Campus Placement and Salary Prediction: Leveraging Machine Learning for Enhanced Employability

Jayashre, Raahul R, Roahith R and Shanmathi Ganesan

Department of Computer Science and Engineering
Shiv Nadar University, Chennai, Tamil Nadu, India
{jaya2004kra, raahulramesh11, roahith11, shanmathi18}@gmail.com

Abstract. In an era of fierce talent competition, optimizing campus placement and predicting graduate salaries is vital. This paper explores 'Campus Placement and Salary Prediction' using SVM, Random Forest, Logistic Regression, KNN, and Gradient Boosting. We analyze a comprehensive dataset with student info, academics, skills, internships, and placement outcomes to identify success factors. We employ a majority voting rule among these models and have created a user-friendly website for practical use in academic institutions. Our multifaceted research supports institutions in enhancing student employability and aligning academic goals with the evolving job market. Logistic Regression, one of the models employed has campus placement prediction, has an accuracy of 84%, whereas Gradient Boosting, one of the models used for salary estimation, has an accuracy of 78%.

Keywords: Campus Placement, Salary Prediction, Machine Learning, Website Interface, SVM, Random Forest, KNN, Logistic Regression, Gradient Boosting Machine, Majority Voting Rule

1 Introduction

Higher education increasingly focuses on campus placement and salary predictions, responding to dynamic job markets. Graduates, particularly in computer and commerce, face career uncertainties. Predictive models are aiding early identification of placement likelihood, aligning academic knowledge with industry needs. These models also guide faculty in improving placement rates.

Machine learning (ML) algorithms offer a promising solution for addressing challenges in campus placement and salary prediction. Research indicates that ensemble algorithms, combining multiple models, consistently outperform individual ones. Balancing datasets is crucial to prevent bias, providing students and institutions with precise, data-driven insights for career development.

Our paper uses innovative computer techniques to predict the first salary and job placements for students. Sophisticated algorithms analyze profession, academic achievement, and various factors. The intention is to provide students with a precise tool to assist schools and universities in refining their placement processes. We are working on creating a desktop/GUI program and a website for simple prediction checks

to improve the user experience. With permission, user-entered data is added to the dataset, enhancing prediction capabilities. Our comprehensive strategy continuously improves machine learning algorithms for more accurate school placement predictions, despite drawbacks such as a short dataset and reliance on statistical data.

Our research focuses on data-driven strategies for campus placement and salary prediction in education, empowering students with career insights. The unique majority voting rule, tailored for computer and commerce undergraduates, aims to enhance prediction accuracy. Our goal is to provide valuable tools for success in a rapidly changing world.

2 Literature Survey

Sachin et al. [2] apply AdaBoost Classifier after data preprocessing, achieving higher accuracy and cross-validation scores. They find "Status" strongly correlates with academic scores and work experience. Animesh et al. [3] address placement challenges for engineering students, achieving 78.57% accuracy with an integrated model, outperforming Logistic Regression and Support Vector Machine. This method shows broader applicability in engineering domains.

Vikas et al. [4] combine Decision Trees, SMOTE, Fuzzy Rules, Clustering Techniques, and Naïve Bayes, considering academic and non-academic factors. They achieve 87.50% accuracy with a 70/30 Decision Tree model and suggest future feature expansion. Nikhil et al. [5] apply various techniques to predict computer science student placements, providing insights for students and institutions and supporting decision-making.

Shahane [6] applies Logistic Regression and other methods, achieving 95.34% accuracy and suggesting deep learning and cross-validation. Jakub et al. [7] use Recurrent Neural Networks with NAdam, reaching 92.42% accuracy and proposing early-age success prediction. Both studies show the potential of machine learning for education and career outcomes.

Reham and Ayed [8] use logistic regression, random forest, and neural network, with the neural network achieving 83.2% accuracy but longer training times. Pornthep and Pokpong [9] use data mining and Random Forest to guide university students in making academic and career choices, improving satisfaction and goal setting. Navuluri et al. [10] develop a placement prediction system with a user-friendly GUI, demonstrating superior algorithm performance.

Jumana and Senthil [11] illustrate exploratory data analysis on placement data, focusing on missing data handling and visualizing factors affecting placements. They compare Decision Tree and Random Forest algorithms, with Random Forest having higher classification accuracy. Arshdeep et al. [12] cover data science applications, skills, cloud computing, salary variations, and gender pay gap. They explore pandemic-induced salary disparities, underscoring the rising demand for data science professionals.

3 Dataset Description & System Flow

Data is pivotal in constructing machine learning (ML) models, necessitating attention to tasks like acquisition, feature engineering, and comprehensive pre-processing. This involves operations on structured CSV data, addressing missing values, scaling, identifying outliers, and examining input-output feature correlations for effective model identification.

The education-focused dataset includes subordinate and higher preparatory school segments, along with business-related aspects like business scope, work knowledge types, and salaries. Table 1 outlines 15 attributes: SL NO, GENDER, SSC_P, SSC_B, HSC_P, HSC_B, HSC_S, DEGREE_P, DEGREE_T, WORKEX, ETEST_P, SPECIALIZATION, MBA_P, STATUS, and SALARY. Figure 2 visually depicts attribute correlations, showing how changes in one attribute affect others in the dataset.

Attribute	Datatype	Description
sl_no	Integer	The unique ID for each record
gender	String	The gender of the candidate
ssc_p	Float	The percentage of candidate results in SSC Board
ssc_b	String	The Board of the SSC (Central or Other)
hsc_p	Float	The percentage of candidate results in HSC Board
hsc_b	String	The Board of the HSC (Central or Other)
hsc_s	String	The Stream of the candidate in HSC Board
degree_p	Float	The percentage of candidates result in a bachelor's degree
degree_t	String	The type of degree the candidate has completed
workex	String	Whether the candidate has any work experience or not
etest_p	Float	The percentage of e-test results that the candidate has given
specialization	String	In which field candidate has done specialization (Marketing & HR or Marketing & Finance)
mba_p	Float	The percentage of candidate's MBA Result
status	String	The Status of the candidate's employment (Placed or Not Placed)
salary	Integer	The Salary of the placed candidate

Table 1. Dataset Description

Figure 1 outlines a systematic approach for preparing the dataset for ML algorithms. The steps include crucial procedures like data gathering, feature engineering, and pre-processing, all contributing significantly to the foundational stages of ML model development.

Fig. 1. Overall System Flow

In Figure 1, data collection began with a CSV dataset from kaggle.com [1], serving as the foundational source. The focus was systematic data deciphering and cleaning for reliability in subsequent analysis. Crucial steps involved addressing NaN values in rows to prevent interference with the model's predictive accuracy due to missing information.

Fig. 2. Attribute Correlation Heatmap of the employed dataset

3.1 Exploratory Data Analysis

Post data gathering and cleansing, exploratory data analysis (EDA) became pivotal, utilizing diverse tools for visual representation. It supplied a comprehensive summary of findings, enabling the interpretation of data, identification of patterns, and trend discovery. Structured data analysis involved a detailed examination of column and row numbers, understanding data structure, and pinpointing crucial columns for analysis.

3.2 Preprocessing Techniques

In preprocessing, removed 'sl_no' and 'status,' filled missing 'salary' values with zeros, and filtered outliers. Also, replaced missing 'Salary' attribute values with zeros. Categorical attributes (gender, SSC_b, HSC_b, WorkEx, Specialization, HSC_s, Degree_t, and Status) underwent One-Hot Encoding using Pandas, streamlining training with a binary matrix and vector, as shown in Table 2.

3.3 Integration of the dataset with the Model

After preprocessing, the dataset integrates directly into the placement prediction model, categorizing instances as 'Placed' or 'Not Placed.' 'Placed' entries trigger the salary prediction model to calculate candidates' salaries. Comparative analysis of results and accuracies across models identifies the most effective ones for production deployment, covering both placement and salary predictions.

	gender	ssc_b	hsc_b	hsc_s	degree_t	workex	specialization	status
2	-	-	-	Science	Sci & Tech	-	-	-
1	Male	Central	Central	Commerce	Comm & Mgmt	Yes	Mkt & HR	Placed
0	Female	Others	Others	Arts	Others	No	Mkt & Fin	Not Placed

Table 2. One Hot Encoding of Categorical Attributes

In conclusion, our systematic approach, including data cleansing, exploratory data analysis, and preprocessing, established a robust foundation for model development. This method ensured the extraction of meaningful information from the dataset, culminating in the selection of an optimal model for real-world applications.

4 Methodology

In Figure 3, our methodology begins with soliciting details from candidates. Table 3 outlines various models, each processing candidate information to yield a binary outcome: "Placed" or "Not Placed." A maximum voting rule, with assigned weights, determines the collective decision. If "Placed," the application proceeds to salary estimation using Table 4 models, employing another maximum voting rule for the final calculated salary. This information is then presented to the user.

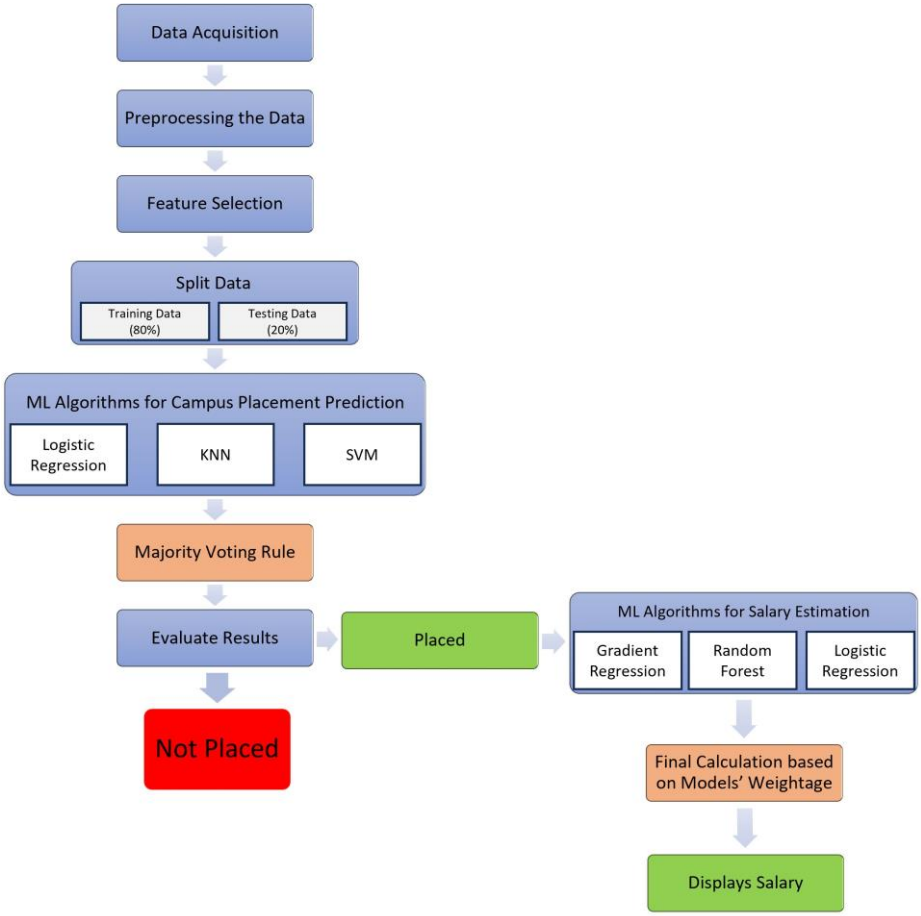


Fig. 3. The Architecture Diagram

4.1 Models Utilized in the Prediction of Campus Placements

Logistic Regression
Logistic regression, ideal for binary classification, establishes a linear decision boundary from probability estimates. Results are expressed in a nonlinear parameterized form, optimized through solvers. This method accommodates both quantitative and qualitative inputs, using the logistic function to compute outcomes within the 0 to 1 range. Table 3 displays attributes importance in the Logistic Regression Model.

K – Nearest Neigbours
K-nearest neighbors (KNN), a supervised learning algorithm, classifies data by calculating distances, usually via Euclidean Distance. It selects the nearest k neighbors based on similarity, and their majority vote determines the label for testing data. This versatile

method works for both classification and regression problems, assuming related entities are close. Table 3 emphasizes attribute importance in the KNN Model.

Support Vector Machine
Support Vector Machine (SVM) excels in binary classification and regression in n-dimensional space, implemented through Python's Scikit-learn. Adapting to diverse machine learning algorithms, the model ensures precise mapping of new data points, ensuring reliability for well-informed classifications. Table 3 highlights attribute importance in the SVM Model.

Attributes given in the dataset	Importance of the Attributes considered by each model		
	Logistic Regression	KNN	SVM
gender	4.2	2.3	4.5
ssc_p	7.2	6.9	6.2
ssc_b	3.1	2.3	3.4
hsc_p	7.3	6.7	6.3
hsc_b	3	2.3	3.2
hsc_s	6.5	4.6	4.2
degree_p	7.3	6.5	8.8
degree_t	5.3	6.3	6.3
workex	1.9	2.3	6.1
etest_p	0.4	0.0	4.3
specialization	6.5	2.3	6.5
mba_p	2.8	4.3	6.2

Table 3. Feature Importance of the attributes considered by each model.

4.2 Implementing Majority Voting Rule in Campus Placement Prediction with proposed models

In this context, individual importance is not assigned to each model, as observed in the Salary Estimation Model. Instead, a straightforward maximum function is applied to the outputs of three models, culminating in the final classification as 'Placed' or 'Not Placed,'.

4.3 Salary Estimation Models for Placed Candidates

After the campus placement prediction model produces a 'placed' output, the subsequent phase involves the activation of the salary estimation model to compute the salary for the placed candidate. The employed models in salary estimation are as follows:

Random Forest
Random Forest collaborates with SVM, Logistic Regression, Decision Tree, and Naive Bayes, using decision trees where nodes represent feature tests and leaf nodes signify class labels. Multiple uncorrelated trees enhance predictive accuracy through collaborative voting. In salary estimation, Random Forest applies diverse decision trees to

dataset subsets for improved accuracy. Its robust capability in handling varied attributes provides a precise model for predicting salary values, highlighted in Table 4.

Gradient Regression
The Gradient Regression model, utilizing XGBoost, predicts salaries based on diverse dataset features. With Python's Scikit-Learn module, data is prepared, loaded, and organized into a multidimensional array. A 70-30 split is applied for training and testing data, where 70% is used for training and the remaining 30% for testing. Table 4 underscores attribute importance of the dataset for the Gradient Regression Model.

Logistic Regression
Logistic Regression serves as a pivotal model for salary estimation, leveraging a straightforward approach to capture the relationship between various features in the dataset and the corresponding salary values. Table 4 shows the importance of attributes in the dataset considered by the Logistic Regression Model.

4.4 Model Weightage Allocation

Differing from traditional models, this approach assigns weights based on individual accuracies, not just majority voting. The Gradient Regression model, known for performance, gets a substantial 70%, while Random Forest and Logistic Regression each receive 15%. This nuanced weighting acknowledges model strengths, refining the prediction framework for increased accuracy.

Attributes given in the dataset	Importance of the Attributes considered by each model		
	Gradient Regression	Random Forest	Logistic Regression
gender	2.2	0.3	0.3
ssc_p	8.2	1.4	-2.3
ssc_b	2.1	0.2	-0.2
hsc_p	8.1	1.3	-0.9
hsc_b	2.0	0.2	2.5
hsc_s	7.6	0.2	2.4
degree_p	8.3	1.4	1.6
degree_t	7.5	0.2	1.9
workex	6.9	0.3	-6.1
etest_p	0.1	1.1	0.6
specialization	3.5	0.2	0.2
mba_p	0.3	1.1	3.7
status	9.7	1.2	4.4

Table 4. Feature Importance of the attributes considered by each model.

5 Results & Analysis

Three models were implemented in VS Code using Python with essential libraries like Scikit-Learn, Matplotlib, Keras, and TensorFlow. These models underwent training and

prediction phases, utilizing the prediction function for forecasting based on sample testing input. Results, along with test sample output, generated accuracy, confusion matrix, and a comprehensive classification report for each model. The classification report function provided precision, recall, f1-score, and support metrics, including average, macro avg, and weighted avg values (detailed in Tables 5 and 6).

	Logistic Regression	KNN	SVM
Precision	88%	78%	75%
Recall	88%	79%	77%
F1 – Score	88%	77%	74%
Support	43	43	43
True Positive	29	29	29
True Negative	9	5	4
False Negative	3	7	8
False Positive	2	2	2
Cross Validation Score	83.8%	81.3%	83.3%
Accuracy	84.1%	81.3%	83.25%

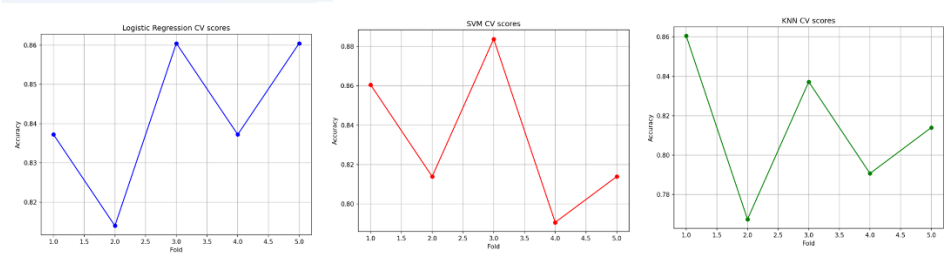
Table 5. Performance Metrics of Classification Models

Tests on the college's placement data involved an 80% training set and a 20% cross-validating testing set (Figure 4 and 5). Input data for the models contributed to metrics and accuracies in Table 5, predicting class based on various attributes. Figure 4 displays cross-validation graphs, and Table 5 outlines accuracy: Logistic Regression at 84%, KNN at 81%, and SVM at 83%.

	Gradient Regression	Random Forest	Logistic Regression
Precision	76%	66.84%	63%
Recall	76.35%	66%	64.5%
F1 – Score	74.87%	60.1%	63.4%
Support	43	43	43
True Positive	29	23	20
True Negative	8	11	7
False Negative	3	5	4
False Positive	1	1	2
Cross Validation Score	69.34%	59%	53%
Accuracy	78.6%	67.39%	65.67%

Table 6. Performance Metrics of Estimation Models

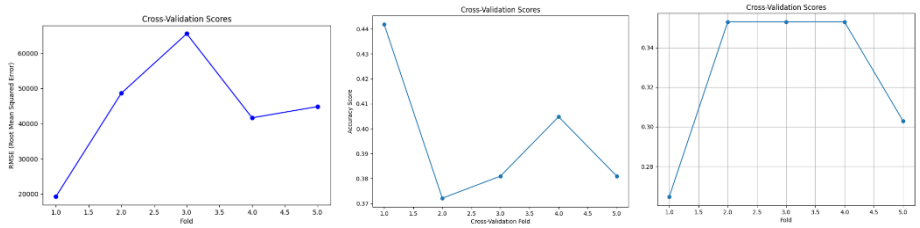
Following the results of the campus placement prediction models, the subsequent phase involved salary estimation using models detailed in Table 6. Gradient Regression Model demonstrated an accuracy of 79%, Random Forest Model exhibited an accuracy of 68%, and Logistic Regression Model displayed an accuracy rate of 66%.



(a) Logistic Regression Model (b) SVM Model (c) KNN Model

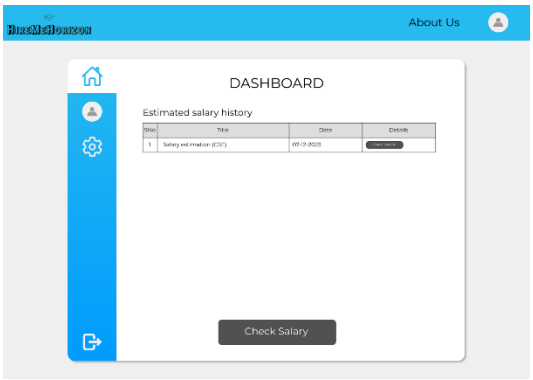
Fig. 4. Cross Validation Graphs of Classification Models

In addressing the utilization of the developed ML algorithms within the interface, a website was constructed from the ground up. Moving forward to Figure 6(a), the focal point is the main dashboard, providing users with a comprehensive view of previously predicted results for monitoring purposes. In Figure 6(b), users are prompted to submit details for placement prediction, and Figure 6(c) displays the ensuing results. Here, a dialog box emerges, indicating the candidate has been placed, accompanied by the corresponding estimated salary.

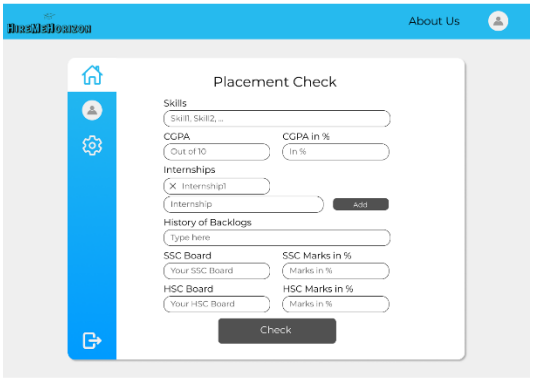


(a) Gradient Regression Model (b) Random Forest Model (c) Logistic Regression Model

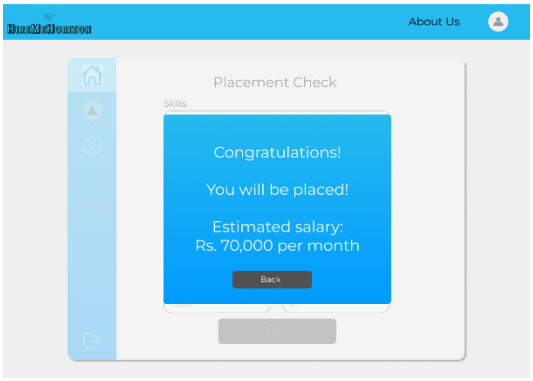
Fig. 5. Cross Validation Graphs of Estimation Models



(a) Dashboard



(b) Entering Details for Placement Prediction



(c) Prediction Results with Estimation Results

Fig. 6. Interface for the Campus Placement and Salary Prediction System

6 Conclusion & Future Scope

This paper introduces a comprehensive campus placement and salary prediction system using data mining techniques tailored for the organization's data source. The research explores data mining classifier algorithms' potential utility for the placement cell of a Higher Education Institution (HEI), predicting student placements during campus recruitments and enhancing institutional responsibilities.

Emphasizing the significance of unplaced students' records, the paper provides insights into robust and fragile aspects of campus placement activities, contributing to refining the entire training and placement process. While acknowledging results are based on a specific dataset, the model's adaptability is highlighted, suggesting its applicability with other supervised classifiers for enhanced analysis and accuracy.

Future work includes expanding datasets with additional attributes (e.g., co-curricular activities, certification status, summer training, demographics, and twenty-first-century skills) and exploring the impact of combining different machine learning models on accuracy across various engineering branches. Considerations involve developing Android and desktop/GUI apps for user-friendly predictions. With user consent, the entered details will contribute to dataset and model training for improved accuracy, acknowledging limitations of a limited dataset and reliance on statistical rather than behavioral data. This integrated approach aims to continually enhance machine learning algorithms' predictive capabilities in campus placements.

References

1. https://www.kaggle.com/datasets/kislaymishra/placement-dataset
2. Sachin, C.H. Patil, Surabhi Thatte, Vikas and Poonam," A Data-Driven Probabilistic Machine Learning Study for Placement Prediction", 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT).
3. Animesh, M Vignesh, Bysani and Naini," A Placement Prediction System Using K-Nearest Neighbors Classifier", 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP).
4. Vikas, Shikha, Ruchi and Varun," Applying SMOTE with Decision Tree Classifier for Campus Placement Prediction", 2021 International Conference on Computing, Communication and Green Engineering (CCGE).
5. Nikhil, Ajay, Thirunavukkarasu and E Rajesh," Campus Placement Predictive Analysis using Machine Learning", 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN).
6. Priyanka Shahane," Campus Placements Prediction & Analysis using Machine Learning", 2022 International Conference on Emerging Smart Computing and Informatics (ESCI).
7. Jakub, Micha l and Marcin," Future Graduate Salaries Prediction Model Based on Recurrent Neural Network", Proceedings of the Federated Conference on Computer Science and Information Systems.
8. Reham and Ayed," Machine Learning Models for Salary Prediction Dataset using Python", 2022 International Conference on Electrical and Computing Technologies and Applications (ICECTA).

9. Pornthep and Pokpong," Random Forest for Salary Prediction System to Improve Students' Motivation", 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems.
10. Navuluri, Sravya and Rajalakshmi," Student Placement Analysis using Machine Learning", 2023 8th International Conference on Communication and Electronics Systems (ICCES).
11. Jumana and Senthil," Utilizing Exploratory Data Analysis for the Prediction of Campus Placement for Educational Institutions", 2020 11th International Conference on Computing Communication and Networking Technologies.
12. Arshdeep, Deval, and Navneet," Utilizing Quantitative Data Science Salary Analysis to Predict Job Salaries", 2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT)