticipated Next Hire system, which is the primary focus of this analysis.

Reddy et al. [1] developed a prediction model based on machine learning to estimate recruitment outcomes based on candidate's qualifications, which include academic score, technical, communication, certification, and personal skill. They used supervised learning techniques such as Logistic Regression, Decision Trees, and Support Vector Machines (SVM). They also stressed the importance of feature selection to enhance prediction accuracy and to avoid overfitting. Though their study concentrated in university placements, the model can also be In this paper,

Dogiparthy et al. [2] attempted to solve issues related to traditional recruitment systems, such as bias, inefficiency and subjectivity, through a machine learning-based candidate selection model utilizing Logistic Regression, Random Forest and CatBoost as well as feature selection methods to eliminate less relevant information to enhance model interpretability. The findings pointed out that recruitment decisions and the process as a whole can benefit from machine learning with regards to efficiency and accuracy. This is a further endorsement of the approach taken in the proposed Next Hire recruitment system.

Yassine and Said [3] developed an intelligent recruitment prediction model using an artificial neural network and gradient descent optimization to assess programming and communication skills, CGPA, as well as co-curricular activities and other. The study shows the potential of machine learning to uncover hidden relationships and to yield valid recruitment predictive analytics.

Fabris et al. [4]. The study explored the extent to which machine learning algorithms may, inadvertently, be biased by gender, socio-economic, and other factors extended to practical recruitment scenarios.

II. EXISTING SYSTEM:

The present system largely depends on student employability assessment through the review of traditional evaluation methods such as reviewing academic scores and semester grades, conducting aptitude and technical tests before placement drives, collecting feedback from faculty and mentors, and observing participation in soft-skill development programs or placement training sessions. Institutions analyze previous placement results to estimate employability trends.

Though these methods give partial insight into the performance of the student, they are at best qualitative and subjective. In most cases, the evaluation depends on human judgment and misses out on all parameters leading to employability. Such assessment is normally carried out at the tail-end of the academic programs or just before the placement drives. By the time a particular student is found to be in need of extra assistance or training, it is too late, and not much scope remains for any changes.

Another major limitation with the existing system is the absence of a single integrated mechanism that would not only consistently track and analyze students' data but also monitor the changes thereof over time. Without predictive analytics, these institutions miss opportunities to provide timely skill development, mentoring, or counseling for those students who are at risk of poor placement outcomes.

III.PROPOSED SYSTEM:

"Next Hire" system proposes a Data Science solution to a placement prediction system that resolves the deficiencies that exist in the existing system. The system proposed not only computes a student's probability of placement, as done in the current system, but also provides learning enhancement tips in case the probability is less.

After determining the possibility of placement, it also determines the impact of varied attributes such as academic skills, technical skills, communication skills, and extracurricular activities on a student in their final result for determining the possibility of placements. This will enable the system to provide suggestions to the student on how they can improve their skills for their placements. The constant evaluation of the data also enables the determination of students who are not performing well in their placements

A. **Data-Driven Insights**: Relies on data that can predict employment outcomes.

B. Early Identification: It identifies pupils who need further training or direction much earlier than a placement initiative.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

C. Objective Evaluation: Reduces human judgment and employs consistent and standardized evaluations.

D. Improved Placement Planning: Helps in planning training and placement in institutions based on analytical insights.

E. Continuous performance tracking: Enables continuous tracking of student performance and identification of improvement points.

F. Bridging Academia and Industry: This goal is to align student skill development with that anticipated by industry and the job market.

A.SYSTEM REQUIREMENTS:

To ensure smooth performance and reliable functionality, the Next Hire – A Data-Driven Recruitment Prediction System requires certain software and hardware support. The system can run on Linux-based servers such as Ubuntu or CentOS, as well as on Windows Server, while users can access the application from Windows, macOS, or Linux through a web browser, and from Android or iOS devices on synchronization.

The system must provide a facility for users to access the application in a secure manner. The system shall be able to distinguish between normal users and administrators, students being the normal users in this User input needs to be validated before it can be processed properly.

The system must be able to handle student data entry for information such as: Secondary education percentage However, Higher secondary education percentage Degree percentage Status of work experience Test Score Employability/Aptitude (if recorded) The system should ensure that no incomplete or invalid data is submitted.

ATTRIBUTE	DESCRIPTION
SSLC percentage	Secondary school academic score
HSC percentage	Higher secondary academic score
CGPA	Cumulative grade average
Communication Skills	Communication score
Technical skills	Technical knowledge, coding ability
Internship experience	No. of interns completed
Certifications	No. of professional certificates
Hackathon Participation	Participation in hackathons

Table 1: Data Attribution table

B. IMPLEMENTATION

Applying the predictive model of analysis is done in a systematic and structured way. It is designed to work with student data by applying machine learning and produce employability prediction results in terms of the probability score. This phase of implementation is significant in turning the conceptual design of the project into a functional one.

i)Data Collection:

The initial step for implementation requires gathering student data from trusted sources like organizational databases or public datasets. The dataset consists of parameters like academic performance, technical skills, soft skills, internship experience, total projects accomplished, and quality of the resume. The final parameter is the target, which denotes the employability aspect of the individual, i.e., the student. Data Preprocessing missing values exist in raw data, as well as inconsistencies and categorical variables that cannot be used directly for training a model

The first and most essential step in the methodology is data collection, which forms the foundation of the predictive system. In this phase, relevant data about candidates is gathered from academic records, online recruitment datasets, or institutional placement databases. The dataset includes both quantitative and qualitative parameters such as programming skills score, communication ability rating, academic performance (CGPA), participation in extracurricular activities, and hiring status (hired/not hired).

ii)Data processing:

Data preprocessing is done to clean the data. Missing values are dealt with using an appropriate method like removal or imputation. The categorical variables, like internship offers and level variables, are converted into numeric variables. Feature scaling is done to give an equal contribution by numeric variables.

This also includes

1.Handling missing values

2.Encoding categorical variables

3.Feature scaling

The system is implemented using Python with machine learning libraries such as Scikit-learn. A Flask-based backend is used to handle user inputs and model inference. The processed data and prediction results are stored in a MySQL database. The user interface allows students to input their academic and skill details and view prediction results along with improvement suggestions.

Once data is collected, it undergoes preprocessing to prepare it for analysis and model training. Real-world datasets often contain missing values,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

redundant entries, and noise that can mislead the model during training. Data preprocessing is carried out to clean, transform, and standardize the dataset.

During this stage, missing data values are replaced using statistical imputation methods such as mean or median substitution. Duplicate entries are removed to avoid bias, and data normalization is performed to bring all attributes into a comparable numerical range, typically between 0 and 1. Categorical features, such as the department or specialization, are converted into numeric codes using encoding techniques like Label Encoding or One-Hot Encoding.

Additionally, the dataset is divided into two parts: a training set (80%) and a testing set (20%). The training set is used to build and optimize the machine learning model, while the testing set is reserved for evaluating the model's performance on unseen data. This ensures that the model generalizes well and performs consistently when predicting new candidate outcomes.

iii)Model Selection and Training

A Random Forest classifier is employed as the primary machine learning model due to its robustness, ability to handle non-linear relationships, and resistance to overfitting. The model is trained using an ensemble of 300 decision trees. Each tree is constructed using a random subset of features and samples, enhancing generalization capability. The dataset is split into training and testing sets using an 80:20 ratio.

After preprocessing, the next phase involves feature selection and engineering, which plays a vital role in enhancing the model's accuracy and efficiency. Not all collected features contribute equally to the final prediction; therefore, identifying the most influential attributes is essential.

Techniques such as correlation analysis, mutual information, and feature importance ranking are used to assess how strongly each variable influences the hiring decision. Features that show minimal correlation or add redundancy are eliminated to reduce model complexity and improve interpretability. For example, programming skills and CGPA may exhibit a stronger relationship with hiring probability compared to less significant factors.

Feature engineering is also applied to create new, meaningful variables by combining existing ones. For instance, an "overall skill index" can be computed by combining programming and communication scores. This step ensures that the dataset is well-structured and optimized for better prediction performance.

iv)Placement Probability Estimation

The trained Random Forest model outputs class probabilities rather than only binary placement results. These probabilities are interpreted as the placement likelihood for each student. The placement probability is computed as the average prediction confidence across all decision trees in the forest. This probability is further used to categorize students into high, medium, and low placement readiness groups with probability percentage

V)Salary Prediction

For placed students, a regression-based Random Forest model is used to estimate the expected salary range. The salary prediction model uses the same feature set with salary as the target variable. The predicted salary provides an approximate estimation based on historical placement trends and student skill profiles.

vi)Improvement Suggestion Module

Based on feature importance and individual student performance, a suggestion module is implemented. This module analyzes weak attributes and generates personalized recommendations such as improving communication skills, increasing coding practice, participating in hackathons, or completing internships. These suggestions aim to enhance the student's placement probability

vii)Training and Testing of the Model:

The dataset is then split into training and testing sets after preprocessing. The steps further involve the application of machine learning algorithms like Random Forest to train the predictive model using the training data. The testing data are used to evaluate the performance of the trained model in terms of accuracy and reliability. Performance metrics, including the accuracy score and confusion matrix, are computed to assess the effectiveness of the model.

viii) **Prediction and Output Generation:**

After training the model and validation, the latter is used for the classification of employability readiness on new student data. It will output a classification result in the form of a yes or no whether the student is employable or not with a probability percentage representing readiness confidence. This would help institutions and students understand the exact level at which employability stands and thus undertake necessary corrective actions

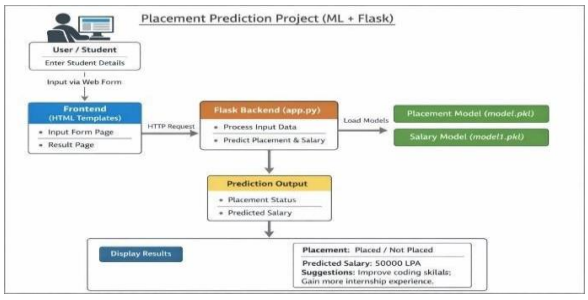


Fig:1 Block Diagram

The placement prediction model is a machine leaning model where a student can use the model to predict their probability to get placed the student login the page and here, we use HTML as the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

frontend design input form the student is asked to enter some academic details include HSC, SSLC percentage, no of internships, certifications, skills, participation in hackathons, etc. After receiving the details, the model like placement model and salary model will be loaded inside the flask backend comparing the values entered by the student it will predict the probability that the student gets placed and also give the predicted salary if the student lacks probability percentage it also gives suggestions for the student improvement. The system is implemented using Python with machine learning libraries such as Scikit-learn. A Flask-based backend is used to handle user inputs and model inference. The processed data and prediction results are stored in a MySQL database. The user interface allows students to input their academic and skill details and view prediction results along with improvement suggestions.

C.PLACEMENT INSIGHTS:

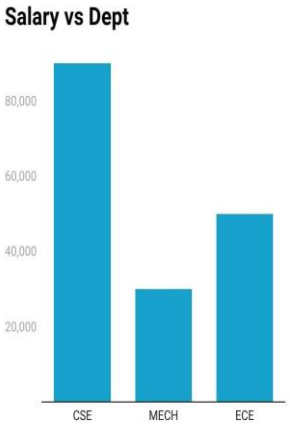


Fig 2: Department wise analysis of average salary

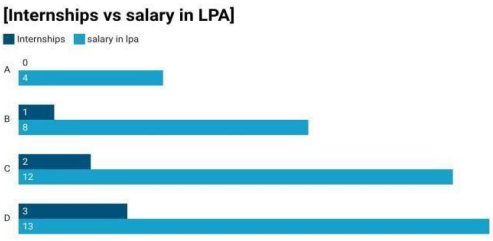


Fig:3 Part of Internships affects salary

Fig 4: Impact of programming on salary

Fig :5 No. of projects affects salary

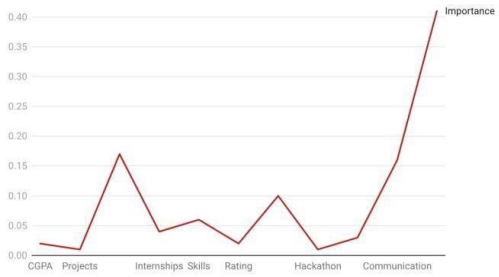


Fig:6 Probability of skills

IV. METHODOLOGY

Fig 7: Architecture diagram

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

The methodological framework in this research project encompasses a systematic and structured process for building a precise and trustworthy predictive analysis model for estimating the employability readiness of students. The overall methodological framework is segmented into various stages for ensuring the exactness, correctness, and effectiveness of the process.

i) **Data acquisition:**

Collection of authentic sources that include institutional databases, official academic records, surveys, or publicly available datasets is done in the initial phase. Academic performance, technical skills, soft skills, internship experience, numbers of projects completed, certification, hackathon participation, and scores of resume evaluation are some of the collected attributes. These parameters act as a basis for consideration in readiness for employability.

ii)Data Preprocessing:

Data preprocessing is a key component in machine learning for improving the quality of data. Initially, the dataset requires preprocessing since the data is incomplete and represented in categorical form that cannot be further used to build the model. Missing values in data can be treated by appropriate methods for removal or imputation. Categorical data, which includes skill level and internship, needs to be represented in a number format by one-hot encoding. Min-Max Scaling is conducted for equal weighting of the quantitative attributes and is given by:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}$$

This ensures that features with large ranges do not dominate the process of learning.

In this step, the missing information in the data is treated using techniques such as removal or imputation. Category variables such as internship and skill levels are transformed into numerical variables using encoding schemes. Feature scaling is done to ensure that numerical variables contribute equally.

iii)Feature selection:

This stage is all about detecting the most pertinent attributes that play a significant role in employability readiness. Correlation analysis, along with the feature importance ranking from the Random Forest model, is used to remove redundant attributes. The most important attributes, including CGPA, internships, communication skills, coding skills, and hackathons, are given more priority. Apart from this, feature engineering is also done. Here, new attributes are created using the existing attributes. . Relevant features that significantly impact employability readiness are identified and selected. Redundant or irrelevant attributes are removed to improve model efficiency. In some cases, new features are derived by combining existing attributes to enhance predictive performance.

iv)Model selection:

A Random Forest classifier is chosen due to its ruggedness, handling capability of non-linear relationships, and effectiveness in preventing overfitting. The classifier is trained on an ensemble of 300 decision trees. A split of 80:20 is used to divide the dataset into training and testing subsets. During the training phase, the classifier is exposed to patterns and relationships among the characteristics of the students and the employability outcomes. Machine learning techniques such as Logistic Regression and Random Forest are applied to build the predictive model. To assess performance, the dataset is split into training and testing subsets. During training, the model identifies patterns and relationships between student-related factors and their employability outcomes

V)Model evaluation and training:

The trained model will then be tested using the test dataset in order to determine the efficiency and reliability of the model. Accuracy, precision, recall, F1 score, and confusion matrix will be calculated. These steps will help in minimizing the errors in the predictions and in the efficiency of the model. Various performance measures, including accuracy, precision, recall, F1-score, and the confusion matrix, are used to analyze how well the model performs. This evaluation process helps ensure that the predictions are reliable and that errors are kept to a minimum.

vi)Result and prediction:

In the final stage, the verified model is employed to predict new profiles of students regarding their employability readiness. The final result provided by the system entails the binary outcome for classifying students (employable or not employable) with a probability score for the confidence of readiness (85% readiness, for example).

The final phase involves result generation and visualization, where the system presents the predicted hiring probability in a clear and interpretable manner. The output is typically displayed as a percentage (e.g., “Candidate has an 85% probability of being hired”). Additionally, graphical visualizations such as bar charts, probability graphs, or ranking indicators are used to enhance understanding and provide a better comparative view among candidates.

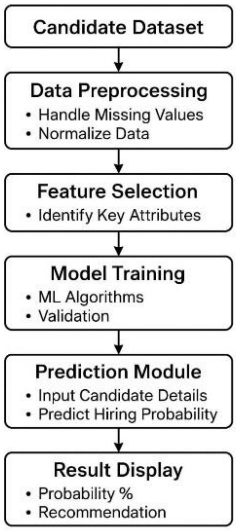
Visualization not only improves the user experience but also helps recruiters quickly identify strong candidates

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

and areas of improvement for others. This stage ensures that the results of complex machine learning computations are communicated in an accessible and meaningful way.

vii)Language Used

The programming language primarily used in the above steps is Python project. The choice of the language is based on the simplicity of the syntax, readability, and overall support for data science and machine learning tasks. Python offers extremely flexible and sophisticated tools. libraries such as Pandas, NumPy, Scikit-learn, Matplotlib, which facilitate data processing, model implementation and result visualization. This is because of flexibility and efficiency; Python is well suited for imputation. Category variables such as internship and skill levels are transformed into numerical variables using encoding schemes. Feature scaling is done to ensure that numerical variables contribute equally.



. Fig:8 User case diagram

FEATURE	TYPE	ENCODING METHOD
Communication skills	Categorical	Low=0 Medium=1 High=2
Technical skills	Categorical	Low=0 Medium=1 High=2
Hackathon participation	Binary	Yes=1 No=2
Placement status	binary	Placed=1 Not placed=0

Table 2: Feature table

WORKING PROCESS:

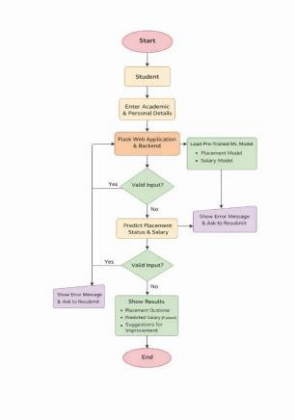


Fig :9 Working process

D.MODEL TRAINING:

Two random forest are trained in this project one is to predict the probability of the student getting placed and another one is to predict the salary of the student The training process involves:

1. Splitting the data into training and testing sets
2. Initializing the Random Forest classifiers
3. Training models on the training sets
4. Fine-tuning hyperparameters using techniques like grid search or random search

a)Dataset Preparation and Splitting

The collected and preprocessed dataset is divided into two subsets: a training set and a testing set. The training set contains 80% of the total records and is used to learn the underlying patterns between student attributes and placement outcomes. The remaining 20% of the data is reserved for testing purposes to evaluate the model's generalization capability on unseen data. This data splitting strategy ensures unbiased performance assessment.

b)Model Training Process

The Random Forest algorithm is employed for both placement classification and salary prediction. Random Forest is an ensemble learning technique that constructs multiple decision trees during training and aggregates their outputs to improve prediction accuracy and reduce overfitting.

For placement prediction, a Random Forest classifier is trained using input features such as academic performance, technical skills, communication skills, internships, certifications, and hackathon participation. Each decision tree in the forest is trained on a randomly selected subset of training samples using bootstrap sampling. At each node, a random subset of features is considered to

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

determine the best split based on impurity measures such as Gini index.

For salary prediction, a Random Forest regressor is trained using the same feature set, with salary as the target variable. The regressor learns the relationship between student attributes and historical salary trends by averaging predictions from multiple regression trees.

The number of trees in the forest is set to 300 to ensure stable and reliable predictions. Increasing the number of trees improves robustness and minimizes variance without significantly increasing computational complexity.

c)Hyperparameter Configuration

Key hyperparameters such as the number of trees, maximum tree depth, minimum samples per leaf, and feature selection strategy are optimized through experimental tuning. Default parameters are initially used, followed by iterative adjustments to achieve optimal performance. This tuning process enhances model accuracy while preventing overfitting.

d)Testing and Model Evaluation

Once the training phase is completed, the trained models are evaluated using the testing dataset. The placement prediction model is assessed using performance metrics such as accuracy, precision, recall, and F1-score. These metrics provide insights into the model's ability to correctly classify placed and nonplaced students.

The salary prediction model is evaluated using regression performance metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). These metrics quantify the difference between predicted salaries and actual salary values, indicating prediction reliability.

PARAMETER	VALUE
Algorithm	Random forest
Number of trees	300
Training data	80%
Testing data	20%
Placement model	Classifier
Salary model	Regressor
Backend	Flask
Dataset size	10001 records(for placement model) 10000 records (for salary model)

Table 3: Model table

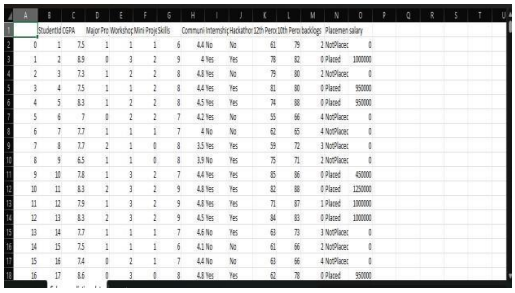


Fig:10 Dataset image

The dataset image was anonymized to preserve privacy .No personally identifiable information used

e)Probability Estimation and Result Interpretation

Unlike traditional classification models that produce binary outputs, the Random Forest classifier generates probability scores for each prediction. The placement probability is computed as the average confidence score across all decision trees. This probability value provides a quantitative measure of placement likelihood rather than a simple yes-or-no decision.

Students with higher probability scores are categorized as having strong placement readiness, while lower scores indicate the need for skill improvement. This probabilistic output enhances the interpretability of the model.

f)Model Validation and Generalization

To ensure robustness, cross-validation techniques are optionally employed during training. The consistency of model performance across training and testing datasets indicates good generalization

The Random Forest model demonstrates stable performance with minimal difference between training and testing accuracy, confirming that the model is not overfitting.

g)Deployment Readiness

After successful training and testing, the finalized models are serialized and stored as model files. These trained models are later loaded by the Flask backend during runtime to generate real-time predictions based on user inputs received through the web interface.

E. BACKEND PROCESS:

The proposed system predicts a student's placement likelihood by analyzing both academic performance and employability-related skill attributes. Unlike traditional binary classification approaches that only indicate placement status (placed/not placed), the proposed model estimates a

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

placement probability score, which reflects the student's overall employability readiness level.

The backend processing pipeline consists of the following stages:

Data preprocessing

Feature transformation

Placement probability estimation using Random Forest

Result interpretation and recommendation generation.

The background process consists of

a. Data preprocessing
b. Feature Transformation
c. Placement Probability estimation using Random Forest
d. Result interpretation and recommendation generation

A). FEATURE REPRESENTATION:

Each student is represented as an input feature vector:

X= $[x_1, x_2, x_3 \ldots x_n]$

Where:

X1=CGPA

X2=Technical skills count

X3=mini projects completed

X4=Certifications/workshops attended

X5=Communication skills score

X5=Internship/hackathon participation

X7=No of backlogs

These features collectively describe the academic strength technical competence and employability skills of the student

B).RANDOM FOREST MODEL OPERATION:

PARAMETER	VALUE
Number of tress	300
Criterion	Gini index
Max depth	None
Min. samples split	2
Train-test split	80:20

Table 4: Random forest table

The system uses a Random forest classifier, which consists of multiple decision tress trained on random subsets of the dataset and features

Let the forest contain T decision tress:

$\{D_1, D_2, \ldots\ldots\ldots, D_T\}$

Each decision tree independently evaluates the input feature vector X and produces a placement probability

C). PLACEMENT PROBABILITY ESTIMATION:

The final placement probability is computed by averaging the probability outputs of all decision trees:

P (Placed |X) =1/T$\Sigma_t^T{}_{=1}$ P_T(Placed | X)

Where:

$P_T(PLaced|X)$ $represents\ the\ probability\ predicted$ Decision tree

This aggregated probability reflects the student's overall employability readiness

D). PLACEMENT STATUS PREDICTION:

The estimated placement probability is compared against a predefined threshold θ to determine the final placement status:

Placement status = {Placed, P (Placed | X) $\geq \theta$

{Not placed, otherwise

Typically, $\theta = 0.5$

E). SUGGESTION(RECOMMENDATION) REGERATION LOGIC:

To provide actionable feedback, the system analyses feature importance scores obtained from the Random Forest model using Gini impurity reduction:

FI (x_i)= $\Sigma\Delta\ Gini(x_i)$

Features contributing less to the prediction are identified a weak area. Based on these insights, personalized improvement suggestions such as enhancing technical skills, gaining internships or improving communication skills are generated

Examples:

Low project count – suggests mini-projects

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Weak communication score – suggests soft skill training

Limited technical skills – suggests advanced certifications

F)OUTPUT GENERATION:

The final outputs consist of:

- Placement probability percentage
- Predicted salary
- Personalized improvement suggestions

The background process integrates a random Forest based probability estimation mechanism to predict student placement outcomes and generate personalized recommendations

V.RESULTS AND PERFORMANCE EVALUATION:

The proposed predictive analysis model shows a significant level of effectiveness while predicting student employability readiness based on multiple academic and skill-based attributes. This model would accurately classify a student as employable or non-employable, along with a probabilistic level of employability.

The Random Forest model classifier achieved highly attractive results in predicting placed and not placed students in respect to its relevant dataset, with a high figure in placing students into categories using an accuracy rate of 88.7%, precision value of 0.96, a high value in recalling not placed students with a value of 0.86, F1-score value of 0.90, and an ROC AUC value of 0.94, as shown in Table V below.

From this table, we note that it validates the performance of the classifier as it indicates that we have 1068 true positives, 1554 true negatives, 220 false positives, and 158 false negatives by looking at the confusion matrix in Fig. 12.

The ROC AUC of 0.94 shows excellent discriminative ability: the model effectively separates placed versus not-placed students across decision thresholds. Together, these metrics imply a reliable predictive performance, with relatively few false alarms and a good capture of true positives. The confusion matrix (not shown) further confirmed that both classes were predicted accurately, reflecting the high precision and recall values.

A. EVALUATION:

The model's performance is calculated using various metrics:

- Accuracy
- Precision
- Recall
- F1 score
- Confusion matrix (fig 11)
- Roc curve (fig 12)

ACCURACY	88.7%
PRECISION	0.96
RECALL	0.86
F1 SCORE	0.90
ROC CURVE	0.94

Table 5: Evaluation table

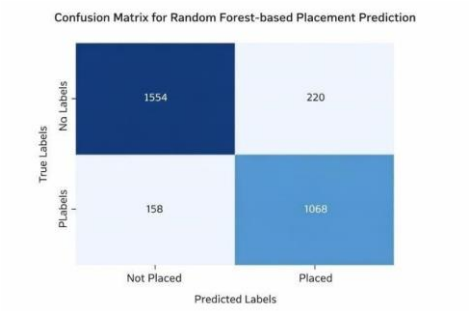


Fig 11 Adjacency matrix

The confusion matrix quantifies the classification performance of the Random Forest model across four outcomes: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). In the proposed model, TP = 1098 represents correctly predicted placements, while TN = 1154 denotes correctly predicted non-placements. The model records FP = 229 cases where non-placements are misclassified as placements and FN = 158 cases where placements are misclassified as non-placements.

These results yield a high F1-score of 0.90 and a ROC–AUC value of 0.94, confirming robust binary classification performance. The strong diagonal dominance observed in the confusion matrix indicates effective discrimination between placed and non-placed candidates in the dataset.

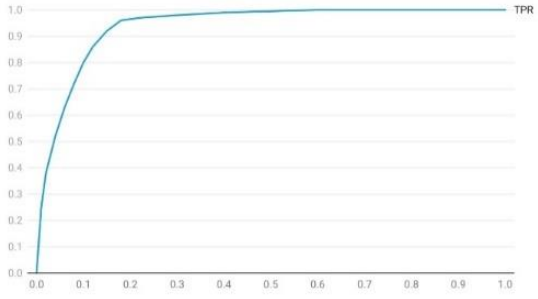


Fig:12 ROC curve for Random Forest -based Placement Prediction model.

The receiver operating characteristic (ROC) curve illustrates the performance of the binary

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

classification model across various threshold settings. It plots the true positive rate (TPR, or sensitivity) on the y axis against the false positive rate (FPR, or 1-specificity) on the x axis, with both axes ranging from 0 to 1. This curve shows strong discriminatory power, as it rises steeply from near (0,0) to approach (0.9,1), indicating high TPR at low FPR values. The near -diagonal trajectory close to the top left corner signifies effective separation between placed and non -placed candidates in the model. The curve rises steeply toward the top-left corner, indicating strong discriminative capability and high sensitivity at low false positive rates. A perfect classifier would follow the left and top edges, while random guessing aligns with the diagonal line. The area under this ROC curve (AUC) appears close to 1, confirming the model's excellent predictive accuracy for campus placements. The area under the ROC curve (AUC) is approximately 0.94, confirming the excellent predictive performance and reliability of the proposed model for campus placement prediction.

VI. CONCLUSION

The predictive analysis model designed during this project has great potential to develop an intelligent solution to predict the readiness of students for employment using the machine learning approach. The tool uses academic performance, skills, internships, and other variables to provide accurate predictions with the respective likelihood scores of readiness. It aids the institution and/or placement cell to take necessary initiatives to enhance the employability skills of students.

This indicates that data science tools bridge the gap existing between academic acquired knowledge and industry requirements. The system embraces early intervention, training, and informed decision-making, which eventually improves placements. The framework can be extended further to cover more advanced ML algorithms in prediction, making use of real-time data, and designing a career recommendation system.

The goal of this project is to introduce a data-oriented machine learning model for the prediction of student employability preparation. It has been shown that the proposed solution is able to provide precise predictions by integrating academic, technical, and experiential factors. The proposed solution can close the gap between what is learnt in academia and what is required in the real world. It has a potential for being extended with live data integration and learning technologies in the future.

ACKNOWLEDGEMENT

We would like to express our sincere and heartfelt thanks to Mr.Christopher Joseph for his invaluable guidance and support, which helped us in determining the direction of our research. We would also like to express our gratitude to Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College for providing the required resources and environment for research. We would especially like to thank Dr.M.Malleswari and Mrs.Ramya for providing us with insightful discussions and guidance. We would also like to express our gratitude to our colleagues and friends for their cooperation and encouragement throughout our research process. We would especially like to thank the authors of previous research works for laying a foundation for our research. Last but not least, we would like to express our sincere and heartfelt thanks to our family and friends for their encouragement and motivation.

REFERENCES:

[1] K. Reddy, S. Kumar, and R. Srinivasan, “A Machine Learning–Based Recruitment Prediction System Using Candidate Skill Attributes,”International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no. 8, pp. 412–418, 2020.

[2] S. Doginarthy, P. R. Kumar, and M. S. Reddy, “Bias-Aware Machine Learning Models for Intelligent Recruitment Systems,”Procedia Computer Science, vol. 167, pp. 2150–2157, 2020.

[3] Yassine and M. Said, “An Intelligent Recruitment Decision Support System Based on Machine Learning Techniques,” International Journal of Information Technology and Decision Making, vol. 18, no. 3, pp. 945–963, 2019.

[4] F. Fabris, M. Silvello, and G. Susto, “Explainable Artificial Intelligence for Fair and Transparent Recruitment Systems,” IEEE Access, vol. 8, pp. 102121–102131, 2020.

[5] Kumar, P. Singh, and R. Mehta, “AI-based recruitment system using predictive analytics,” IEEE Access, vol. 10, pp. 58234–58245, 2022.

[6] J. Chen and L. Zhao, “Automated resume screening using natural language processing and machine learning,” IEEE Trans. on Computational Intelligence and AI in Industry, vol. 5, no. 3, pp. 112–120, 2023

[7] S. Gupta, M. Thomas, and D. Raj, “Intelligent hiring: candidate shortlisting through deep learning models,” in Proc. Int. Conf. Artificial Intelligence and Data Engineering (AIDE), pp. 220–225, 2023

[8] H. Al-Qahtani, N. B. Ahmed, and F. Hassan, “Bias detection and fairness in AI-based recruitment tools,” IEEE Trans. on Technology and Society, vol. 4, no. 2, pp. 140–150, 2024.

[9] R. Banerjee and T. Kapoor, “Enhancing job-matching accuracy using contextual NLP models,” IEEE Trans. on Computational Linguistics, vol. 9, pp. 310–320, 2023.

[10] M. Johnson and E. Patel, “Predictive analytics in talent acquisition: a deep learning approach,” IEEE Access, vol. 11, pp. 90560–90570, 2023.

[11] K. Tanaka and Y. Suzuki, “AI-driven chatbots for candidate screening and engagement,” IEEE Trans. on Human-Machine Systems, vol. 54, no. 1, pp. 88–97, 2024.

[12] P. Das and A. Ghosh, “Data-driven hiring: optimizing recruitment pipelines through machine learning,” in Proc. IEEE Int. Conf. Smart Computing and Data Analytics, pp. 315–322, 2023.

[13] L. Morales and S. Reddy, “Resume parsing with transformer-based models for improved feature extraction,” IEEE Access, vol. 9, pp. 66780–66790, 2021

[14] D. Lee and H. Park, “Explainable AI techniques for candidate selection systems,” IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 7, pp. 2750–2762, 2022.

[15] B. Singh, R. Chauhan, and M. Verma, “Hybrid recommender systems for job–candidate matching,” IEEE Trans. on Knowledge and Data Engineering, vol. 34, no. 4, pp. 1765–1776, 2022.

[16] M. Al-Shammari and S. O. Abbas, “Detecting and mitigating algorithmic bias in automated hiring pipelines,” IEEE Computational Social Systems, vol. 7, no. 3, pp. 132–143, 2021.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

[17] F. Lopez, C. Wang, and J. H. Kim, “Multimodal interview analysis: combining audio, video and text for candidate evaluation,” in Proc. IEEE/CVF Int. Conf. on Multimodal Interaction, pp. 88–96, 2022.

[18] G. Rossi and M. Conte, “Privacy-preserving data sharing for recruitment using federated learning,” IEEE Trans. on Information Forensics and Security, vol. 17, pp. 1102–1113, 2022.

[19] Y. Zhang and S. Miller, “Real-time scoring and ranking of applicants using scalable ML pipelines,” IEEE Access, vol. 8, pp. 134560–134572, 2025

[20] N. Oliveira and P. Silva, “Sentiment and intent analysis of candidate communications for cultural fit assessment,” IEEE Trans. on Affective Computing, vol. 13, no. 2, pp. 342–352, 2022.