



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

INTELLIGENT CAMPUS PLACEMENT PORTAL

¹Prof. Madhuri Kale, ²Harsh Rawate, ³Kaivalya Satav, ⁴Riten Chaudhari, ⁵Soham Kore

¹Professor, Department Of Computer Engineering, DPCOE, Pune-412207, India

^{2,3,4,5}Student, Department of Computer Engineering, DPCOE, Pune-412207, India

Abstract: An Intelligent Campus Placement Portal that predicts each student's placement probability and recommends targeted improvements to increase chances of selection. The system centralizes student, company, and placement records while offering dashboards for students, faculty, and administrators. Using academic scores, aptitude and programming test results, skills, and historical placement data, the portal computes a confidence score for likely companies and highlights specific skill or aptitude gaps with prioritized, actionable suggestions. Faculty gain visualization tools to analyze placement trends, monitor cohort readiness, and run targeted interventions. The portal is web-based, cloud-deployable, and designed for real-world integration with college workflows .

KeyWords : XG Boost, NLP, Random Forest, Vote Classifier.

1. INTRODUCTION

An AI-Driven Campus Placement Portal that combines student academic records, aptitude and programming assessments, and historical placement data to predict placement probabilities and provide targeted, actionable recommendations. The system offers role-based dashboards for students, faculty, and administrators to centralize placement workflows and improve transparency.

Students receive individualized placement confidence scores and prioritized skill-gap suggestions that quantify how specific improvements raise their chances with particular companies. Faculty and administrators gain cohort-level visualizations and exportable reports to monitor trends, identify at-risk students, and design focused interventions that increase overall placement effectiveness.

Studies provide empirical evidence on how Machine Learning (ML) models can optimize student learning paths and predict employability. Research has shown that algorithms like **Random Forest and Logistic Regression** can effectively predict student placement success using features such as academic performance (CGPA), internship experience, and specific skills. This allows institutions to identify the most significant factors influencing hiring and guide students accordingly.

2. CORE AI APPLICATIONS IN THE PLACEMENT PORTAL:

Using AI in the campus placement office is already showing big improvements in how things are run, how prepared students are, and how university resources are used. Think of it this way: AI-powered

predictive tools can spot students who might be at risk of not getting a job early on. This means staff can step in and help before it's too late.

We have solid proof from studies that Machine Learning (ML) models can help optimize a student's learning path and even predict if they'll get a job. Research shows that models like **Random Forest and Logistic Regression** are really good at predicting a student's placement success. They do this by looking at things like their grades (CGPA), internship experience, and specific technical skills. This gives the college powerful insights into what actually matters for getting hired, so they can guide students better.

AI can also help optimize a student's profile. For example, it can recommend specific courses or projects for them to take. This helps students use their time at college more wisely. By looking at data from past hires, the AI can figure out the best set of skills for certain jobs, leading to **better placement rates** and fewer students in jobs they are overqualified for. All this shows how AI can completely change a student's journey from the classroom to their career, with huge benefits for everyone involved.

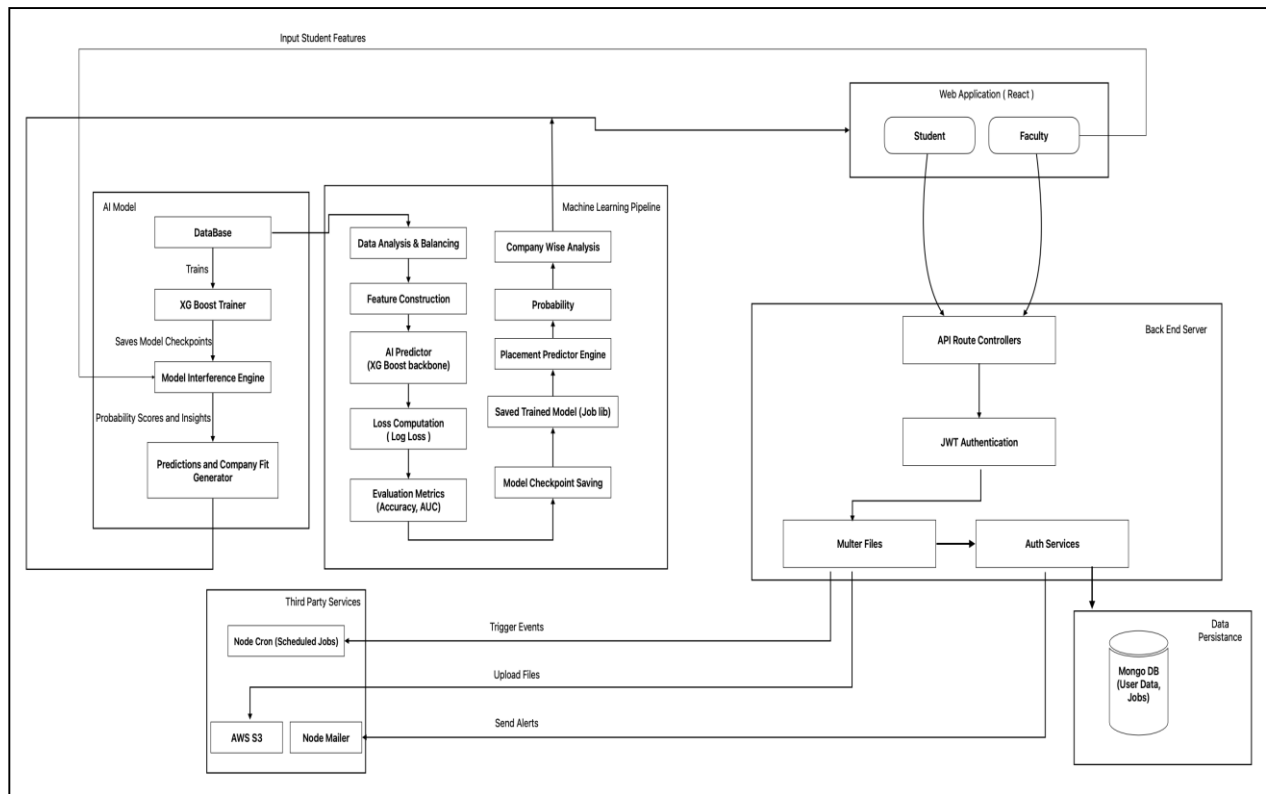
Bringing AI into the hiring process has totally changed how we find and match students with the right jobs. Instead of just matching keywords, AI models can look at a student's whole profile to see if they're a good fit, guess how well they'd do in an interview, and **find great candidates way faster** than any person could.

3. SYSTEM DESIGN :

Portal is designed as a modern web application. The **frontend (what the user sees and clicks)** is built using **React**. This allows us to create a fast, responsive, and user-friendly interface for both students and recruiters.

The **backend (the "brains" of the operation)** is where the **AI/ML** models live. The React frontend communicates with this backend, sending requests (like "a student just uploaded their resume") and This separation makes the whole system scalable and easier to maintain.

4. SYSTEM ARCHITECTURE:



The proposed system follows a modular architecture that integrates a web application, backend services, database, and a machine learning module for placement prediction. The frontend, developed using **React**, allows students and faculty to interact with the system, input data, and view prediction results.

The backend, implemented using **Node.js**, manages API requests, authentication, and communication between components. Student and placement **data are stored in MongoDB** for efficient access and processing.

The system includes a machine learning pipeline based on **XGBoost**, which analyzes student features to generate placement probability scores and company-specific insights. The trained model is saved and used by the inference engine to provide real-time predictions.

Additional services such as scheduled tasks and cloud storage support automated updates, file handling, and notification features. Overall, the architecture ensures smooth data flow, scalability, and effective prediction capabilities.

5. WORKFLOW :

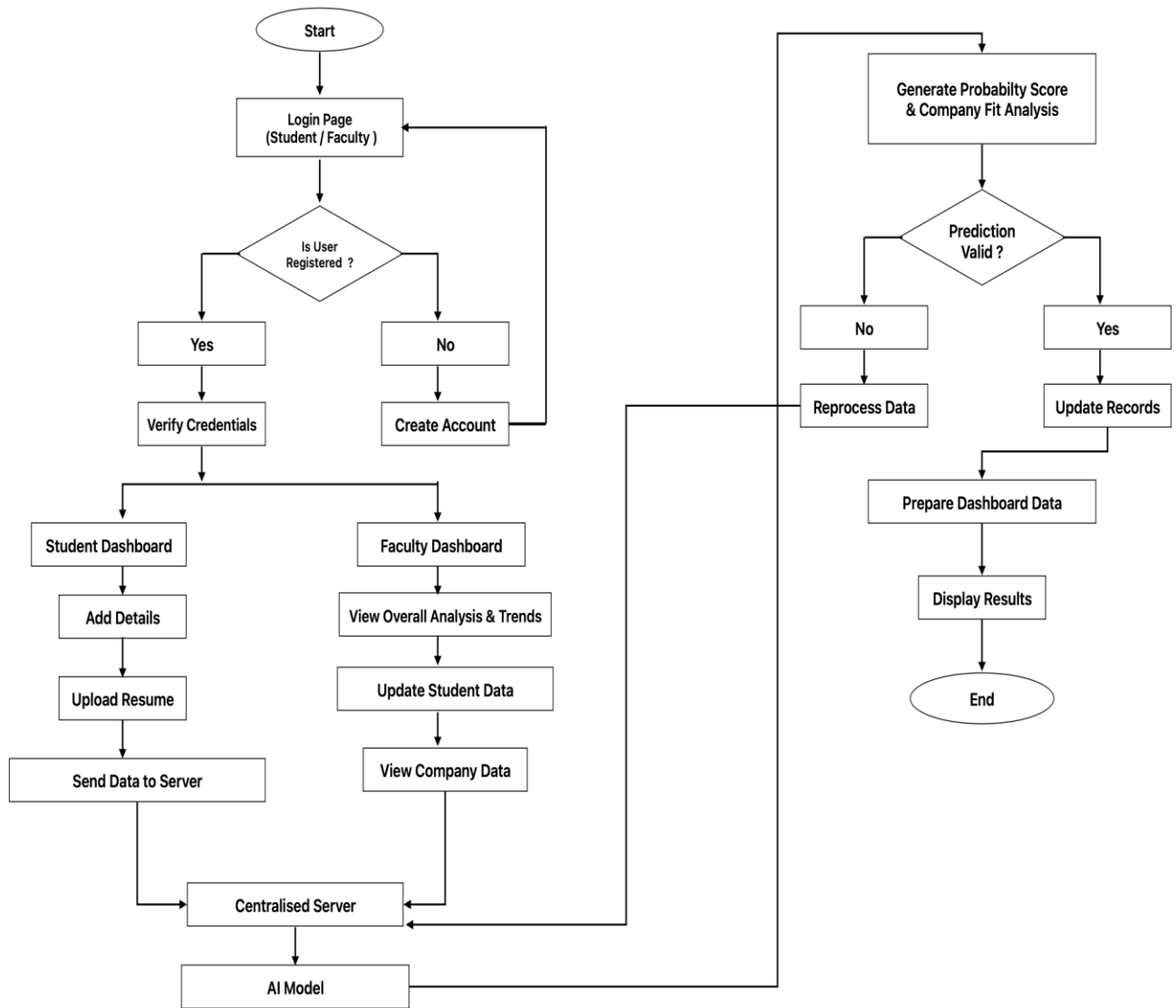


Figure 2. Model Workflow

The workflow of the system begins with the user accessing the login page, where students or faculty either register or log in using valid credentials. After successful authentication, users are directed to their respective dashboards.

Students can enter their academic details and upload resumes, which are then sent to the server for processing and storage. Faculty members can view student data, company information, and overall placement trends through analytical dashboards.

The collected data is processed by the AI model, which generates placement probability scores and company-fit analysis. If the prediction is valid, the results are displayed on the dashboard; otherwise, the data is reprocessed to improve accuracy. Finally, the system updates records and presents the results to users, completing the workflow.

6.METHODOLOGY:

This section gets into the "how" of our Intelligent placement portal. We'll talk about the specific AI techniques we used, the kind of data that makes it all work, and the models we built to turn that data into useful predictions.

6.1 AI TECHNIQUES

To build a system this smart, you need a few key AI ingredients:

6.1.1. PREDICTIVE MACHINE LEARNING (ML):

This is our forecasting engine. We use it to predict a student's "**likelihood of placement.**"

- We've implemented powerful models like **XGBoost** and **Random Forest**.
- We also created a **Vote Classifier**, which acts as an ensemble. It combines the predictions from both XGBoost and Random Forest and "votes" on the final answer. This method is typically more robust and accurate than any single model on its own.
- These models are trained on structured data points like **CGPA, Aptitude_Score, Number_of_Internships, and Number_of_Skills**.

6.1.2. CONTENT-BASED RECOMMENDER SYSTEM:

This is our "matching" engine. Instead of reading paragraphs, this system works by directly comparing the structured tags from students and jobs.

- For example, a student's profile might have `Skills = ["Python", "React", "SQL"]`.
- A job posting might have `Required_Skills = ["Python", "SQL", "AWS"]`.
- The recommender system directly compares these lists to calculate a **match score**. This allows us to instantly find the best-fit students for a job and the most relevant jobs for a student, which are then displayed in the React UI.

6.2 THE DATA :

Our system's success depends on clean, structured data, which we collect in two main categories:

- **Student Data:** All information is entered into specific fields. This includes academic records (e.g., CGPA), Aptitude_Score, Branch, Number_of_Internships, and a list of Skills.
- **Recruiter Data:** All job postings are also structured. This includes Company_Name, Job_Title, Required_CGPA, and a corresponding list of Required_Skills.

6.3 WEB DEVELOPMENT :

6.3.1 : MERN STACK IMPLEMENTATION :

The proposed system is developed using the MERN Stack, which is a modern full-stack JavaScript framework consisting of **MongoDB, Express.js, React.js, and Node.js**. This stack enables the development of a scalable, efficient, and dynamic web application using a unified programming language across both frontend and backend.

6.3.2 : COMPONENTS OF MERN STACK :

MongoDB (Database Layer) :

MongoDB is a NoSQL database used to store application data in a flexible, JSON-like format. It allows efficient handling of large volumes of structured and unstructured data such as user profiles, job listings, and placement records. Its schema-less design makes it suitable for applications requiring frequent updates and scalability.

Express.js (Backend Framework) :

Express.js is a lightweight web application framework for Node.js that simplifies the creation of server-side logic and APIs. It is used to handle routing, middleware, and HTTP requests, ensuring smooth communication between the frontend and the database.

React.js (Frontend Library) :

React.js is used to build an interactive and responsive user interface. It enables the development of reusable UI components and efficient rendering using a virtual DOM, improving the overall user experience of the system.

Node.js (Runtime Environment) :

Node.js is used to execute JavaScript code on the server side. It provides an event-driven and non-blocking architecture, making it suitable for handling multiple user requests simultaneously.

6.4 MODELING :

Our AI models—**XGBoost, Random Forest**, and the **Vote Classifier**—are trained on our extensive historical placement datasets. To ensure they are reliable and not just "memorizing" the old data, we validate them using a technique called **k-fold cross-validation**.

We then evaluate their performance using a full set of key classification metrics. This is crucial because just "Accuracy" isn't enough. Our evaluation includes:

- **Accuracy:** How often the model is right (Placed vs. Not Placed).
- **Precision:** Of all the students we *predicted* would be placed, how many actually were? (This is important for recruiter trust).
- **Recall:** Of all the students who *actually* got placed, how many did our model find? (This is important for helping students).

- **F1-Score:** The balanced average of Precision and Recall, giving us a single number to judge model quality.
- **Support:** This just tells us how many samples were in each class (e.g., 400 Placed, 100 Not Placed), which helps us check for data imbalance.

7. RESULTS AND EVALUATION:

7.1 DATA VISUALIZATIONS :

1. **CGPA vs. Placement Probability:** We generated a scatter plot to show the relationship between a student's CGPA and their probability of being placed. This plot helps us visually confirm the correlation between academic performance and placement success.

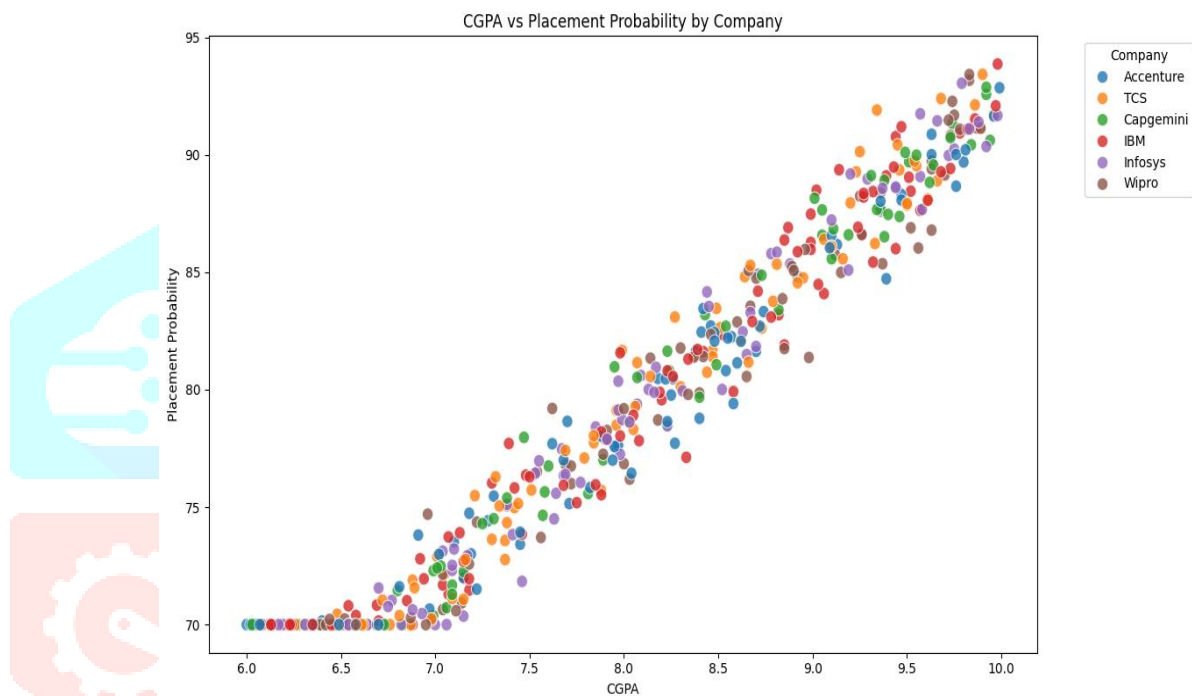


Figure 3. CGPA vs. Placement Probability

2. **Distribution of Placements:** A counterplot was used to show the distribution of 'Company' versus 'Number of Placements' students in our dataset. This was a critical step to check for any class imbalance (e.g., having far more 'Placed' data than 'Not Placed'), which can affect how we interpret model performance.

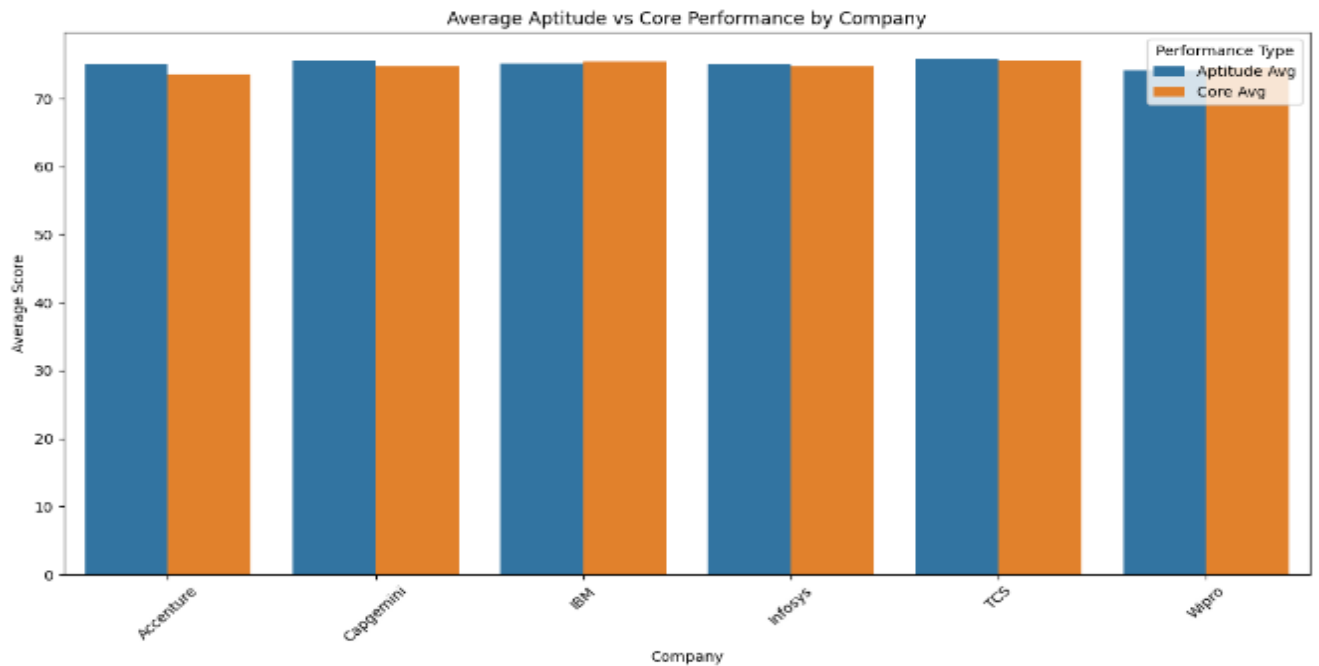


Figure 4 : Distribution of Placements

3. **Company-Specific Insights:** We created a **bar plot** showing the average aptitude scores versus core subject performance, grouped by **company names on the x-axis** and the **average scores on the y-axis**. This plot provides powerful, actionable insights. For example, it might show that Company A values high aptitude scores above all, while Company B focuses more on strong core technical grades. This is similar to the "Feature Importance" plot in the source paper, as it highlights which factors are most important for a successful outcome.

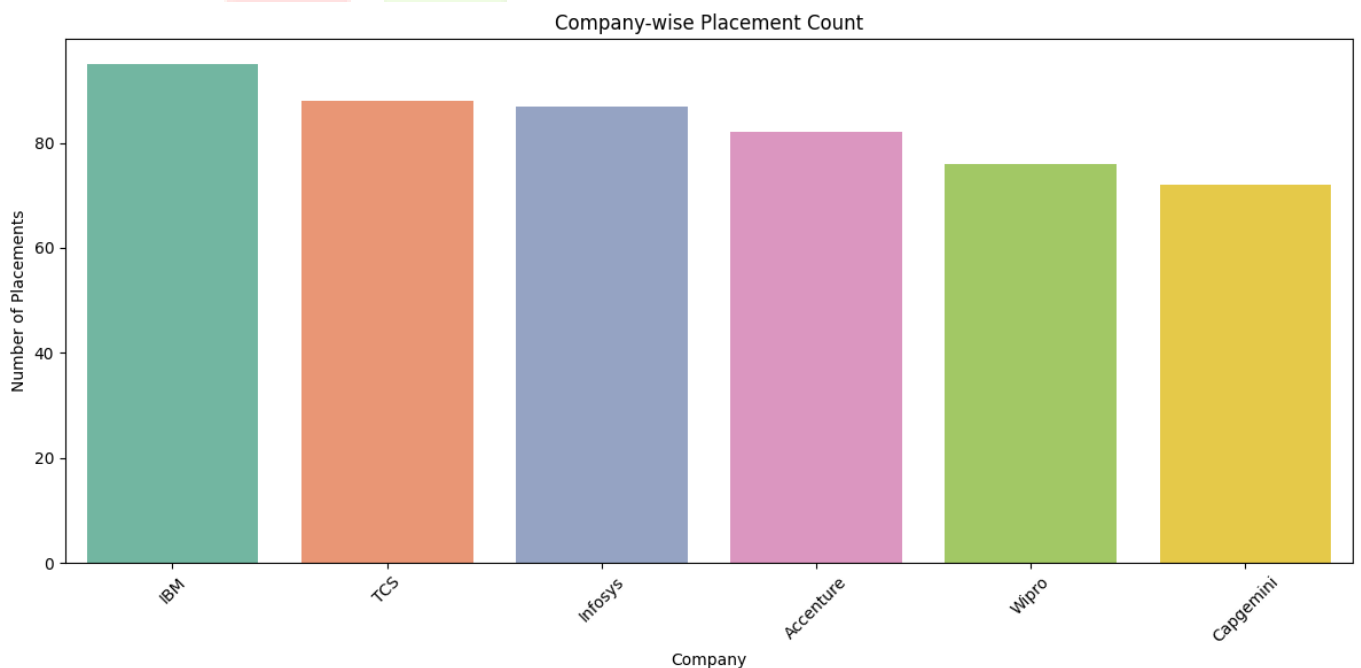


Figure 5. Company Specific Insights

7.2 MODEL PERFORMANCE :

To test our predictive models, we used a **Confusion Matrix** to see the raw counts of our correct and incorrect predictions (True Positives, True Negatives, False Positives, and False Negatives).

From this matrix, we calculated a full set of evaluation metrics to judge the performance of our **XGBoost**, **Random Forest**, and **Vote Classifier** models.

- **Accuracy:** The percentage of total predictions the model got right (both placed and not placed).
- **Precision:** This answers: "Of all the students the model *predicted* would be placed, what percentage *actually* got placed?" A high precision builds trust with recruiters.
- **Recall:** This answers: "Of all the students who *actually* got placed, what percentage did the model *find*?" A high recall is crucial for making sure students don't get missed.
- **F1-Score:** The balanced average of Precision and Recall. This is often the most important metric, especially if our countplot showed an imbalance between 'Placed' and 'Not Placed' students.
- **Support:** This simply shows how many students were in each class, confirming the data distribution for our tests.

8. FUTURE SCOPE:

- **Skill-Gap Detector:** It can analyse resume and detect the skill-gap and recommend some courses or tech-stack to learn.
- **Enhanced Prediction Analysis:** As more people will register themselves on the platform , more data trends can be extracted ,which will ultimately help in improved efficiency.
- **Cloud Deployment:** By deploying the platform over the cloud, several other campuses can also access the features and functionalities

9. CONCLUSION:

The AI-Driven Campus Placement Portal provides an intelligent, automated solution to simplify the placement process in educational institutions. By using AI within a React-based Web platform, It will predict the student's chances of getting selecting at a company based on the student's CGPA, Skills, No of Internships done, Aptitude test scores, Technical test scores.

Campus Placement Portal not only streamlines the tedious process but also act as bridge between manual management and data-based decision making. It will assist students in their lacking areas along with their changes at certain companies. It will also help faculty manage the student's data and also help in aggregates functionalities such as Number of students placed, Average Package offered, Maximum Package offered, etc.

With its scalable design, the system will be more transparent and efficient to use and it also has several potential ahead in future – as cloud integration, resume analysis, etc.

10. REFERENCES :

1. T. S. Suma, S. R. Anitha, and M. R. Preethi, “**Student Placement Prediction System using Machine Learning Algorithms**,” *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, vol. 8, no. 5, pp. 231–236, 2020.
2. A. Gupta and M. Sharma, “**Predictive Analysis of Campus Placement using Machine Learning**,” *International Journal of Emerging Trends in Engineering Research (IJETER)*, vol. 9, no. 8, pp. 1056–1061, 2021.
3. B. Kaur, S. Arora, and A. Singh, “**AI-Based Student Employability Prediction using Neural Networks**,” *International Journal of Computer Applications*, vol. 183, no. 42, pp. 25–30, 2022.
4. R. Patil, V. More, and P. Jadhav, “**Online Placement Management System**,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 4, pp. 3501–3504, 2019.
5. S. Reddy, K. Rao, and P. Thomas, “**Automated Placement Portal using Web Technologies**,” *International Journal of Advanced Research in Computer Science (IJARCS)*, vol. 11, no. 2, pp. 115–119, 2020.
6. V. Kumar, R. Singh, and A. Mehta, “**Artificial Intelligence in Higher Education: Predictive Analytics for Student Success**,” *IEEE Access*, vol. 9, pp. 108321–108334, 2021.

