Recruitment Outcome Forecasting Using Machine Learning & AI

Prof. Madhav Ingale
Dept. Computer Engineering
JSPM's JSCOE Pune, India
ni3zinzurke4040@gmail.com

Aryaman Chintalboina
Dept. Computer Engineering
JSPM's JSCOE Pune, India
phinglaspure123@gmail.com

Shivakant Pandey
Dept. Computer Engineering
JSPM's JSCOE Pune, India
aniket2659@gmail.com

Shrikant Patane
Dept. Computer Engineering
JSPM's JSCOE Pune, India
svghadle@gmail.com

Shivam Nimbalkar
Dept. Computer Engineering
JSPM's JSCOE Pune, India
svghadle@gmail.com

***Abstract*— Improving student placement rates remains a crucial challenge for higher education institutions. This paper presents a novel machine learning-based recommendation system that predicts potential placement outcomes for students within five categories: Dream Company, Core Company, Mass Recruiters, Not Eligible, and Not Interested. Leveraging historical student data, the system identifies and classifies students based on their academic performance, skills, and past placement trends. This empowers placement cells to proactively support high-potential students and guide them in honing their technical and interpersonal skills. Additionally, students themselves can utilize the system to gain insights into their likely placement outcomes and tailor their efforts accordingly. By demystifying the placement process and providing actionable insights, this system has the potential to significantly enhance student employability and optimize institutional placement strategies.**

Keywords: Numpy, Pandas, Matplotlib, Scikit-Learn, API, Flask , React , Python

I.INTRODUCTION

In the ever-evolving educational landscape, securing successful campus placements for students remains a paramount concern for higher education institutions. While traditional methods have served their purpose, the sheer volume and complexity of data generated during the placement process present a unique opportunity: to unlock valuable insights through data analysis and machine learning. Enter the Campus Placement Analysis project, a revolutionary initiative that aims to leverage this data to empower institutions, enhance student employability, and bridge the gap between academia and industry.

By harnessing the power of data and technology, the Campus Placement Analysis project stands poised to transform the way institutions approach placements. It's a win-win for everyone: institutions make informed decisions, students get personalized guidance, and the industry benefits from a workforce equipped with the right skills. This is not just a project; it's a revolution in the making, paving the way for a future where data-driven insights shape successful careers and bridge the gap between academia and industry.efficient data mining and data analytics capability- for the implementation of the system.

II.OVERVIEW OF THE PROJECT OBJECTIVES:

The main purpose of the Campus Placement Analyzer is to provide colleges and universities with a comprehensive tool that leverages data analysis and machine learning to optimize the campus placement process. This tool aims to offer valuable insights, personalized recommendations, and predictions for students' placement outcomes. Analyzing student data and historical placement trends, helps institutions enhance their placement strategies, improve student employability, and foster stronger connections with potential employers. Data Collection and Input: Colleges provide an Excel sheet containing detailed information about students, including academic records, skills, internships, and any other relevant details. Data Processing and Analysis: The system takes the Excel sheet and processes the data to create a structured dataset. Feature engineering is performed to extract key attributes like academic scores, skills proficiency, number of internships done, and more. Stats and Insights Dashboard: The system generates a comprehensive dashboard with detailed statistics about upcoming placements. Colleges can view placement stats across all branches and the entire campus. Insights include placement success rates and average salary offers. Personalized Recommendations: Based on individual student profiles and skill sets, the system provides personalized recommendations. Salary Prediction: Utilizing machine learning algorithms, the system predicts potential salary ranges for each student based on their data and skills. Visualization and Reporting: The dashboard includes visualizations such as graphs, and charts to present data trends and patterns. Colleges can use these insights to make informed decisions about refining placement strategies. Student Engagement: Students can access the system to view their personalized recommendations and predicted salary ranges..

III.Technology in project

Server-Side Infrastructure:

- **Hardware:** Redhat Openshift container platform provides scalability and high availability for the application. Load balancers can be added if required to distribute traffic and ensure seamless operation.
- **Operating System:** Linux serves as the reliable and secure foundation for the server-side environment.

Web Development:

- **Server-Side Scripting:** Python, a versatile and widely used language, powers the back-end logic and API development.
- **Front-End Technologies:** HTML, CSS, and JavaScript form the core building blocks of the user interface, ensuring a dynamic and interactive experience.
- **Framework:** Flask, a lightweight and flexible framework, simplifies back-end development and API creation. Additionally, React, a popular JavaScript library, enhances the front-end development process.

Data Science and Machine Learning:

- **IBM Watson Studio:** This platform provides a cloud-based environment for data analysis and machine learning tasks. Jupyter notebooks serve as the primary interface for data exploration, model development, and experimentation.
- **Libraries:** Numpy, Pandas, and Matplotlib are essential Python libraries for data manipulation, analysis, and visualization. Scikit-Learn, a powerful machine learning library, is used for building and training predictive models.

Deployment and Collaboration:

- **Redhat Openshift:** This container platform hosts the application in a scalable and secure manner.
- **IBM Watson Machine Learning:** Deploys trained machine learning models for production use within the Watson Studio environment.
- **Git:** This version control system enables collaborative development and ensures code traceability.
- **Docker Hub:** Docker images are uploaded to this repository for efficient deployment and container management.

External APIs:

- **Affinda Resume Parser API:** This API extracts key information from student resumes, enhancing data accessibility and analysis capabilities.
- **Affinda Skill Suggestions API:** This API recommends relevant skills based on existing skills, providing personalized suggestions to students.

IV. Implementation

1. Data Collection and Management:

Gather Student Data: Collaborate with institutions to acquire historical placement data including student profiles, academic achievements, skills, internships, and placement outcomes. Ensure proper data security and privacy protocols are in place.

Data Preprocessing: Cleanse and standardize the collected data using Python libraries like Pandas and Numpy. Address missing values, inconsistencies, and formatting issues.

Data Storage: Choose a secure and scalable cloud storage solution like Amazon S3 or Google Cloud Storage to house the cleansed data.

2. System Development and Deployment: Software Development: Build the back-end using Python and Flask, implementing APIs for data access, analysis, and predictions. Design the front-end with React or a similar framework, prioritizing user-friendliness and information clarity.

Model Training: Utilize Jupyter notebooks within IBM Watson Studio to train machine learning models for personalized recommendations, salary prediction, and other functionalities. Scikit-Learn provides a versatile toolkit for model development.

API Integration: Integrate the Affinda Resume Parser and Skill Suggestions APIs to enrich the project's functionalities and enhance user experience.

Deployment: Employ Redhat Openshift to host the application and deploy the trained models using IBM Watson Machine Learning. Ensure robust security measures and load balancing strategies.

3. Testing and Refinement:

Thorough Testing: Conduct comprehensive unit testing, integration testing, and user acceptance testing to ensure the system's functionality, accuracy, and user experience meet expectations.

Model Evaluation: Evaluate the performance of machine learning models using relevant metrics like accuracy, precision, and recall. Refine the models based on evaluation results and consider cross-validation techniques.

Feedback and Iteration: Gather feedback from institutional stakeholders and students to continually improve the system's usability and effectiveness. Adapt based on feedback and findings from pilot deployments.

4. Institutional Integration and User Access:

Training and Support: Provide training materials and workshops for placement teams at institutions to effectively utilize the system's features and interpret insights.

Student Access: Develop a user-friendly interface for students to access personalized recommendations, skill gap analysis, and salary predictions. Ensure data privacy and security are maintained.

Continual Collaboration: Maintain ongoing communication and collaboration with institutions to address their evolving needs and incorporate suggestions for future enhancements.

5. Sustainability and Future Development:

Monitoring and Maintenance: Regularly monitor the system's performance and address any technical issues promptly. Implement security patching and updates to maintain operational integrity.

Data Enrichment: Explore integrating additional data sources, such as industry trends and job market data, to further refine predictions and insights.

Advanced Features: Consider incorporating natural language processing techniques for sentiment analysis of job descriptions and company reviews to gain deeper insights.

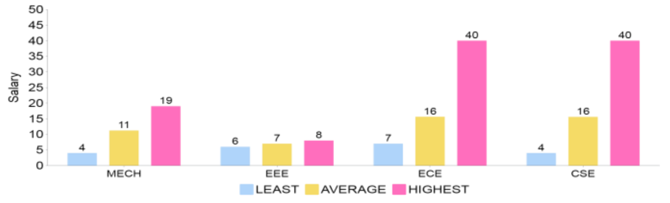
Scalability: Plan for future scalability as the system expands and user base grows, potentially by adopting a cloud-based infrastructure with auto-scaling capabilities.

MODEL EVALUATION:

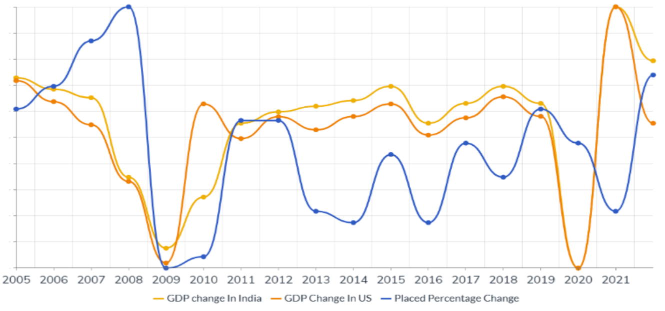
The model achieves an impressive accuracy of 98%

CHARTS:

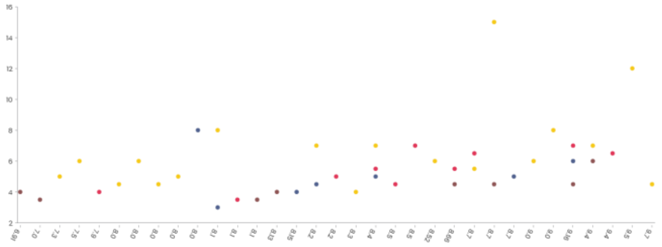
BRANCH WISE SALARY



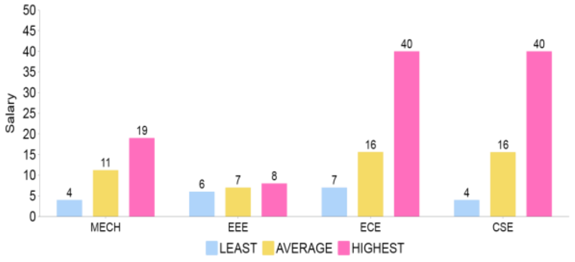
GDP V/S SALARY



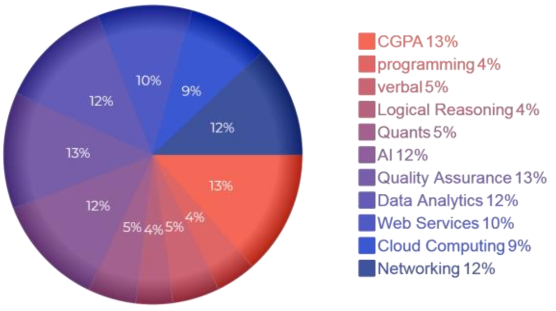
CGPA V/S SALARY



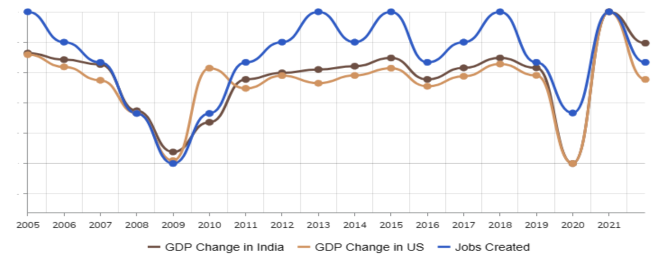
LEAST AVERAGE SALARY



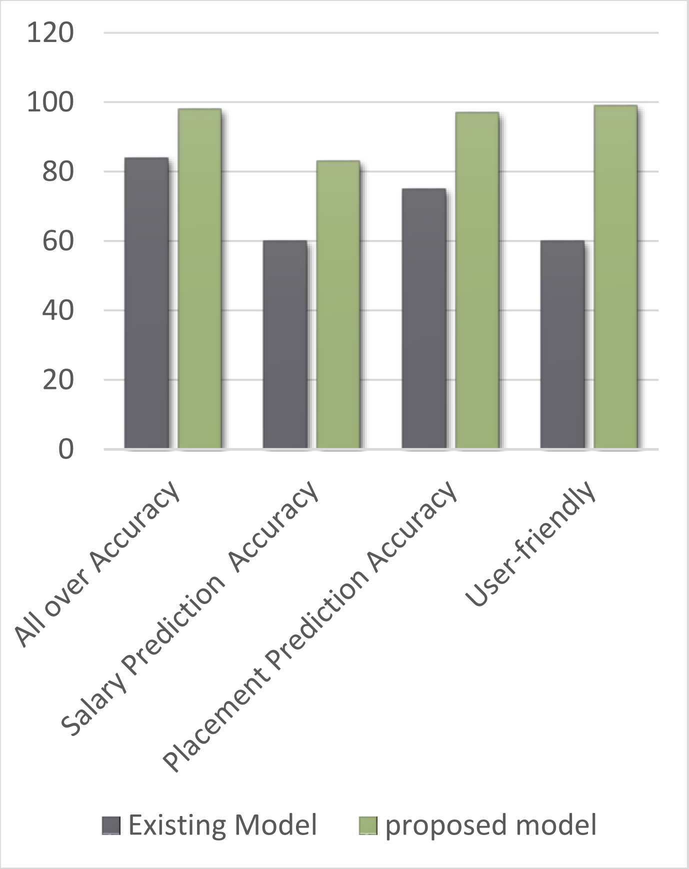
FACTORS AFFECTING ON PLACEMENTS



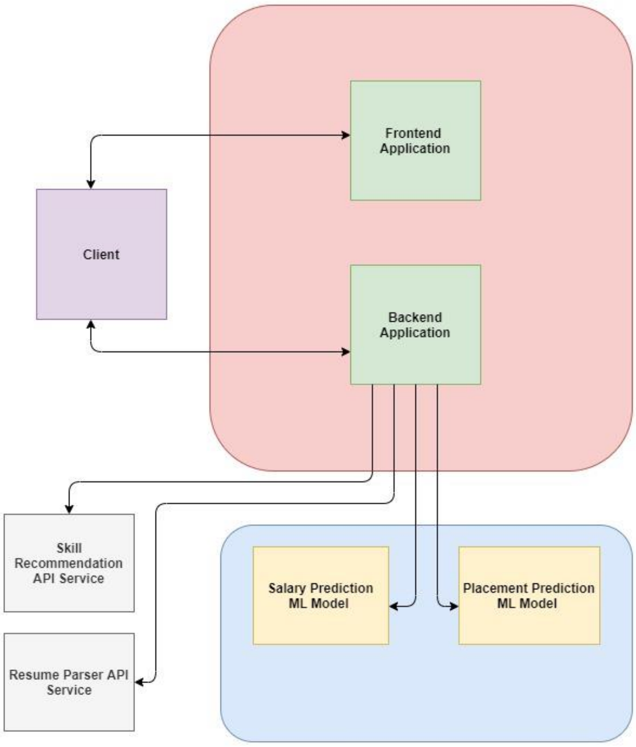
GDP V/S JOB



Result Comparison:

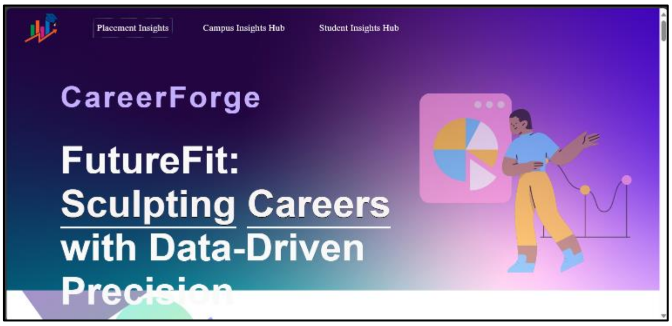


BLOCK DIAGRAM:


Frontend Application
Client
Backend Application
Skill Recommendation API Service
Resume Parser API Service
Salary Prediction ML Model
Placement Prediction ML Model

RESULTS:

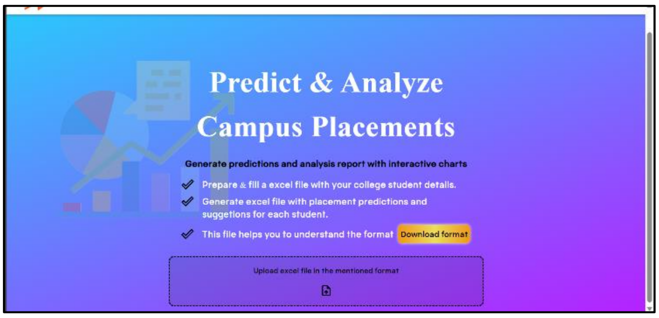
Placement insights:



CareerForge

FutureFit: Sculpting Careers with Data-Driven Precision

Campus Insights:



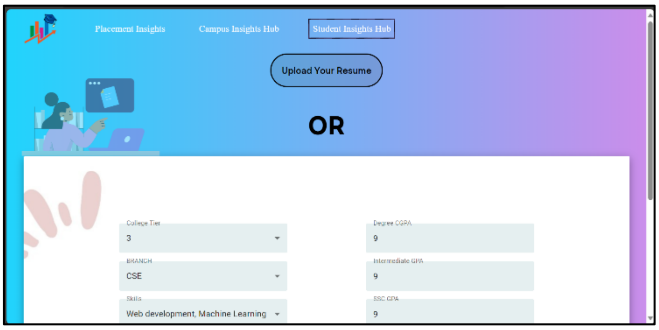
Predict & Analyze Campus Placements

Generate predictions and analysis report with interactive charts

- ✓ Prepare & fill a excel file with your college student details.
- ✓ Generate excel file with placement predictions and suggetions for each student.
- ✓ This file helps you to understand the format

Upload excel file in the mentioned format

Student Insights:



OR

College Tier: 3

Degree CGPA: 9

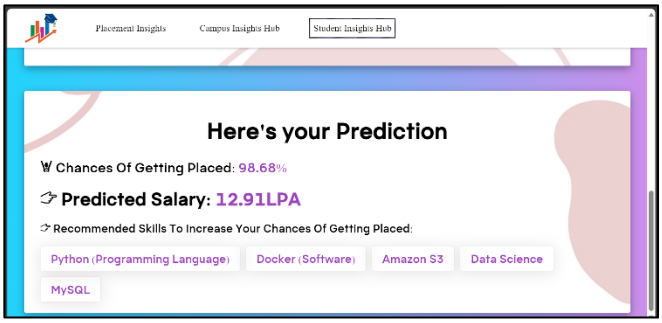
BRANCH: CSE

Intermediate GPA: 9

Skills: Web development, Machine Learning

SSC GPA: 9

Prediction and Recommendation:



Here's your Prediction

Chances Of Getting Placed: 98.68%

Predicted Salary: 12.91LPA

Recommended Skills To Increase Your Chances Of Getting Placed:

Python (Programming Language) Docker (Software) Amazon S3 Data Science MySQL

V.Conclusion

In conclusion, the Campus Placement Analysis project represents a pivotal step towards transforming the landscape of campus placements and career development. By harnessing the power of data analysis, machine learning, and web technologies. Through standardised data collection, insightful data visualizations, personalized recommendations, and accurate placement predictions, the project offers an integrated solution that empowers institutions to make informed decisions, enhance curriculum offerings, and strengthen ties with industries. For students, it means personalized guidance, skill development opportunities, and a clearer path towards desired career goals. Recruiters benefit from a refined talent acquisition process and access to candidates aligned with their needs. The real-world applications of this project extend beyond campus boundaries, impacting educational policies, industry dynamics, and workforce planning. As academia and industry become more closely intertwined, the Campus Placement Analysis project stands as a catalyst for a more efficient, responsive, and effective transition from education to employment. It paves the way for a future where data-driven insights shape careers, educational strategies, and the way society approaches the ever-evolving job market.

VI. REFERENCES

[1] Pang B Lee L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, The Association for computational linguistics, pp. 271ˆa278, 2004.

[2] Kajal Sarawgi, Vandana Parthak. Opinion Mining: Aspect evel sentiment analysis using SentiWordNet and Amazon web services, IJCA, 2019.

[3] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In The adaptive web, pages 325ˆa341. Springer, 2020.

[4] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In Recommender systems handbook, pages 73ˆa105. Springer, 2021.

[5] Pushpalatha, A., Sanmugam J. Harish, Pradeepa K. Jeya, and S. Madhu Bala. ”Gadget Recommendation System using Data Science.” In 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), pp. 1003 1005. IEEE, 2020.

[6] Pasumponpandian, A. ”A Hybrid-Algorithm for E-project Selection on Social Media.” Journal of Information Technology 2, no. 02 (2020): 116-122

[7] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). IEEE. Miami, FL, USA .

[8] Escalera, S., Baró, X., Gonzàlez, J., Bautista, M., Madadi, M., Reyes, M., . . . Guyon, I. (2014). ChaLearn Looking at People Challenge 2014: Dataset and Results. Workshop at the European Conference on Computer Vision (pp. 459 473). Springer, . Cham.