**STUDENT PLACEMENT PREDICTION USING MACHINE LEARNING**

**A Project Work Report**

***Submitted in Partial Fulfillment for the Award of the Degree Of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND BUSINESS SYSTEMS**

**Submitted by**

**MIRIYALA LOKESH PHANI KUMAR 21B91A5735**

**KAPISETTY BHANU SANDEEP 21B91A5721**

**B Y VAMSI KRISHNA 21B91A5705**

**Under the esteemed guidance of**

**Dr. P Ravi Kiran Varma**

**(Professor and HOD)**

****

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE (AUTONOMOUS)**

(Approved by AICTE, New Delhi, Affiliated to JNTU University, Kakinada)

**CHINNA AMIRAM :: BHIMAVARAM-534202**

**April - 2025**

**SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE (AUTONOMOUS)**

**(Approved by AICTE , New Delhi, Affiliated to JNTU UNIVERSITY,KAKINADA}**

**CHINNA AMIRAM, BHIMAVARAM-534204**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

****

**Certificate**

**This is to certify that the Project Work report entitled "STUDENT PLACEMENT PREDICTION USING MACHINE LEARNING,”, is Bonafide work submitted by MIRIYALA LOKESH PHANI KUMAR (Regd.No: 21B91A5735), KAPISETTY BHANU SANDEEP (Regd.No: 21B91A5721), B Y VAMSI KRISHNA (Regd.No: 21B91A5705) in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Business Systems during the Academic year 2024-2025.**

**HOD GUIDE**

Dr. P RAVI KIRAN VARMA Dr. P RAVI KIRAN VARMA

HOD-IT Dept HOD-IT Dept

**CERTIFICATION OF EXAMINATION**

This to certify that I have examined the concept and hereby accord my approval of it as a Project Work entitled “**STUDENT PLACEMENT PREDICTION USING MACHINE LEARNING**” carried out and presented in a manner required for its acceptance on partial fulfillments for the award of the degree of BACHELOR OF TECHNOLOGY in Computer Science and Business Systems for which it has been submitted.

This approval does not necessarily endorse or accept every statement made opinion expressed or conclusions drawn as recorded in the Project Work report it only signifies the acceptance of the report for the purpose for which submitted.

**Signature**

**Project Guide :**

**Senior Faculty :**

**External Examiner :**

**HOD :**

**DECLARATION**

This Project Work report entitled “**STUDENT PLACEMENT PREDICTION USING MACHINE LEARNING**” has been carried out by us in the partial fulfillment of the requirements for the award of the degree of B.Tech (CSBS), SAGI RAMA KRISHNAM RAJU Engineering College(A). We hereby declare this project work/project report has not been submitted to any of the other university/Institute for the award of any other degree/diploma.

Name Register No Signature

MIRIYALA LOKESH PHANI KUMAR 21B91A5735

KAPISETTY BHANU SANDEEP 21B91A5721

BOBBA YASWANTH VAMSI KRISHNA 21B91A5705

**ACKNOWLEDGEMENT**

The victorious completion of this project would be incomplete without greeting those who made it possible and whose guidance and encouragement made efforts that taken to the success.

The thesis presented here is the work accomplished under the enviable and scholarly guidance of **Dr. P RAVI KIRAN VARMA**, Professor of Information Technology Department. We are grateful for the indomitable support. We express deep gratitude for her inspiring remarks and for helping with this project.

We take this opportunity to express our sincere thanks to **Dr. P RAVI KIRAN VARMA**, Head of the Department, Information Technology, for his support and encouragement.

We express our sincere thanks to **Prof. K V Murali Krishnam Raju**, PRINCIPAL, SAGI RAMA KRISHNAM RAJU Engineering College, Bhimavaram for giving us this opportunity for the successful completion of this project.

Finally, we also express our sincere thanks to other **Teaching and non-Teaching staff** for their support in accomplishing this project.

1. MIRIYALA LOKESH PHANI KUMAR 21B91A5735
2. KAPISETTY BHANU SANDEEP 21B91A5721
3. B Y VAMSI KRISHNA 21B91A5705

**ABSTRACT**

Institutions rely on student placement prediction to evaluate employability through academic and non-academic student attributes. An analysis based on machine learning methods aims to forecast student placement outcomes as well as salary package potential. This dataset contains features organized under three categories: academic features including CGPA and project count and technical skills and internships and extracurricular features consisting of certifications and hackathon participation as well as personal features such as communication skills leadership aptitude scores and physical fitness. The analysis employed Logistic Regression and Random Forest along with SVM and ExtraTree Classifier and Gaussian NB and Decision Tree and KNN as classification tools. Feature selection involved utilizing Chi-Square Test and Mutual Information along with Recursive Feature Elimination (RFE) and Lasso (L1 Regularization) and Random Forest Embedded Method to boost model performance. Experimentation took place on 70% training data combined with 30% testing data using models that included feature selection and those without feature selection. The Random Forest model demonstrated placement prediction accuracy of 96% when using and not using feature selection and achieved 96.03% in salary prediction without feature selection but dropped to 92.85% with Chi-Square feature selection. A web application incorporating placement forecasting together with resume assessment and career readiness guidance was built to help students plan their careers.

**Table of Contents**

**NAME OF THE CHAPTER PAGE NO**

**List of Figures i-ii**

**List of Tables iii-iv**

**1 INTRODUCTION** 1–2

1.1 Placement Prediction using Machine Learning 1

1.2 Our Major Contributions for this Study on Student Placement

Prediction System 2

**2 LITERATURE SURVEY** 3-8

**3 METHODOLOGY** 9-49

3.1 Data Collection 9-11

3.2 Data Preprocessing 11-13

3.2.1 Handling Categorical Data 11-13

3.3 Model Training and Evaluation 14-23

3.3.1 Binary Classification for Campus Placement Prediction 14-18

3.3.2 Multiclass Classification for Package Prediction 18-23

3.4 Impact of Feature Selection on Binary and Multiclass

Classification Performance 23-49

3.4.1 Enhancing Placement Prediction for Binary Classification

of Placed or Not with Feature Selection Techniques 23-35

3.4.2 Enhancing Placement Prediction for Multiclass Classification

of Package Categories with Feature Selection Techniques 35-49

**4 MACHINE LEARNING ALGORITHMS** 50-51

4.1 Logistic Regression 50

4.2 Decision Tree 50

4.3 Random Forest Classifier 50

4.4 Extra Tree Classifier 50

4.5 K Neighbours Classifier 50

4.6 Support Vector Machine 51

**5 IMPLEMENTATION AND PERFORMANCE EVALUATION** 52-53

5.1 Implementation of model 52

5.2 Performance Evaluation Metrics 52

**6 Conclusion** 54

**References** 55-56

**List Of Figures**

**S. NO Figure No.** **Name of the Figure**  **Page No.**

1 Fig 3.1 Process Flow Diagram 21

2 Fig 3.2 Placement Percentage by originality of final

year project 39

3 Fig 3.3 Placement Percentage by habit of asking doubts 40

4 Fig 3.4 Placement Percentage by medium of study 41

5 Fig 3.5 Placement Percentage by number of siblings 42

6 Fig 3.6 Placement Percentage by receiving stipend during

final year internship 43

7 Fig 3.7 Placement Percentage by interview confidence

level 44

8 Fig 3.8 Placement Percentage by hackathons participated

during btech 45

9 Fig 3.9 Placement Percentage by knowledge and skills in

coding and programming 46

10 Fig 3.10 Placement Percentage by problem solving and coding

during btech 47

11 Fig 3.11 Placement Percentage by engaging in online gaming

during btech 48

12 Fig 3.12 Accuracy Comparison Before and After Feature

Selection 51

13 Fig 3.13 Salary distribution based on self-rating in technology 57

14 Fig 3.14 Salary distribution based on stipend received during

Final year internship 58

15 Fig 3.15 Salary distribution based on number of siblings 59

16 Fig 3.16 Salary distribution based on number of hackathons

won during BTech 60

17 Fig 3.17 Salary distribution based on Online Gaming

participation during BTech 61

18 Fig 3.18 Salary distribution based on Medium of Study 62

19 Fig 3.19 Salary distribution based on Hackathons Participated 63

20 Fig 3.20 Salary distribution based on Merit Scholarship 64

21 Fig 3.21 Salary distribution based on self-rated communication

Skills 65

22 Fig 3.22 Salary Distribution based on Self-Rated Coding Skills 66

23 Fig 3.23 Salary Distribution based on confidence in final

year project originality 67

24 Fig 3.24 Accuracy comparison before and after feature selection 72

**List Of Tables**

**S. NO Table No.** **Name of the Table**  **Page No.**

1 Table 3.1 List of Questionnaires 13-14

2 Table 3.2 Year-wise Branch Distribution of Students

for Placement Analysis 15

3 Table 3.3 sample data of 1-12 from collected data 17

4 Table 3.4 sample data of 13-24 columns from collected

data 18

5 Table 3.5 sample data of 25-36 columns from collected

data 19

6 Table 3.6 sample data of 25-36 columns from collected

data 20

7 Table 3.7 classification output on placement prediction

without feature selection 26

8 Table 3.8 classification output on package prediction

without feature selection 31

9 Table 3.9 confusion Matrix for logistic regression 31

10 Table 3.10 confusion Matrix for logistic regression 32

11 Table 3.11 confusion Matrix for logistic regression 32

12 Table 3.12 confusion Matrix for extra tree classifier 33

13 Table 3.13 feature selection results table 38

14 Table 3.14 Results obtained based on Logistic Regression

with L1 Regularization feature selection method

for placement prediction 50

15 Table 3.15 Feature selection on Package Prediction Result 57

16 Table 3.16 Results obtained based on Logistic Regression

with L1 Regularization feature selection method

for package prediction 69

17 Table 3.17 Confusion Matrix for Logistic Regression 70

18 Table 3.18 Confusion Matrix for Decision Tree Classifier 70

19 Table 3.19 Confusion Matrix for Random Forest Classifier 71

20 Table 3.20 Confusion Matrix for Random Forest Classifier 71

**Chapter 1**

**INTRODUCTION**

**1.1 Placement Prediction using Machine Learning:**

Machine learning applications deliver student placement prediction as a critical use case which allows educational institutions to evaluate student information for job placement predictions. The technology functions as a key instrument which helps career counseling services by using information about academic achievements and skills to enhance student hireability potential. Educational institutions use predictive modeling technology to predict student job placement outcomes which enhances recruitment decisions and accelerates hiring practices while connecting job seekers to available opportunities thus enhancing total placements results.

The method of predicting student placement has made substantial progress beyond approved assessment techniques. Educational institutions used to base their employability assessment on academic performance and interview evaluations while both methods required significant time without delivering clear results. Early predictive models reviewed each student independently by concentrating exclusively on limited input variables. Modern placement prediction systems powered by machine learning methods perform a detailed analysis that assesses academic results through CGPA and EAMCET rank and technical competence by measuring coding ability and certification attainment as well as hackathon performance along with physical activities and faculty engagement and public speaking activities. A system that combines the evaluation of many students at once through varied aspects delivers both improved predictions and faster placement program administration and useful guidance for career choices.

Student placement prediction systems in current times integrate multiple components into a unified system design. Different machine learning prediction models undertake placement and salary estimation tasks using Logistic Regression, Random Forest, Decision Tree, Support Vector Machines (SVM with Linear, Polynomial, RBF) and ExtraTree Classifier and Gaussian Naïve Bayes (Gaussian NB).Predictive accuracy of the system increases through a systematic preprocessing workflow.

Label Encoding operates on categorical data by transforming their values into numbers.The significant features for our system are discovered through combinations of Chi-Square Test, Mutual Information, Recursive Feature Elimination (RFE), Lasso (L1 Regularization) and the Random Forest Embedded Method .Min-Max Scaling transforms all numeric attributes between 0 to 1 to create better models so data spreads evenly across the range .The dataset divides information into three main sections .Academic Features: CGPA, project count, EAMCET rank, backlogs. Technical and extracurricular attributes of the program include certifications together with hackathons and internships while coding abilities and technical competition events complete the set. The evaluation includes assessment of communication abilities along with leadership potential and interview skills and how students interact with faculty staff and their social media habits. A multistep process applies machine learning methods together with feature engineering methods and data normalization techniques to process student attributes and increase both prediction effectiveness and meaningful placement outcome and salary prediction capabilities.

### **1.2 Our Major Contributions for this Study on Student Placement Prediction System:**

1. Design of a predictive model that optimizes the assessment of placement probabilities and salary packages through accurate predictions using simplified computational methods.
2. Sophisticated attribute selection methods are employed to maintain essential features, enhancing system precision and reliability.
3. Development of a complete dataset that classifies student traits into academic, technical, and personal characteristics, creating an all-encompassing evaluation system.
4. A responsible web application development process merges several services into one integrated system:

* Placement Prediction – Assists in forecasting student job acquisition outcomes.
* Package Prediction – Estimates the expected salary students can receive.
* Resume Quality Assessment – Identifies improvement opportunities in resumes.
* Placement Preparation Guide – Provides insights and resources for interview readiness.

The research aims to establish a highly efficient and scalable student placement forecasting system by eliminating redundant attributes (Name, Department, Mobile Number) through manual exclusion and feature selection techniques. Through its design, the system delivers precise job placement forecasts using a clean and student-friendly interface.

**Chapter 2**

**LITERATURE SURVEY**

Priyadarsini et al., [1] a study titled "Placement Prediction Using the Artificial Neural Network (ANN)." This study involved students from Presidency College, Bengaluru, comprising an undisclosed number of samples and multiple attributes, including gender, academic records, coding scores, aptitude test results, group discussion performance, extracurricular activities, and soft skills. An Artificial Neural Network (ANN) model was employed for binary classification to predict student placement outcomes. The model was implemented using TensorFlow and Keras, incorporating multiple layers with ReLU and sigmoid activation functions, along with dropout layers for regularization, and optimized using Adam and SGD optimizers. The study achieved a maximum prediction accuracy of 97%, with continuous training over 500 epochs and fine-tuning through hyperparameter optimization and k-fold cross-validation. Although no explicit feature selection or dimensionality reduction technique like PCA was used, preprocessing techniques such as normalization and encoding were applied to enhance model performance. The purpose of this research was to help students identify areas for improvement and to support institutions in making data-driven placement decisions. The study highlights several advantages, such as high prediction accuracy, model adaptability, and effective handling of both categorical and numerical data. However, it also notes disadvantages, including dependency on the accuracy of self-reported aptitude scores, risk of overfitting without proper regularization, and the lack of interpretability often associated with deep learning models.

Ragu et al., [2] a study titled "Historical Data-Driven Placement Analysis Using Explainable Artificial Intelligence." This study involved the analysis of 1,009 student records with 15 attributes, including secondary and higher secondary percentages, degree specialization, coding and aptitude scores, work experience, and placement status. Multiple machine learning algorithms—K-Nearest Neighbors (KNN), Naive Bayes, Logistic Regression, and Support Vector Machine (SVM)—were applied to build predictive models. Among these, the Support Vector Machine (SVM) algorithm achieved the highest accuracy of 92%, followed by Logistic Regression (91%), KNN (86%), and Naive Bayes (83%). SHAP (SHapley Additive exPlanations) was employed for feature attribution and model interpretability, allowing a deeper understanding of the key features influencing placement predictions, with coding and aptitude scores identified as major factors. The purpose of this research was to enhance placement strategies through transparent AI systems capable of explaining the reasoning behind predictions, thus aiding institutions in aligning academic training with industry requirements. The study highlights several advantages, such as high model accuracy, explainability of predictions, and data-driven identification of critical skills and attributes. However, it also notes disadvantages, including dependency on the quality of input data, potential issues related to class imbalance, and the necessity for careful feature preprocessing and interpretation to prevent overfitting or misinterpretation.

Maragatham et al., [3] conducted a study titled "Student Placement Prediction using Deep Learning Techniques." This study utilized data from Kongu Engineering College’s placement cell, containing various attributes such as SSLC Percentage, HSC Percentage, Number of Standing Arrears, Skill Sets, Skills Lagging, Internship Experiences, and Career Aspirations. The study employed advanced deep learning algorithms including Feedforward Neural Networks (FNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM). While no explicit dimensionality reduction techniques like PCA were used, data preprocessing steps such as duplicate removal, handling missing values, and label encoding were performed to ensure model readiness. Among the models, LSTM achieved the highest prediction accuracy of 99.2%, followed by RNN at 97.4%, and FNN at 91.7%. The purpose of this research was to develop a robust and intelligent predictive framework capable of analyzing diverse student data and accurately forecasting placement outcomes, thereby empowering institutions to provide personalized career guidance and optimize placement strategies. The study’s advantages include exceptional accuracy, the ability to capture both temporal and structured patterns, scalability across larger datasets, and personalized support for students. However, disadvantages noted include lack of dimensionality reduction techniques, potential interpretability issues with deep learning models, and higher computational resource requirements.

Krishna et al., [4] conducted a study titled "Comprehensive Career Placement Predictor: An Analytical Tool for Optimizing Job Placement Outcomes." This study involved more than 500 students and considered multiple attributes, including 10th and 12th-grade marks, CGPA, Backlogs, Attendance, Communication Skills, Programming Skills, Verbal and Aptitude Test Scores, Certifications, Internship Experiences, Soft Skills, and Personal Preferences. The study utilized various machine learning algorithms, including Logistic Regression, Random Forest, Decision Tree, K-Nearest Neighbors (KNN), XGBoost, AdaBoost, and Gradient Boosting. Principal Component Analysis (PCA) was applied in relevant experiments for dimensionality reduction to extract the most influential features, and the Random Forest algorithm achieved a prediction accuracy of 97%, while Logistic Regression alone reached 87%. The purpose of this research was to forecast student placement chances based on comprehensive academic and skill-based profiles to enhance employability, guide skill development, and align student capabilities with industry expectations. The study highlights several advantages, such as delivering personalized career guidance, improving prediction accuracy, identifying skill gaps, and enabling data-driven institutional decision-making. However, it also notes disadvantages, including dependency on data quality and diversity, the need for consistent updates, and the possibility of bias or fairness issues in algorithmic outcomes.

Gaurkhede et al. [5], in a study titled “Predictive Modeling and Analysis of Campus Placements Using Machine Learning Techniques”, utilized a dataset of B.Tech students from 2023 containing multiple academic and professional attributes such as CGPA, SSC and HSC percentages, specialization, certifications, internship experience, project work, placement status, and salary. The study employed five machine learning algorithms—Logistic Regression, Decision Tree, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest—to predict placement outcomes. As part of feature selection and dimensionality reduction, exploratory data analysis (EDA) and correlation matrix analysis were used to identify the most impactful attributes influencing placement success. The model performance was evaluated using accuracy, precision, recall, and F1-score. Among the models, Decision Tree achieved the highest accuracy of 98.46%, followed closely by KNN at 96.92% and Random Forest at 95.38%, while Logistic Regression and SVM lagged with 80% and 67.69% accuracy, respectively. The purpose of the study was to enhance institutional placement strategies and student employability by providing predictive insights based on historical data and skill indicators. The key advantages include high model accuracy, identification of critical placement features, and actionable feedback for curriculum development. However, disadvantages include potential model overfitting, dependency on high-quality input data, and limited generalizability if applied outside the original institutional context.

Hegde et al., [6] a study titled "Predicting Student Placement using PCA and Machine Learning Technique." This study involved more than 500 students and considered nine attributes, including Attendance, Backlog, CGPA, Communication Skills, Programming Skills, Verbal Test, Aptitude Test, Teacher Remarks, and NPTEL Certificates. Principal Component Analysis (PCA) was used for dimensionality reduction to identify the key features influencing student placement, followed by the application of the Random Forest algorithm, which achieved an accuracy of 97%. The purpose of this research was to predict the placement of students based on their profiles, qualifications, and performance to aid academic success and skill improvement. The study highlights several advantages, such as improving prediction accuracy and efficiency, identifying key factors influencing placements, and providing actionable insights for students and institutions. However, it also notes disadvantages, including dependency on the quality and relevance of input data and potential issues with bias and fairness in predictions.

Kadu et al., [7] a study titled "Student Placement Prediction and Skill Recommendation System using Machine Learning Algorithms." This study utilized machine learning techniques to predict student placements and recommend skills for improvement based on eight features: CGPA, skills, internships, projects, certifications, soft skills, extracurricular activities, and placement status. Machine learning algorithms including Random Forest, Decision Tree, and Gaussian Naive Bayes were applied, with Random Forest achieving the highest accuracy of 79.33%. The purpose of this research was to assist students in enhancing their placement readiness by identifying necessary skills and improvements through predictive modeling. The study highlights several advantages, such as providing personalized skill recommendations and leveraging historical academic and extracurricular data for prediction. However, it also notes disadvantages, including dependency on data quality, potential biases affecting the recommendations, the inability to capture all possible factors influencing placement outcomes, and limitations in future applicability due to reliance on historical data.

Amzad Basha et al., [8] a study titled "Unraveling Campus Placement Success Integrating Exploratory Insights with Predictive Machine Learning Models." This study explored the use of machine learning models, including Logistic Regression, Support Vector Machine (SVM), Random Forest, XGBoost, Decision Tree, and Naive Bayes, to predict campus placements based on 14 features such as gender, academic scores, degree type, specialization, work experience, placement status, and salary. The researchers employed Exploratory Data Analysis (EDA) and correlation analysis to better understand feature relationships before model development. Logistic Regression and SVM achieved the highest accuracy rates of 88.37%, followed by XGBoost (85.19%) and Random Forest (76.74%). The purpose of this research was to enhance campus placement prediction by integrating exploratory insights with predictive machine learning models, providing institutions with data-driven strategies to improve student outcomes. The study highlights several advantages, including high predictive accuracy for structured data, and actionable insights gained from exploratory analysis. However, it also notes disadvantages, such as dependency on the quality of structured input data and reduced robustness when dealing with unstructured or qualitative information.

Khandelwal et al., [9] a study titled "The Study of Machine Learning Classification Algorithm for Student Placement Prediction." This study analyzed 2,966 student records using eight attributes, including age, gender, stream, internships, CGPA, hostel residence, backlog history, and placement status. Various machine learning classification algorithms such as Decision Tree, Random Forest, Logistic Regression, Gaussian Naive Bayes, and K-Nearest Neighbors (KNN) were utilized, alongside correlation analysis and feature encoding during preprocessing. The key influencing factors identified were CGPA, internships, and backlog history. The models achieved the following accuracies: Decision Tree at 88%, Random Forest at 87%, KNN at 87%, Gaussian Naive Bayes at 78%, and Logistic Regression at 72%. The purpose of this research was to predict student placement status effectively, aiding institutions in better preparing students for recruitment processes. The study highlights several advantages, such as high accuracy with tree-based models and effective identification of major placement factors. However, it also notes disadvantages, including reduced accuracy with Logistic Regression and Gaussian Naive Bayes models, and a strong dependency on the quality and completeness of historical placement data, which could impact the overall model performance.

Rai et al., [10] a study titled "Anticipating Placement Status of Engineering Students using Machine Learning based Prediction Models - A Case Study of Computer Science Stream." This study focused on predicting the placement status of engineering students specifically from the Computer Science stream using machine learning techniques. A dataset comprising 2,656 student records with 12 attributes—including 10th percentage, 12th percentage, current course percentage, attendance percentage, number of placement offers, number of internship offers, and placement status—was analyzed. Machine learning models including Logistic Regression, Decision Tree, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Naive Bayes were tested. Logistic Regression and Decision Tree models achieved 100% accuracy, while SVM, KNN, and Naive Bayes obtained accuracies of 80.14%, 78.78%, and 99.9% respectively. The purpose of this research was to identify key factors influencing placement success and accurately predict placement outcomes for Computer Science engineering students. The study highlights several advantages, such as perfect prediction performance with Logistic Regression and Decision Tree models and valuable insights into factors affecting placements. However, it also notes disadvantages, including comparatively lower accuracy for KNN and SVM models, and the lack of exploration of more advanced machine learning algorithms that could potentially further enhance model performance.

Spandana et al., [11] in "Placement Prediction System using Machine Learning," focused on forecasting the placement status of engineering students by applying machine learning techniques. The study considered more than 250 key attributes, including quantitative aptitude scores, hackathon participation, subject grades, and others. Machine learning models such as Logistic Regression, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Random Forest were used, with preprocessing techniques like scaling, normalization, and dimensionality reduction applied. KNN achieved the highest accuracy of 75.90%, followed by the other algorithms. The study aimed to help students improve their preparation and assist institutions in optimizing their placement strategies. However, the model’s performance is highly dependent on the quality of the data and preprocessing. Disadvantages include challenges in handling imbalanced datasets, which could affect prediction fairness, and the system's accuracy being sensitive to data inconsistencies.

Shahane et al., [12] in "Campus Placements Prediction & Analysis using Machine Learning," aimed to predict the placement likelihood of students using historical data, enabling institutions to better guide students and improve placement strategies. The study used a dataset of 215 records from Kaggle, including SSC percentage, HSC percentage, HSC stream, degree percentage, degree type, work experience, and e-test percentage. Machine learning models such as Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN), and Random Forest were applied, with features visualized to identify correlations with placement outcomes. Logistic Regression achieved the highest accuracy of 95.34%, followed by KNN (90.69%), Random Forest (88.67%), and Decision Tree (83.72%). Missing values were handled, and categorical data was mapped to numerical equivalents. However, the model's accuracy is influenced by the choice and quality of input features. Disadvantages include limited generalization to unseen scenarios without retraining on new data and the model’s performance being sensitive to the quality of input features.

Jadhav et al., [13] in "Placement Readiness Check: Predicting Placement Status Using Supervised ML Methods," focused on predicting campus placement chances and salary ranges to help students assess their readiness and encourage focused preparation. The study utilized a dataset of 500+ records collected from the institution's placement department, including features like X\_score, XII\_score, UG\_score, Aptitude\_score, Coding\_score, Domain\_score, and WrittenEnglishTest\_score. The study applied machine learning models such as XGBoost, Linear Regression, Logistic Regression, SVM, and Random Forest. Preprocessing techniques included handling missing values, standardization, and feature selection based on data correlations.The highest accuracy was achieved by the SVM classifier (80.72%), followed by Logistic Regression (80.12%), XGBoost (79.51%), and Random Forest (78.31%).However, the accuracy of the model is limited by the quality and quantity of input data. Disadvantages include slightly lower predictive performance compared to more advanced models like deep learning techniques, which could potentially yield better results.

Divya et al., [14] in "Student Placement Analysis using Machine Learning," developed a system to predict the likelihood of students being placed in companies based on their academic credentials and historical data, aiming to help students improve their profiles before the hiring process. The study utilized approximately 3,000 student records, including attributes like gender, tenth grade scores, intermediate grade scores, engineering percentage, specialization, and work experience. The study applied existing algorithms such as Support Vector Machine (SVM), Random Forest, and Decision Tree, along with proposed algorithms like Logistic Regression, K-Nearest Neighbors (KNN), and Gradient Boosting Classifier. Entropy and information gain were used to evaluate and select the most relevant features for effective placement prediction. However, the accuracy of the model can vary significantly between different algorithms. Disadvantages include the dependence on historical data, which may not account for future changes in job market requirements.

Surya et al., [15] in "Student Placement Prediction Using Supervised Machine Learning," focused on predicting student placement opportunities using historical data, helping educational institutions enhance their placement strategies. The dataset consisted of approximately 330 student records, including attributes like gender, tenth grade scores, twelfth grade scores, undergraduate scores, technical skills, and coding knowledge. The study implemented algorithms such as Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT), Logistic Regression (LR), and K-Nearest Neighbors (KNN). Feature selection techniques were used to identify key factors, with the primary focus on academic performance and technical skills.However, there was variability in the accuracy across different algorithms, with Random Forest achieving 75%, Decision Tree and Logistic Regression at 68%, and Support Vector Machine at 57%. Disadvantages include that the predictions may not fully account for changes in industry demand or individual student circumstances .

Dhumane et al., [16] a study titled "Machine Learning Approach for Predicting the Placement Status of Students." This study focused on predicting the placement status of final-year B.Tech students using data such as CGPA, academic scores, and attendance. Multiple machine learning algorithms, including Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF), were applied. The authors compared these algorithms based on metrics such as accuracy, precision, and recall. SVM and Logistic Regression achieved the best testing accuracy rates of 81.86% and 82%, respectively. The purpose of this research was to demonstrate the utility of machine learning in optimizing institutional resources, improving employer engagement, and providing students with more realistic placement expectations. The study highlights several advantages, such as the ability to enhance institutional strategies and aid in better resource allocation. However, it also identifies disadvantages, including the dependency on the quality of input data, and potential issues of bias, fairness, and overfitting, particularly in algorithms like Decision Tree and Random Forest. The results underscore the importance of dataset quality and feature diversity for model performance, and this research serves as a benchmark for institutions to align resources and strategies with placement objectives.

Manike et al., [17] a study titled "Student Placement Chance Prediction Model using Machine Learning Techniques." This study focused on predicting student placement probabilities and categorizing students into specific company types such as Day Sharing, Dream, Super Dream, and Marquee companies. The authors employed XGBoost, a high-speed implementation of gradient-boosted decision trees, and used features such as academic performance (CGPA), technical skills, training experiences, projects, and other key attributes obtained through resume parsing. The model achieved a precision of 80% and an accuracy of 70%, providing actionable placement insights. The purpose of this research was to help students and institutions assess placement chances and identify areas for improvement in their profiles. The study highlights several advantages, such as its ability to categorize students by company types and provide useful placement predictions. However, it also notes disadvantages, including the need for careful data preprocessing, including feature selection and noise handling, to ensure accurate predictions.

Chavhan et al., [18] a study titled "Machine Learning Based Placement Prediction - A Comparative Study." This paper evaluated the performance of three machine learning algorithms: Decision Tree, Naive Bayes, and K-Nearest Neighbors (KNN). The dataset included features such as age, gender, academic stream, CGPA, internships, hostel status, and backlog history. Among the algorithms, Decision Tree emerged as the best performer, achieving an F1-Score of 0.883 and an accuracy of 87.75%, followed by KNN (85.7% accuracy with k=5) and Naive Bayes (81.3% accuracy). The purpose of this research was to compare the effectiveness of different machine learning algorithms in predicting student placements. The study highlights several advantages, such as Decision Tree’s high accuracy and performance metrics. However, it also notes disadvantages, including the importance of dataset characteristics and preprocessing steps, like encoding categorical variables and carefully splitting training/testing data, to achieve optimal results.

Chandra Sekhar et al., [19] a study titled "Data Preprocessing and Visualizations Using Machine Learning for Student Placement Prediction." This study analyzed 1,000 student records over four years from a technical institute, considering attributes such as Gender, SSC%, Inter%, EAMCET Rank, CGPA, and Backlogs. The research emphasized preprocessing techniques like handling null values, normalization, and data visualization using pair plots and box plots. The key insights revealed that EAMCET Rank, CGPA, and Branch were significant predictors of placement, with IT and ECE students achieving higher placement rates. The purpose of this research was to enhance placement predictions by understanding data patterns and relationships. The study highlights several advantages, such as the ability to identify critical features for placement prediction and the use of effective data preprocessing and visualization techniques. However, it also notes some disadvantages, including the need for careful handling of null values and the potential challenges in visualizing large datasets for better clarity.

Wajdi Alghamdi, [20] a study titled "A Comparative Analysis on Machine Learning-Based Student Placement Prediction." This study explored the use of machine learning algorithms, including KNN, SVM, Decision Trees, and Random Forest, to predict student placements. The dataset included academic performance and extracurricular participation. Among the algorithms tested, SVM achieved the highest accuracy of 100%, followed by Logistic Regression with an accuracy of 97.59%. The study focused on preprocessing techniques such as handling missing values and feature selection to optimize model predictions. The purpose of this research was to provide actionable insights for institutions to improve student placement rates and guide skill development. The study highlights several advantages, including the high prediction accuracy of the SVM model and the usefulness of feature selection in enhancing model performance. However, it also notes disadvantages, such as the dependence on high-quality data and the challenge of selecting the most relevant features for placement prediction.

**MOTIVATION:**

Extensive research into this domain has been enabled by rising needs for effective placement prediction models according to various literature surveys. The research uses multiple machine learning methods to evaluate student placement results by studying performance measures alongside technical competencies and school participation alongside many other characteristics. The promising prediction results from these models stem from limited feature usage because they do not consider all relevant aspects that influence student placement success.

The current analysis expands prediction modeling by building from existing findings. Our research team carefully amassed and studied 40 different student characteristics which extend beyond scholarly accomplishment and technical abilities to include physical health indicators and family background information and social aptitude and environmental factors. The chosen attributes originated from multiple dimensions of student life because this selection process ensures that predictions built from the model will be highly accurate and reflect students' comprehensive profiles.

The research includes mote than 40 attributes because it recognizes that placement success results from various connected elements which traditional research frequently neglects. The physical condition of students together with their familial support networks and their socioeconomic status play determining roles in their placement readiness. The evaluation of numerous attributes in placement prediction aims to build both a precise and comprehensive model which extends past standard academic abilities and technical skills for institutions to support students effectively for placement achievement enhancement areas.

The primary purpose of this study is to add to placement prediction technology development through enhanced prediction capabilities alongside novel fundamental success barriers discovery in the student experience. The proposed method will benefit students when coupled with educational institutions as it generates practical recommendations which drive increased preparation and support results in higher placement success rates.

**Chapter 3**

**METHODOLOGY**

**3.1 DATA COLLECTION**

We developed a Google Form to collect immediate information from students about academic measurements as well as technical capabilities and personal attributes. The research survey reached students in every level and major area of our institution to acquire data from diverse participant groups. The questionnaires selected for the placement prediction are given in table 3.1.

**List of questionnaires that are considered**:

**Table 3.1: List of Questionnaires**

|  |  |
| --- | --- |
| **S.No** | **Student Attributes** |
| 2 | Year Of Graduation |
| 3 | Status |
| 4 | Department |
| 5 | Mobile Number |
| 6 | 10th Percentage |
| 7 | 12th Percentage |
| 8 | BTech seat allocation |
| 9 | CGPA in B.Tech |
| 10 | EAMCET Rank |
| 11 | How many certification courses have you completed during you BTech program(Other than NPTEL)? |
| 12 | How many NPTEL courses completed during BTech program |
| 13 | Did you engage in online gaming during your BTech program? |
| 14 | How many hours did you spend each day on an average for online games ? |
| 15 | Did you do any physical activities like sports, running, gym workouts during your BTech program? |
| 16 | How many hours did you spend each day on an average on physical exercises? |
| 17 | How would you rate your knowledge and skills in coding and programming on the scale of 5? |
| 18 | How many hours per day did you dedicate to problem-solving or coding during your BTech program on an average ? |
| 19 | How many projects have you completed during your BTech program on your own? |
| 20 | How many hackathons did you participate in during your BTech program? |
| 21 | How many hackathons did you win during your BTech program ?(any prize) |
| 22 | How many hours per day did you spend throughout your BTech program preparing for campus placements, including aptitude, reasoning, verbal skills, coding, technical on an average? |
| 23 | How many mock interviews did you attend for placements during your BTech program? |
| 24 | Have you been placed in any company? |
| 25 | What package were you offered during your placement, either on-campus or off-campus? |
| 26 | How would you rate your communication skills on the scale of 5? |
| 27 | How many backlogs did you have throughout your entire BTech program? |
| 28 | In how many years after 4 year BTech program did you clear your backlogs? |
| 29 | Did you receive any merit scholarship during BTech program apart from regular government fee reimbursement? |
| 30 | How many hours per day did you spend on social media platforms during your BTech program on an average? |
| 31 | How would you rate your confidence levels during interviews on the scale of 5 ? |
| 32 | Did you receive any stipend from the companies during your final year internship ? |
| 33 | How would you rate yourself in a particular technology (e.g., full stack, front-end, back-end, cloud, AI/ML, app development)? |
| 34 | How many coding competition did you participate during your BTech program ? |
| 35 | What is the percentage of contribution in your final year project ? |
| 36 | Are you confident that your final year project problem statement taken is a real time project and not copied from any online resources?(Please provide genuine answer) |
| 37 | What is your annual family income ? |
| 38 | What is average Attendance percentage did you maintained in your B.Tech program |
| 39 | How many siblings do you have ? |
| 40 | Do you have habit of asking doubts to your faculty |
| 41 | From which area are you from |
| 42 | In which medium did you studied upto 10th standard |
| 43 | What is your father's occupation ? |
| 44 | Do you have habit of asking doubts of your classmates or friends |
| 45 | What were your confidence levels in public speaking on a scale of 5? |

From **table 3.1**, Our research team gathered 842 substantial samples from both our own and our seniors' placement data, enabling accurate prediction of placement outcomes and financial packages through this real-time database.

**Table 3.2: Year-wise Branch Distribution of Students for Placement Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **Year Of Graduation** | **Branch** | **No Of Students** |
| 1. | 2023 | CSE | 10 |
| 2. | 2023 | IT | 32 |
| 3. | 2023 | ECE | 6 |
| 4. | 2023 | EEE | 4 |
| 5. | 2023 | CIVIL | 6 |
| 6. | 2023 | MECH | 4 |
| 7. | 2024 | CSE | 38 |
| 8. | 2024 | IT | 76 |
| 9. | 2024 | AIDS | 12 |
| 10. | 2024 | ECE | 36 |
| 11. | 2024 | CSBS | 62 |
| 12. | 2024 | EEE | 20 |
| 13. | 2024 | CIVIL | 14 |
| 14. | 2024 | MECH | 8 |
| 15. | 2025 | CSE | 6 |
| 16. | 2025 | IT | 208 |
| 17. | 2025 | AIDS | 200 |
| 18. | 2025 | ECE | 36 |
| 19. | 2025 | CSBS | 60 |
| 20. | 2025 | EEE | 20 |
| 21. | 2025 | CIVIL | 14 |
| 22. | 2025 | MECH | 8 |
| 23. | 2025 | Bsc | 2 |
|  |  |  | 842 |

The analysis from **table 3.2** is based on placement statistics from 842 enrolled students who study at different branches and belong to multiple graduating classes from 2023 to 2025. The institution comprises departments that consist of CSE, IT, ECE, EEE, AIDS, CSBS, and additional fields. The largest student population stems from the 2025 group in both IT and AIDS program departments. The data record helps predict placement patterns to make better academic or professional choices.

**3.2 DATA PREPROCESSING**

**3.2.1 HANDLING CATEGORICAL DATA:**

We applied Label Encoding to convert categorical values and multi-category attributes into numerical values since machine learning models cannot directly handle these types of data. Object data types get converted to strings before LabelEncoder from Scikit-learn applies numerical encoding to the data. We performed an encoding operation on all categorical along with multi-category attributes to generate numerical format data which allowed more efficient calculations. Examples of Encoded Attributes: The following features fall under the binary category: Gender, Seat Allocation, Merit Scholarship, and Stipend Received. Multi-Category Features (e.g., Annual Family Income, Average Attendance Percentage in B.Tech, Father’s Occupation, Communication Skills, Coding Skills, Technical Competitions Participation)

# **Python Implementation:**

from sklearn.preprocessing import LabelEncoder

#Identify categorical columns

col = []

for i in data\_copy.columns:

if data\_copy[i].dtypes == 'object':

col.append(i)

# #Initialize Label Encoder

obj = LabelEncoder()

# #Apply Label Encoding to all categorical attributes

for i in col:

data\_copy[i] = data\_copy[i].astype(str) # Convert to strings

data\_copy[i] = obj.fit\_transform(data\_copy[i]) # Apply encoding

**Table 3.3: Sample data of 1-12 columns from collected data**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **2** | **3** | **4** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| 2024 | Alumni | AIDS | 100 | 97.5 | Convenor quota(EAMCET) | 7.99 | 5974 | 0 | 0 |
| 2025 | Current Student | IT | 98 | 97.1 | Convenor quota(EAMCET) | 7.99 | 6848 | 1 | 0 |
| 2025 | Current Student | IT | 98 | 95.9 | Management quota | 9.1 | 22969 | 0 | 2 |
| 2025 | Current Student | AIDS | 98 | 94.6 | Convenor quota(EAMCET) | 7.57 | 10140 | 0 | 2 |
| 2025 | Current Student | AIDS | 100 | 96.9 | Convenor quota(EAMCET) | 9.15 | 12996 | 3 | 1 |
| 2025 | Current Student | IT | 93 | 90.2 | Convenor quota(EAMCET) | 7.32 | 19258 | 3 | 0 |
| 2025 | Current Student | AIDS | 98 | 98 | Convenor quota(EAMCET) | 9.03 | 8236 | 1 | 1 |
| 2025 | Current Student | AIDS | 95 | 96.6 | Convenor quota(EAMCET) | 8.75 | 7893 | 0 | 0 |
| 2025 | Current Student | AIDS | 95 | 98.2 | Convenor quota(EAMCET) | 8.62 | 8893 | 5 | 0 |

**Table 3.4: Sample data of 13-24 columns from collected data**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** |
| No | 0 | Yes | 1 | 4 | 2 | 4 | 1 | 0 | 3 | 7 | off-campus |
| Yes | 4 | Yes | 1 | 4 | 2 | 4 | 2 | 0 | 2 | 0 | Not yet placed |
| No | 1 | No | 0 | 3 | 2 | 3 | 1 | 0 | 0 | 1 | on-campus |
| Yes | 3 | Yes | 2 | 4 | 4 | 3 | 0 | 0 | 3 | 2 | Not yet placed |
| No | 0 | Yes | 1 | 4 | 4 | 4 | 0 | 0 | 2 | 3 | on-campus |
| No | 0 | Yes | 2 | 3 | 2 | 2 | 1 | 0 | 5 | 1 | Not yet placed |
| Yes | 2 | Yes | 1 | 4 | 1 | 3 | 1 | 0 | 3 | more than 10 | on-campus |
| Yes | 2 | Yes | 2 | 3 | 3 | 3 | 2 | 0 | 3 | 4 | on-campus |
| No | 0 | Yes | 1 | 3 | 1 | 0 | 0 | 0 | 2 | 2 | Not yet placed |
| No | 0 | Yes | 1 | 4 | 2 | 4 | 1 | 0 | 3 | 7 | off-campus |

**Table 3.5: Sample data of 25-36 columns from collected data**

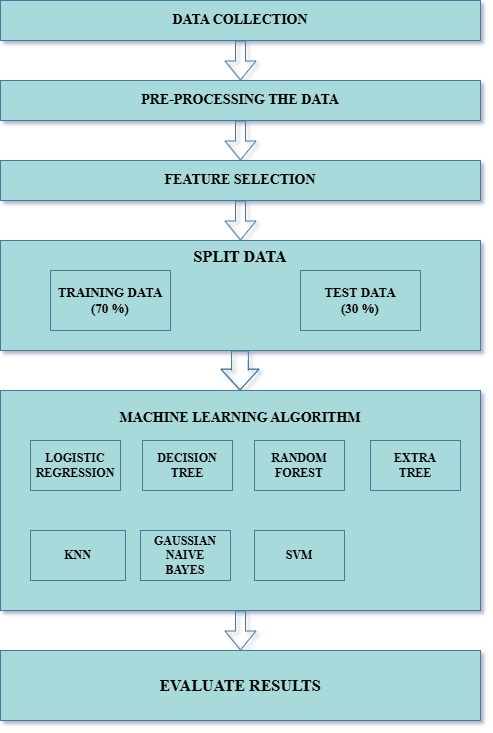
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **25** | **26** | **27** | **28** | **29** | **30** | **31** | **32** | **33** | **34** | **35** |
| 3-5 LPA | 4 | 0 | 0 | No | 4 | 4 | Yes | 3 | 1 | 40-60 |
| Not Placed | 3 | 0 | 0 | No | 0 | 5 | No | 5 | 2 | 0-20 |
| 3-5 LPA | 3 | 0 | 0 | Yes | 3 | 4 | No | 0 | 0 | 40-60 |
| Not Placed | 4 | 0 | 0 | No | 3 | 3 | Yes | 4 | 0 | 20-40 |
| 3-5 LPA | 4 | 0 | 0 | No | 1 | 4 | No | 4 | 3 | 20-40 |
| Not Placed | 4 | 0 | 0 | No | 3 | 4 | No | 4 | 0 | 40-60 |
| 3-5 LPA | 4 | 0 | 0 | No | 8 | 4 | No | 3 | 0 | 80-100 |
| 3-5 LPA | 3 | 0 | 0 | No | 2 | 4 | No | 3 | 2 | 60-80 |
| Not Placed | 4 | 0 | 0 | No | 2 | 3 | No | 3 | 0 | 80-100 |
| 3-5 LPA | 4 | 0 | 0 | No | 4 | 4 | Yes | 3 | 1 | 40-60 |

**Table 3.6: Sample data of 37-45 columns from collected data**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **36** | **37** | **38** | **39** | **40** | **41** | **42** | **43** | **44** | **45** |
| Yes | Less than 2 lakha | 85-100 | 1 | Yes | Rural | English | Private employee | Yes | 4 |
| Yes | 2-5 lakhs | 75-85 | 1 | No | Rural | English | Farmer | Yes | 3 |
| Yes | 2-5 lakhs | 75-85 | 1 | Yes | Urban | English | Government employee | Yes | 3 |
| Yes | Less than 2 lakha | 75-85 | 1 | Yes | Rural | English | Farmer | Yes | 5 |
| Yes | Less than 2 lakha | 85-100 | 1 | Yes | Rural | English | Farmer | Maybe | 4 |
| Yes | 5-10 lakhs | 65-75 | 1 | Yes | Urban | English | Government employee | Yes | 3 |
| Yes | 2-5 lakhs | 85-100 | 1 | Yes | Rural | English | Government employee | Yes | 3 |
| Yes | 2-5 lakhs | 65-75 | 2 | Yes | Urban | English | Government employee | Yes | 2 |
| Yes | 2-5 lakhs | 75-85 | 1 | Yes | Rural | English | Farmer | Yes | 3 |
| Yes | Less than 2 lakha | 85-100 | 1 | Yes | Urban | English | Labour | Yes | 3 |

**Tables 3.3, 3.4, 3.5**, and **3.6** present a sample of 10 records each to illustrate the structure and format in which the data is stored. In these tables, the columns are numbered from 1 to 45 for simplification. **Table 3.1** serves as a reference, mapping each column number to its corresponding attribute name

**PROCESS FLOWCHART:**



**Fig 3.1: Process Flow Diagram**

Fig **3.1** outlines the placement prediction process using machine learning, beginning with data collection, preprocessing, and feature selection. The dataset is split 70/30 into training and testing sets, and various algorithms like Logistic Regression, Decision Tree, Random Forest, KNN, and SVM are applied. The best-performing model is identified through result evaluation.

**3.3 Model Training and Evaluation:**

**3.3.1 Binary Classification for Campus Placement Prediction (Placed/Not Placed):**

The target variable "Have you been placed in any company?" serves as the binary classification problem with possible outcomes Selected (1) or Not Selected (0). The target variable forms a binary classification structure which leads to two possible outcomes between Selected (1) and Not Selected (0). The system integrates all accessible parameters for input data processing.

***Feature Selection for Placement Prediction:***

Except for the target variable, all the remaining features were taken as input for training the model to learn patterns and make accurate predictions.

# ***Python Implementation:***

import pandas as pd

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import StandardScaler

X = data\_copy.drop(columns=['Have you been placed in any company?']) # Replace with your feature columns

y = data\_copy['Have you been placed in any company?'] # Replace with your target column

***Train-Test split :***

Here, data is divided into two parts i.e. training data & testing data. Where 70 % data is taken for training our machine learning algorithm and remaining 30 % data is used for testing whether our trained machine learning model is working correctly or not.

# ***Python Implementation:***

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.3,random\_state=60)

x\_train.shape,x\_test.shape,y\_train.shape,y\_test.shape

To ensure uniform feature scaling, **Min-Max Scaling** is applied to normalize the data

***MinMaxScaler:***

Min-Max scaling was deployed to normalize our features so they matched a uniform scale from 0 through 1 for achieving consistent data comparisons. The normalization technique both maintains original data relation and transforms multiple variables belonging to different scales into proportional values. This normalization method allowed us to prevent numerical variance from skewing the model analysis yet it maintained correct relative weights between all variables in our predictive model.

# ***Python Implementation:***

from sklearn.preprocessing import MinMaxScaler

scal\_obj=MinMaxScaler(feature\_range=(0,1)) x\_train=scal\_obj.fit\_transform(x\_train)

x\_train=pd.DataFrame(x\_train) x\_test=scal\_obj.fit\_transform(x\_test)

x\_test=pd.DataFrame(x\_test)

Subsequently, four different classification algorithms are implemented to evaluate their performance:

1)Logistic Regression

2)Decision Tree Classifier

3)Random Forest Classifier

4)Extra Trees Classifier

# ***Comparison of Classification Models: Performance Evaluation and Metrics***

# ***Analysis****:*

# ***Python Implementation:***

import pandas as pd

import matplotlib.pyplot as plt

from math import sqrt

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import ( confusion\_matrix, classification\_report, roc\_auc\_score, roc\_curve, )

ModelLR = LogisticRegression()

ModelDC = DecisionTreeClassifier()

ModelRF = RandomForestClassifier()

ModelET = ExtraTreesClassifier()

MM = [ModelLR, ModelDC, ModelRF, ModelET]

EMResults = pd.DataFrame( columns=[ "Model Name", "True\_Positive", "False\_Negative", "False\_Positive", "True\_Negative", "Accuracy", "Precision", "Recall", "F1 Score", "Specificity", "MCC", "ROC\_AUC\_Score", "Balanced Accuracy", ] )

for models in MM:

print(f"\n{'-' \* 60}\nEvaluating Model: {models.**class**.**name**}\n{'-' \* 60}")

# Fit the model  
models.fit(x\_train, y\_train)  
  
# Predictions  
y\_pred = models.predict(x\_test)  
y\_pred\_prob = models.predict\_proba(x\_test)[:, 1]

if hasattr(models, "predict\_proba") else None  
  
# Confusion Matrix  
matrix = confusion\_matrix(y\_test, y\_pred, labels=[1, 0])  
tp, fn, fp, tn = matrix.ravel()  
  
# Classification Report  
C\_Report = classification\_report(y\_test, y\_pred, labels=[1, 0])  
print("Confusion Matrix:\n", matrix)  
print("Classification Report:\n", C\_Report)  
  
# Metrics Calculation  
sensitivity = round(tp / (tp + fn), 3) if (tp + fn) != 0 else 0  
specificity = round(tn / (tn + fp), 3) if (tn + fp) != 0 else 0  
accuracy = round((tp + tn) / (tp + fp + tn + fn), 3)  
balanced\_accuracy = round((sensitivity + specificity) / 2, 3)  
precision = round(tp / (tp + fp), 3) if (tp + fp) != 0 else 0  
f1Score = round((2 \* tp) / (2 \* tp + fp + fn), 3) if (2 \* tp + fp + fn) != 0 else 0  
mcc\_denom = (tp + fp) \* (tp + fn) \* (tn + fp) \* (tn + fn)  
MCC = round(((tp \* tn) - (fp \* fn)) / sqrt(mcc\_denom), 3)

if mcc\_denom != 0 else 0  
roc\_auc = round(roc\_auc\_score(y\_test, y\_pred), 3)

if y\_pred\_prob is not None:  
 fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred\_prob)  
 plt.figure()  
 plt.plot(fpr, tpr, label=f"{models.\_\_class\_\_.\_\_name\_\_} (AUC = {roc\_auc})")  
 plt.plot([0, 1], [0, 1], "r--")  
 plt.xlabel("False Positive Rate")  
 plt.ylabel("True Positive Rate")  
 plt.title(f"ROC Curve - {models.\_\_class\_\_.\_\_name\_\_}")  
 plt.legend(loc="lower right")  
 plt.show()  
  
 # Add metrics to results DataFrame

new\_row = d.DataFrame([{  
 “Model Name": models.\_\_class\_\_.\_\_name\_\_,  
 "True\_Positive": tp,  
 "False\_Negative": fn,  
 "False\_Positive": fp,  
 "True\_Negative": tn,  
 "Accuracy": accuracy,  
 "Precision": precision,  
 "Recall": sensitivity,  
 "F1 Score": f1Score,  
 "Specificity": specificity,  
 "MCC": MCC,  
 "ROC\_AUC\_Score": roc\_auc,  
 "Balanced Accuracy": balanced\_accuracy,  
 }])

EMResults = pd.concat([EMResults, new\_row], ignore\_index=True)

***Preliminary Results:***

**Table 3.7: Classification output on placement prediction without feature selection**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model**  **Name** | **True**  **Positive** | **False Negative** | **False**  **Positive** | **True Negative** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **Specificity** |
| Logistic  Regresssion | 106 | 28 | 27 | 91 | 0.78 | 0.79 | 0.79 | 0.79 | 0.77 |
| Decision Tree  Classifier | 116 | 18 | 5 | 113 | 0.90 | 0.95 | 0.86 | 0.91 | 0.92 |
| Random Forest  Classifier | 122 | 12 | 2 | 116 | 0.94 | 0.98 | 0.91 | 0.94 | 0.98 |
| ExtraTrees  Classifier | 128 | 6 | 4 | 114 | 0.96 | 0.97 | 0.95 | 0.96 | 0.96 |

The ExtraTrees Classifier demonstrated the most effective results in **Table 3.7** with 96% accuracy together with high precision values at 0.97 and recall value at 0.95. This indicates impressive performance in identifying placed students without producing false outcomes. The Random Forest algorithm demonstrated comparable performance to ExtraTrees Classifier while maintaining the best specificity rate of 0.98 which makes it an excellent method to identify students who are not placed. Decision Tree achieved an accuracy level of 90% yet Logistic Regression performed below with 78% accuracy together with reduced specificity.

### **3.3.2 Multiclass Classification for Package Prediction:**

In this approach, the target variable "What package were you offered during your placement, either on-campus or off-campus?" is treated as a multiclass classification problem. The target variable tracks different placement package options between on-campus and off-campus establishment locations. To classify salary packages effectively the system requires a multiclass model that groups salaries into these four groups: Not Applicable and 3-5 LPA, 5-10 LPA, and 10+ LPA. Every available parameter functions as an input component which allows the system to discover patterns for predicting results accurately.

***Feature Selection for Package Prediction:***

* Except for the target variable, all the remaining features were taken as input for training the model to learn patterns and make accurate predictions.

# ***Python Implementation:***

target\_var = "What package were you offered during your placement, either on-campus or off-campus?"

x = data\_copy.drop(columns=[target\_var]) # Drop the target variable to get all other features y = data\_copy[target\_var] # Target variable

print("Independent Variables (Features):", x.columns.tolist())

print("Target Variable:", target\_var)

The dataset is first split into training and testing sets using the **train-test split** method.

***Train-Test split:***

Here, data is divided into two parts i.e. training data & testing data. Where 70 % data is taken for training our machine learning algorithm and remaining 30 % data is used for testing whether our trained machine learning model is working correctly or not.

# ***Python Implementation:***

from sklearn.model\_selection import train\_test\_split x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.3,random\_state=60) x\_train.shape,x\_test.shape,y\_train.shape,y\_test.shape

To maintain consistency in feature scaling, Min-Max Scaling is applied to normalize the data

# ***MinMaxScaler:***

Min-Max scaling was deployed to normalize our features so they matched a uniform scale from 0 through 1 for achieving consistent data comparisons. The normalization technique both maintains original data relation and transforms multiple variables belonging to different scales into proportional values. This normalization method allowed us to prevent numerical variance from skewing the model analysis yet it maintained correct relative weights between all variables in our predictive model.

# ***Python Implementation:***

from sklearn.preprocessing import MinMaxScaler

scal\_obj=MinMaxScaler(feature\_range=(0,1))

x\_train=scal\_obj.fit\_transform(x\_train)

x\_train=pd.DataFrame(x\_train)

x\_test=scal\_obj.fit\_transform(x\_test)

x\_test=pd.DataFrame(x\_test)

**Four different classification algorithms are then implemented to evaluate their performance:**

1)Logistic Regression

2)Decision Tree Classifier

3)Random Forest Classifier

4)Extra Trees Classifier

# ***Comparison of Classification Models: Performance Evaluation and Metrics Analysis:***

# ***Python Implementation:***

import pandas as pd

import matplotlib.pyplot as plt

from math import sqrt

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import ( confusion\_matrix, classification\_report, roc\_auc\_score, roc\_curve, accuracy\_score, precision\_score, recall\_score, f1\_score, balanced\_accuracy\_score, matthews\_corrcoef, )

ModelLR = LogisticRegression(multi\_class='ovr', solver='liblinear')

ModelDC = DecisionTreeClassifier()

ModelRF = RandomForestClassifier()

ModelET = ExtraTreesClassifier()

models\_list = [ModelLR, ModelDC, ModelRF, ModelET]

EMResults = pd.DataFrame(columns=[ "Model Name", "Accuracy", "Precision", "Recall", "F1 Score", "Specificity", "Balanced Accuracy", "MCC", "ROC AUC Score" ])

for model in models\_list:

print(f"\n{'-' \* 60}\nEvaluating Model: {model.**class**.**name**}\n{'-' \* 60}")

# Fit the model  
model.fit(x\_train, y\_train)  
  
# Predictions  
y\_pred = model.predict(x\_test)  
y\_pred\_prob = model.predict\_proba(x\_test) if hasattr(model, "predict\_proba") else None  
  
# Confusion Matrix  
matrix = confusion\_matrix(y\_test, y\_pred, labels=list(set(y\_test)))  
print("Confusion Matrix:\n", matrix)  
  
# Metrics  
accuracy = accuracy\_score(y\_test, y\_pred)  
precision = precision\_score(y\_test, y\_pred, average='weighted', zero\_division=0)  
recall = recall\_score(y\_test, y\_pred, average='weighted')  
f1 = f1\_score(y\_test, y\_pred, average='weighted')  
balanced\_accuracy = balanced\_accuracy\_score(y\_test, y\_pred)  
specificity = sum(matrix[i, i] / matrix[:, i].sum() for i in range(matrix.shape[0]) if matrix[:, i].sum() != 0) / len(set(y\_test))  
mcc = matthews\_corrcoef(y\_test, y\_pred)  
  
# Adjust ROC AUC Score for multi-class classification  
if y\_pred\_prob is not None and len(set(y\_test)) > 2:  
 roc\_auc = roc\_auc\_score(y\_test, y\_pred\_prob, multi\_class='ovr')  
elif y\_pred\_prob is not None:  
 roc\_auc = roc\_auc\_score(y\_test, y\_pred\_prob[:, 1])  
else:  
 roc\_auc = None  
# Print Metrics  
print(f"Accuracy: {accuracy \* 100:.2f}%")  
print(f"Precision: {precision \* 100:.2f}%")  
print(f"Recall: {recall \* 100:.2f}%")  
print(f"F1 Score: {f1}")  
print(f"Specificity: {specificity \* 100:.2f}%")  
print(f"Balanced Accuracy: {balanced\_accuracy \* 100:.2f}%")  
print(f"MCC: {mcc}")  
print(f"ROC AUC Score: {roc\_auc}")  
# ROC Curve

if y\_pred\_prob is not None:  
 for i in range(y\_pred\_prob.shape[1]):  
 fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred\_prob[:, i], pos\_label=i)  
 plt.figure()  
 plt.plot(fpr, tpr, label=f"Class {i} (AUC = {roc\_auc:.2f})")  
 plt.plot([0, 1], [0, 1], "r--")  
 plt.xlabel("False Positive Rate")  
 plt.ylabel("True Positive Rate")  
 plt.title(f"ROC Curve - {model.\_\_class\_\_.\_\_name\_\_}")  
 plt.legend(loc="lower right")  
 plt.show()  
  
# Add metrics to results DataFrame  
new\_row = pd.DataFrame([{  
 "Model Name": model.\_\_class\_\_.\_\_name\_\_,  
 "Accuracy": accuracy,  
 "Precision": precision,  
 "Recall": recall,  
 "F1 Score": f1,  
 "Specificity": specificity,  
 "Balanced Accuracy": balanced\_accuracy,  
 "MCC": mcc,  
 "ROC AUC Score": roc\_auc,  
}])  
EMResults = pd.concat([EMResults, new\_row], ignore\_index=True)  
 print("\nConsolidated Results:\n", EMResults)

***Preliminary Results:***

**Table 3.8: Classification result on package prediction without feature selection**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model Name** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **Specificity** | **Balanced Accuracy** | **MCC** | **ROC AUC Score** |
| Logistic Regression | 0.67 | 0.65 | 0.67 | 0.65 | 0.50 | 0.45 | 0.45 | 0.82 |
| Decision Tree Classifier | 0.83 | 0.83 | 0.83 | 0.83 | 0.77 | 0.79 | 0.74 | 0.86 |
| Random Forest  Classifier | 0.92 | 0.93 | 0.92 | 0.92 | 0.94 | 0.84 | 0.88 | 0.98 |
| Extra Trees Classifier | 0.92 | 0.92 | 0.92 | 0.92 | 0.95 | 0.84 | 0.88 | 0.98 |

Random Forest along with Extra Trees Classifiers produced maximum accuracy and ROC AUC Score of 0.98 for predicting student packages according to **Table 3.8**. These models achieved an MCC score of 0.88 due to their reliable predictive power under imbalanced conditions. The accuracy rate of Decision Tree reached 83% yet logistic Regression performed the worst with 67% accuracy accompanied by a minimal MCC score of 0.45 thus demonstrating low reliability levels.

**CONFUSION MATRIX:**

1. **Logistic Regression:**

**Table 3.9: Confusion Matrix For Logistic Regression**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Not Placed | 3-5lpa | 5-10lpa | 10+lpa |
| Not Placed | 0 | 2 | 3 | 0 |
| 3-5lpa | 0 | 69 | 2 | 30 |
| 5-10lpa | 0 | 20 | 11 | 3 |
| 10+ lpa | 0 | 23 | 0 | 89 |

From **table 3.9** : The confusion matrix demonstrates that the Logistic Regression model operates effectively for 10+ LPA and 3-5 LPA predictions since it achieves 89 and 69 correct classifications respectively. The model demonstrates high erroneous rates between sequential salary ranges where it bundles 3-5 LPA with 10+ LPA at thirty times while mixing 5-10 LPA together with 3-5 LPA twenty times. No correct predictions were made for classifications that fell under Not Applicable. The proposed model performs poorly when identifying similar features available within the adjacent salary brackets.

1. **Decision Tree Classifier:**

**Table 3.10: Confusion Matrix For Decision Tree Classifier**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Not Placed | 3-5lpa | 5-10lpa | 10+lpa |
| Not Placed | 3 | 0 | 2 | 0 |
| 3-5lpa | 2 | 83 | 4 | 12 |
| 5-10lpa | 0 | 2 | 32 | 0 |
| 10+ lpa | 0 | 17 | 2 | 93 |

From **table 3.10**: The Decision Tree model achieves high predictive accuracy when identifying the target categories of 3–5 LPA and 10+ LPA securing 83 and 93 correct predictions. A significant portion of the 10+ LPA category gets misidentified as 3–5 LPA by the model, equaling up to 17 instances of the incorrect classification. The predictions for 5–10 LPA were compatible with the expected results in 32 out of 34 instances but the model shows some uncertainty when classifying adjacent salary groups. The model delivers good results yet its ability to differentiate salaries within close pay ranges needs improvement.

1. **Random Forest:**

**Table 3.11: Confusion Matrix For Random Forest**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Not Placed | 3-5lpa | 5-10lpa | 10+lpa |
| Not Placed | 3 | 0 | 2 | 0 |
| 3-5lpa | 0 | 91 | 0 | 10 |
| 5-10lpa | 0 | 2 | 30 | 2 |
| 10+ lpa | 0 | 2 | 0 | 110 |

From **table 3.11**: The Random Forest model showed remarkable accuracy in its classification tasks specifically among 10+ LPA and 3–5 LPA groups resulting in 110 correct predictions for the first and 91 correct predictions for the second category. The model displays low misclassification rates when it comes to salary bands because confusion exists only within the 2 instances of 5–10 LPA and 10 mistakes between 3–5 LPA and 10+ LPA. Overall the model provides excellent discrimination power between salary categories while showing proportional accuracy across the entire range of salary bands.

1. **Extra Trees Classifier:**

**Table 3.12: Confusion Matrix For Extra Trees Classifier**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Not Placed | 3-5lpa | 5-10lpa | 10+lpa |
| Not Placed | 3 | 0 | 2 | 0 |
| 3-5lpa | 0 | 93 | 0 | 8 |
| 5-10lpa | 0 | 2 | 30 | 2 |
| 10+ lpa | 0 | 4 | 0 | 108 |

From **table 3.12**: The Extra Trees Classifier delivers excellent predictive results that generate 93 correct predictions for the 3–5 LPA category and 108 for the 10+ LPA category. Among identified misclassifications there are primarily four occurrences of 10+ LPA assignments being categorized as 3–5 LPA while a few 5–10 LPA cases experienced miscounting. The model demonstrates effective salary bracket differentiation that proves its reliability and accuracy in predicting packages.

**3.4 Impact of Feature Selection on Binary and Multiclass Classification Performance:**

**3.4.1 Enhancing Placement Prediction for Binary Classification of Placed or Not with Feature Selection Techniques:**

Different feature selection methods were used to select the most influential features that would improve our binary classification model's performance in predicting placement status. The following methods were used:

1. ***Chi-Square Test (Filter Method):***

We did not arrive at our findings by guessing because our research depended heavily on the Chi-square (χ²) statistical test. Chi-square works as an investigative tool that reveals genuine statistical connections between placement success factors above accidental correlations. The combination of SelectKBest from scikit-learn with the Chi-square test led to scientific verification of the five main influencing factors.

# ***Python Implementation:***

from sklearn.feature\_selection import SelectKBest, chi2

import pandas as pd

X = data\_copy.drop(columns=['Have you been placed in any company?'])

y = data\_copy['Have you been placed in any company?']

selector = SelectKBest(chi2, k=10)

X\_new = selector.fit\_transform(X, y)

selected\_features = X.columns[selector.get\_support()]

print(f"Selected Features: {selected\_features}")

***Features Selected for Chi-Square Test:***

1. 12th Percentage?
2. CGPA in B.Tech?
3. EAMCET Rank?
4. How many NPTEL courses completed during BTech program?
5. How many projects have you completed during your BTech program on your own?
6. How many hackathons did you participate in during your BTech program?
7. How many hackathons did you win during your BTech program ?
8. How many mock interviews did you attend for placements during your BTech program?
9. How would you rate your confidence levels during interviews on the scale of 5 ?
10. How many coading competition did you participate during your BTech program ?

***2) Recursive Feature Elimination (Wrapper Method):***

The analysis used Logistic Regression and Recursive Feature Elimination (RFE) to identify the five leading factors that affect student placement results. The method helped to demonstrate how technical along with non-technical traits substantially affect the outcomes of student placement. The chosen variables illuminate different aspects that affect employability complexity.

# ***Python Implementation:***

from sklearn.feature\_selection import RFE

from sklearn.linear\_model import LogisticRegression

import pandas as pd

X = data\_copy.drop(columns=['Have you been placed in any company?'])

y = data\_copy['Have you been placed in any company?']

model = LogisticRegression()

selector = RFE(model, n\_features\_to\_select=10)

X\_new = selector.fit\_transform(X, y)

selected\_features = X.columns[selector.get\_support()]

print(f"Selected Features: {selected\_features}")

***Features Selected for Recursive Feature Elimination:***

1. How many NPTEL courses completed during BTech program?
2. How would you rate your knowledge and skills in coding and programming on the scale of 5?
3. How many hackathons did you participate in during your BTech program?
4. Did you receive any merit scholarship during BTech program apart from regular government fee reimbursement?
5. How would you rate your confidence levels during interviews on the scale of 5 ?
6. Did you receive any stipend from the companies during your final year internship ?
7. Are you confident that your final year project problem statement taken is a real time project and not copied from any online resources?
8. What is average Attendance percentage did you maintained in your B.Tech program?
9. Do you have habit of asking doubts to your faculty?
10. In which medium did you studied upto 10th standard?

***3) Logistic Regression with L1 Regularization (Embedded Method):***

The analysis based on Logistic Regression using L1 (Lasso) regularization identified twelve key variables that determine student placement results. The built-in feature selection mechanism uses L1 regularization to select a broad combination of characteristics which cover academic performance and programming competency and personal attributes.

***Python Implementation:***

from sklearn.linear\_model import LogisticRegression

from sklearn.feature\_selection import SelectFromModel

import pandas as pd

X = data\_copy.drop(columns=['Have you been placed in any company?'])

y = data\_copy['Have you been placed in any company?']

model = LogisticRegression(penalty='l1', solver='liblinear')

model.fit(X, y)

selector = SelectFromModel(model, threshold="mean")

X\_new = selector.transform(X)

selected\_features = X.columns[selector.get\_support()]

print(f"Selected Features: {selected\_features}")

***Features Selected for Logistic Regression with L1 Regularization:***

1. Did you engage in online gaming during your BTech program?
2. How would you rate your knowledge and skills in coding and programming on the scale of 5?
3. How many hours per day did you dedicate to problem-solving or coding during your BTech program on an average ?
4. How many hackathons did you participate in during your BTech program?
5. How many backlogs did you have throughout your entire BTech program?
6. How would you rate your confidence levels during interviews on the scale of 5 ?
7. Did you receive any stipend from the companies during your final year internship ?
8. Are you confident that your final year project problem statement taken is a real time project and not copied from any online resources?
9. How many siblings do you have ?
10. Do you have habit of asking doubts to your faculty?
11. 'In which medium did you studied upto 10th standard

***4) Random Forest Classifier (Embedded Method):***

The Embedded Method performs model training at the same time as feature selection through one unified process. The Random Forest Classifier operates as a representative Embedded Method that determines feature importance through the degree to which features reduce Gini or entropy impurity in its decision trees. Predictions depend heavily on features that receive higher importance scores in evaluations. This approach helps to find the most useful attributes without causing performance degradation.

# ***Python Implementation:***

from sklearn.ensemble import RandomForestClassifier

import pandas as pd

X = data.drop(columns=['Have you been placed in any company?'])

y = data['Have you been placed in any company?']

model = RandomForestClassifier()

model.fit(X, y)

importances = model.feature\_importances\_

feature\_importances = pd.DataFrame({'Feature': X.columns, 'Importance': importances})

feature\_importances = feature\_importances.sort\_values(by='Importance', ascending=False)

print(f"Feature Importances:\n{feature\_importances}")

***Feature Selected for Random Forest Classifier:***

1. CGPA in B.Tech?
2. 12th Percentage?
3. How many hackathons did you win during your BTech program ?
4. EAMCET Rank?
5. How would you rate your confidence levels during interviews on the scale of 5 ?
6. 10th Percentage?
7. How would you rate your confidence levels during interviews on the scale of 5 ?
8. How would you rate your knowledge and skills in coding and programming on the scale of 5?
9. How would you rate yourself in a particular technology (e.g., full stack, front-end, back-end, cloud, AI/ML, app development)?
10. How many NPTEL courses completed during BTech program?

The evaluation showed that Logistic Regression with L1 Regularization obtained 96.8% accuracy which outperformed the other methods including Recursive Feature Elimination (95.8%) and Random Forest Classifier (95.2%) as well as Chi-Square (94.4%). L1 regularization delivered maximum performance in feature selection which allowed its selected features to become the basis for the final model. L1 Regularization effectively extracted behavioral together with academic traits so the model demonstrated better generalization capability and enhanced interpretability.

***Selected Features from Logistic Regression with L1 Regularization for Placement Prediction :***

1. Did you engage in online gaming during your BTech program?
2. How would you rate your knowledge and skills in coding and programming on the scale of 5?
3. How many hours per day did you dedicate to problem-solving or coding during your BTech program on an average ?
4. How many hackathons did you participate in during your BTech program?
5. How many backlogs did you have throughout your entire BTech program?
6. How would you rate your confidence levels during interviews on the scale of 5 ?
7. Did you receive any stipend from the companies during your final year internship ?
8. Are you confident that your final year project problem statement taken is a real time project and not copied from any online resources?
9. How many siblings do you have ?
10. Do you have habit of asking doubts to your faculty?
11. In which medium did you studied upto 10th standard?

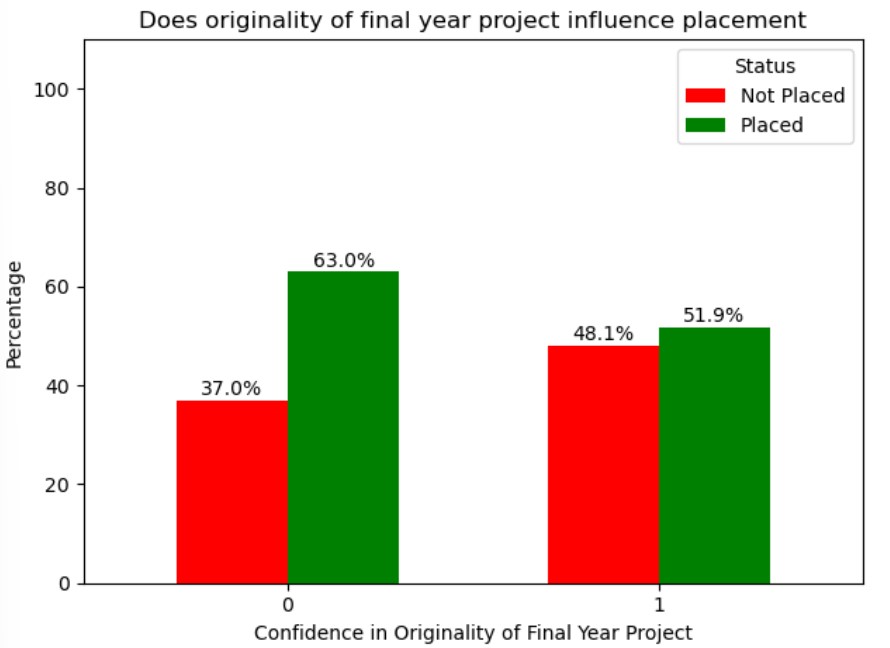
**Table 3.13 : Feature selection results table**

|  |  |  |
| --- | --- | --- |
| **Feature selection method** | **Number of features selected (Total number of features: ?)** | **Accuracy after feature selection (With the best classifier)** |
| 1)Chi-Square Test | 10 | Random Forest-94.4% |
| 2)Recursive Feature Elimination | 10 | Random Forest-95.2% |
| 3)Logistic Regression with L1 Regularization | 11 | ExtraTreesClassifier-96.8% |
| 4)Random Forest Classifier | 10 | Random Forest-95.2% |

Different selection methods yield results which appear in this **table 3.13**. A Chi-Square Test chooses 10 features which enables Random Forest to achieve 94.4% accuracy. When using Recursive Feature Elimination and Random Forest the algorithm picks 10 features that result in a 95.2% accuracy score. By using L1 Regularization on Logistic Regression the model picks 11 features which enables ExtraTreesClassifier to reach 96.8% accuracy. Random Forest identifies 10 features to reach 95.2% accuracy. The various selection methods of features changes the performance characteristics of classifier models.

***Visualizing Feature Importance for Placement Prediction:***

**1) Does originality of final year project influence placement?**



**Fig 3.2: Placement Percentage by originality of final year project**

The provided **fig 3.2** visualizes the effects of student perceptions about their original final projects upon placement success rates. The horizontal axis measures confidence in originality from low (0) to high (1) and the data axis displays percentage of students who received placement or not.

**Key Observations**:

* Students lacking confidence about their project originality finished their placements but students expressing confidence finished at 63 and 37 percentages respectively.
* Evaluators who believed their project was original secured placement for 51.9% of their members and 48.1% did not get placement.

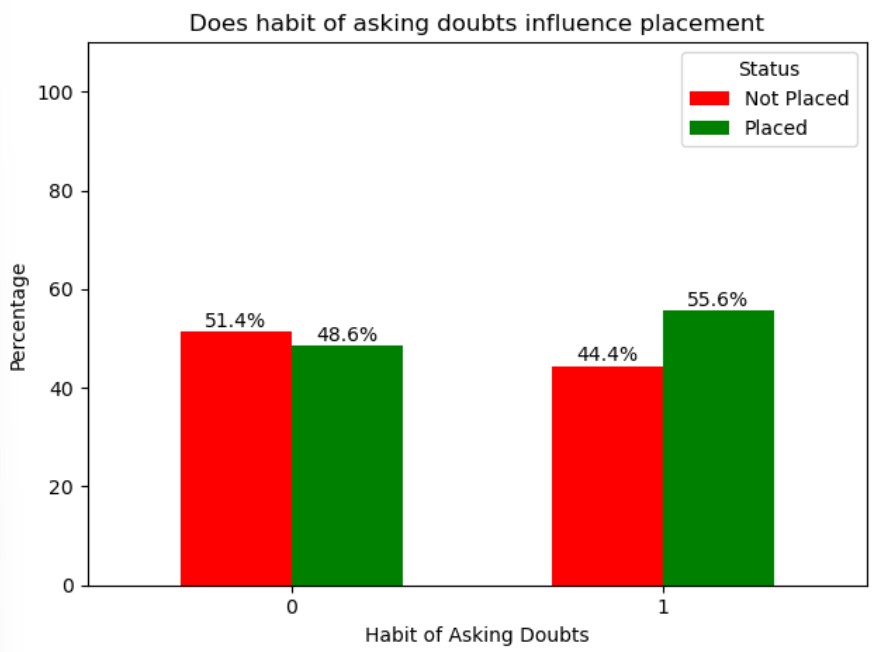
**Key Insights:**

* The data shows that students who did not express trust in their original project submissions performed well for placements although those who demonstrated belief in their project's originality achieved only a slight better placement success rate. Weak positive statistical association exists between these two elements.

**Research Implications:**

* The data indicates that last-year project originality coupled with confidence leads to modest placement success but has no significant impact on placement outcomes. Justifying authentic project work through self-initiative can slightly increase employability outcomes although students must possess additional strong academic capabilities and soft competencies.

**2) Does habit of asking doubts influence placements?**

****

**Fig 3.3: Placement Percentage by habit of asking doubts**

The **fig 3.3** illustrates the relationship between students who asked doubts from their classmates or friends in relation to their placement achievements. The x-axis uses a scale of 1 for students who had the doubt-asking habit while 0 indicates students who did not have this habit and the percentage of placed and non-placed students appears on the y-axis.

**Key Observations:**

* Students who did not ask doubts received placements in 48.6% of cases but 51.4% of them did not secure placement.
* This acts as the baseline. Students who asked their classmates for help regarding doubts reached placement at 55.6% while 44.4% of them did not obtain placement.

**Key Insights:**

* The practice of doubting helps students achieve a better job placement outcome. Placements increased moderately according to student survey results about students who asked specific questions.

**Research Implications:**

* The evidence indicates that students who pose questions for clarification demonstrate better outcomes in their placement exams. Thoughtful academic communication sustains good employability but remains secondary to other factors in the placement success outcome.

**3) Does the medium of study(Up to 10th) influence placement?**

**A graph of a number of people

AI-generated content may be incorrect.**

**Fig 3.4: Placement percentage by medium of study**

A graphic analysis evaluates how students performing during placements depends on their 10th-grade instruction medium (English or non-English). The **fig 3.4** depicts the placement and non-placement statistics against the medium of instruction where 0 designates non-English and 1 stands for English.

**Key Observations:**

* Out of all students who learned in a non-English medium during schooling 46.4% did not obtain placement opportunities while the remaining 53.6% achieved placement success.
* The baseline numbers come from 53.6% placement success and 46.4% of students who were not placed.
* The English medium students obtained the best placement outcome as 66.7% found employment while 33.3% remained unemployed.

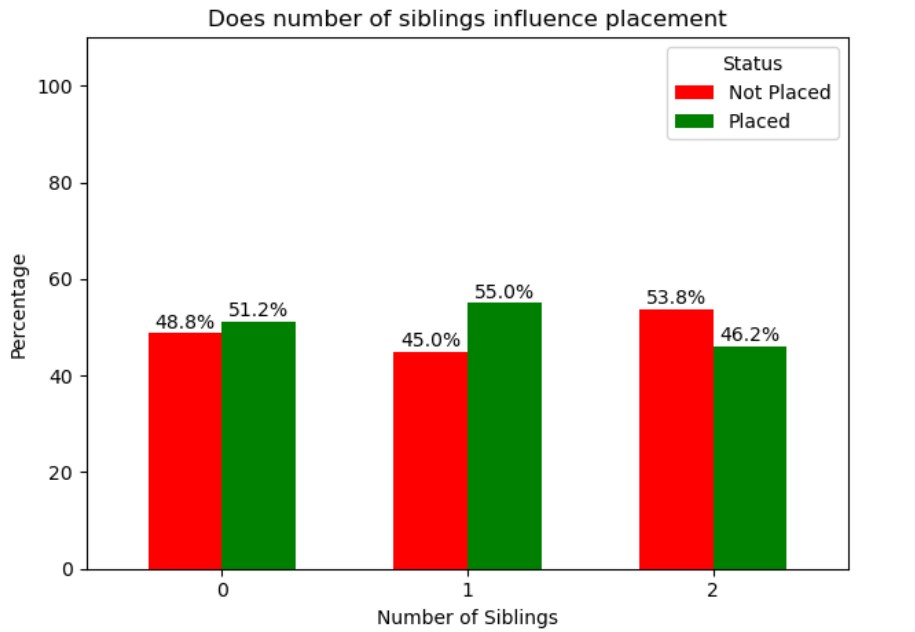
**Key Insights:**

* The results demonstrate that high school education through English as a medium leads student to achieve better placement results. Students undergo better career placement results when they start studying through English Language at an early stage.

**Research Implications:**

* The investigation demonstrates that basic language used for educational instruction strongly affects employment opportunities. Students who receive English as their main instruction medium earn a better chance in successful placements because they attain superior communication and understanding abilities in early education.

**4) Does number of siblings influence placement?**



**Fig 3.5: Placement percentage by number of siblings**

**Fig 3.5** demonstrates that student placement status relates to the number of siblings. The placement and non-placement percentages appear on the left y-axis of the chart which displays sibling count (0, 1, 2) on the x-axis.

**Key Observations:**

* A total of 51.2% students who did not have any siblings got placed while 48.8% remained unplaced. This sets the reference level.
* The placement rate reached 55.0% among students with one sibling but their non-placement percentage amounted to 45.0%.
* Students with two siblings experienced decreased placement success (46.2%) though 53.8% of them could not obtain a placement.

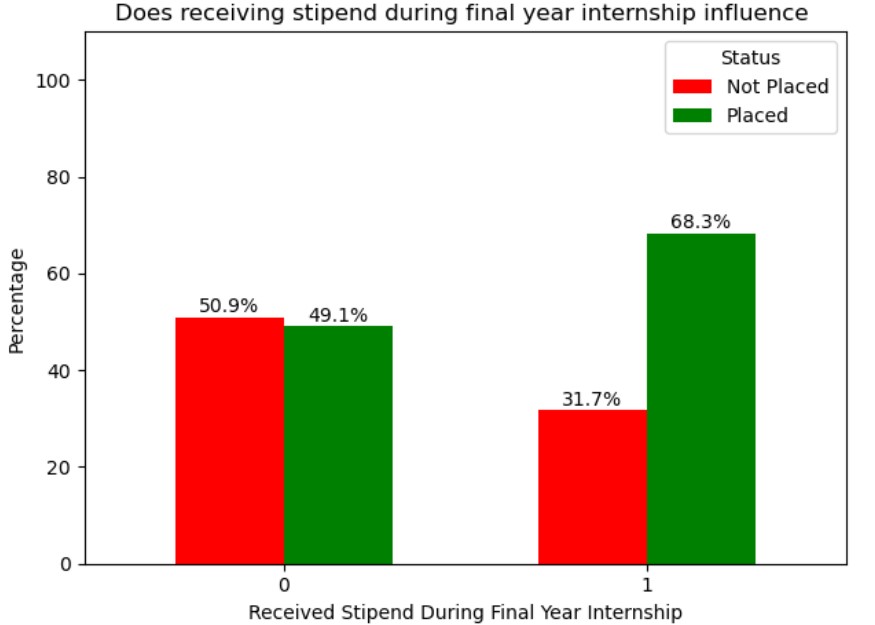
**Key Insights:**

* The graphical data reveals minor alterations between the placement patterns of students with different sibling compositions although it does not indicate any distinct rising or declining tendencies. The research findings demonstrate that sibling number does not create a clear relationship to placement results.

**Research Implications:**

* The research data demonstrates minimal association between the number of siblings and students' job placement possibilities despite minor statistical variances. The results indicate that placement achievements are mainly determined by additional personal and academic variables.

**5) Does receiving stipend during final year internship influence?**



**Fig 3.6: Placement percentage by receiving stipend during final year internship**

**Fig 3.6** depicts placement status changes when interns get stipends during their final year of work. The plot utilizes stipend receipt as the x-variable with values ranging from 0 for no stipend allocation to 1 for students who received financial support. Y-Variable measures percentage distribution between placement success and failure rates.

**Key Observations:**

* A stipend was not distributed to students so their placement success rate reached 49.1% while their unplaced numbers reached 50.9%.
* This represents the baseline data. Research findings exposed that students who received financial support during their final internship achieved 68.3% placement success.
* Slightly more than one-third of stipend-receiving students (31.7%) could not find employment placement during their internship.

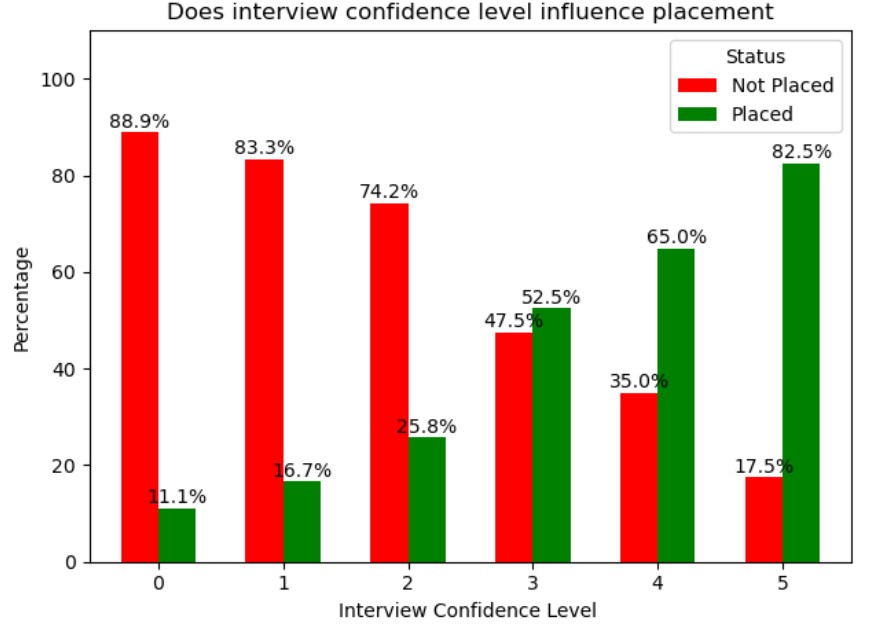
**Key Insights:**

* The research shows that internships with stipends play a crucial role in enhancing student placement results. Students who receive stipends demonstrate higher placement opportunities.

**Research Implications:**

* The research outcome indicates that paid internships prove vital for improving student professional readiness. Students who receive compensation during their internships demonstrate stronger placement performance according to research findings.

**6) Does interview confidence levels influence placement?**

****

**Fig 3.7: Placement percentage by interview confidence level**

Different levels of interview confidence demonstrate their effect on student placement distribution through the graphic chart. **Fig 3.7** demonstrates confidence level measurements on the x-axis ranging from 0 to 5 together with placement and non-placement percentage data shown on the y-axis.

**Key Observations:**

* Among all participants with zero confidence the placement percentage was only 11.1% indicating 88.9% of students failed to secure placement.
* At the third confidence level more students find placement opportunities (52.5%) than they do not (47.5%).
* Out of the students with the highest self-confidence levels at five, 82.5% managed to get placed.

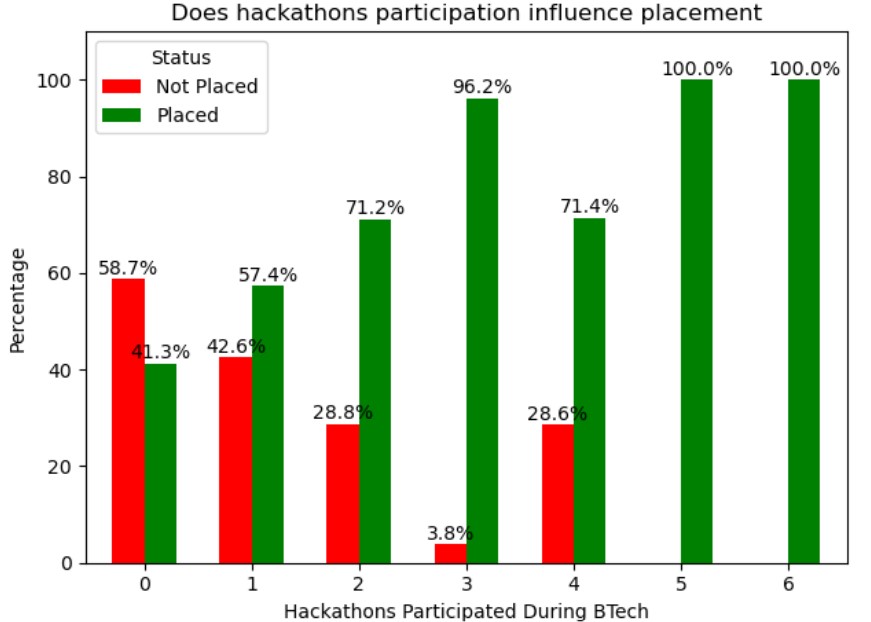
**Key Insights:**

* The data shows placement percentages increasing consistently as confidence levels in the participants increase. Students who demonstrate higher confidence exist in a better placement position.

**Research Implications:**

* Candidacy outcomes strongly depend on candidates developing the ability to confidently handle job interviews. Actual mock interviews combined with training programs focused on confidence building produce successful results as interventions.

**7) Does hackathons participation influence placement?**

****

**Fig 3.8: Placement percentage by hackathons participated during btech**

**Fig 3.8** demonstrates that students attending hackathon events succeed in their placement process. The horizontal bar axis reflects hackathon participation along with placement percentage shown on the vertical axis scale.

**Key Observations:**

* The placement statistics for people who did not participate in Hackathons measured at 41.3%. This established the baseline figures.
* Students who participate in three hackathon events find placements in 96.2% of cases.
* Five or More Hackathons: Achieves a 100% placement rate

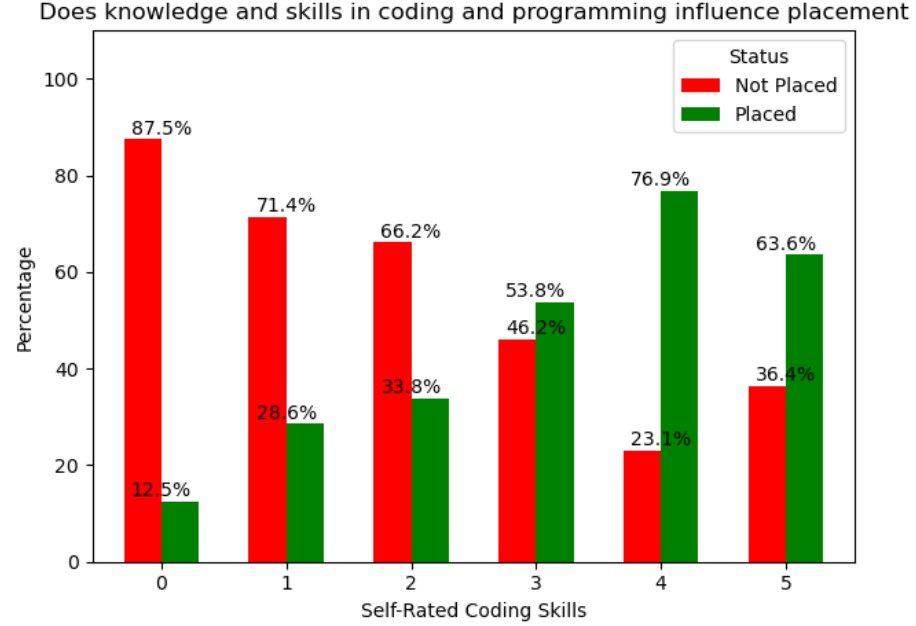
**Key Insights:**

* Results demonstrate that placement opportunities stand strongly correlated to higher hackathon participation rates based on this study's findings. The presented data demonstrates that student placement percentages steadily increase consistently as more hackathons are completed by students.

**Research Implications:**

* Students who enter additional hackathons gain improved placement prospects according to research evidence that confirms hackathon involvement as a vital factor for enhancing employability of university students.

**8) Does knowledge and skills in coding and programming influence placement?**

****

**Fig 3.9: Placement percentage by knowledge and skills in coding and programming**

**Fig 3.9** illustrates how different levels of students' self-evaluated coding ability affect placement statistics. Placement statistics are displayed on the y-axis next to coding skill level categories presented on the x-axis.

**Key Observations:**

* Students who selected zero as their coding skill level managed to get placed at a rate of 12.5%. This established the baseline figures.
* Among students who selected level 4 on the coding skill assessment the placement achievement rate was 76.9%.
* Highest Skill Level (5): Achieves a strong placement rate of 63.6%.

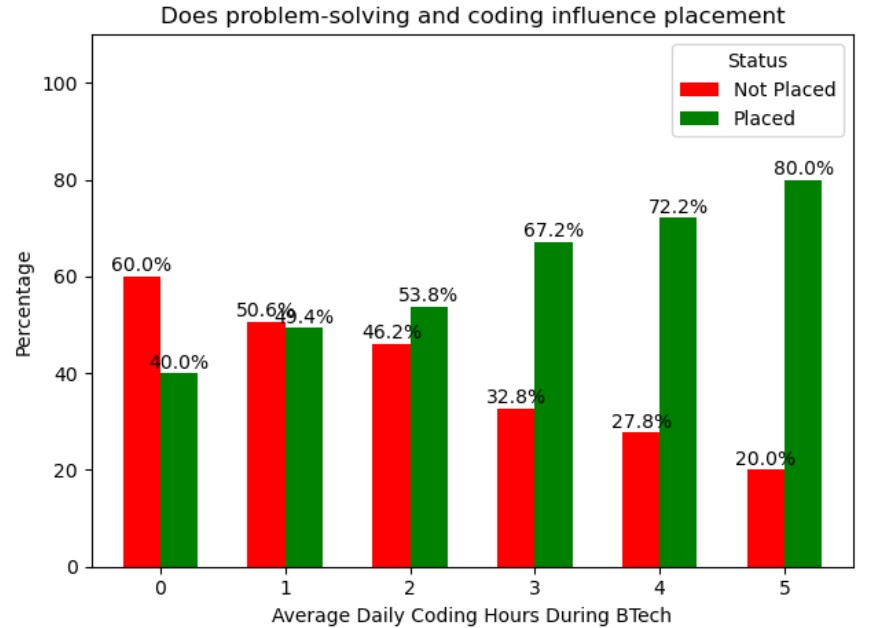
**Key Insights:**

* Attendees with superior evaluations about their coding abilities demonstrate greater placement opportunities. Self-rated coding competence growth leads to parallel improvements in placement opportunities throughout the presented dataset.

**Research Implications:**

* Higher implementation and programming skills development emerges as a critical factor in developing student employability because students who develop competence experience noticeable benefits in their skills.

**9) Does problem-solving and coding influence placement?**

****

**Fig 3.10: Placement percentage by problem-solving and coding during BTech**

From **fig 3.10** Mean daily coding hours from BTech students demonstrate patterns that help reveal their status for placement opportunities in the workforce. The placement percentage for placed and not placed students is displayed on the y-axis together with daily coding hours shown on the x-axis.

**Key Observations:**

* Placed students whose coding time amounts to zero have a placement rate as low as 40% which functions as the minimum level.
* Students who dedicate their time to coding for three hours per day increase their chances of placement to 67.2% which suggests effective progress.
* Fields of study in which students code for 5 hours each day produce the best placement results at 80 percent.

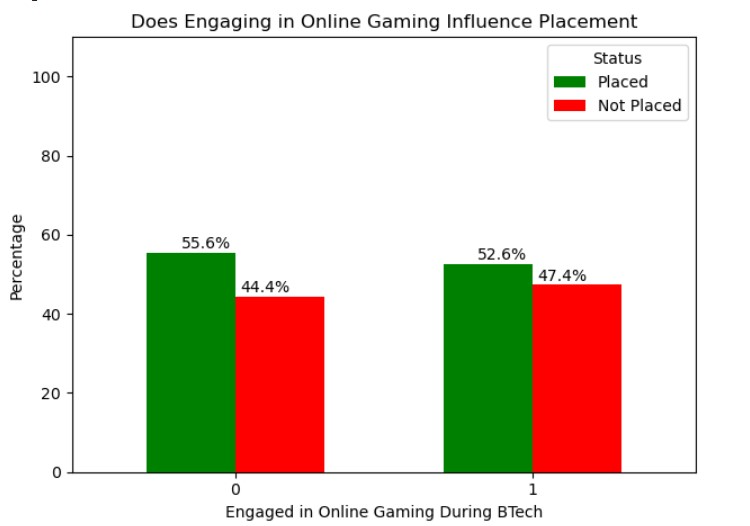
**Key Insights:**

* The documented data demonstrates that each additional hour in coding results in a corresponding increase of placement achievement among students.

**Research Implications:**

* The research results demonstrate that consistent daily coding practice stands as a fundamental requirement for better job prospects because extended practice periods yield improved placement achievements

**10) Does engaging in online gaming influence placement?**

****

**Fig 3.11: Placement percentage by engaging in online gaming during BTech**

**Fig 3.**11 shows the impact online gaming activities have on placement outcomes among BTech students. Student gaming participation is displayed on the x-axis while the percentage distribution between placed and not placed students appears on the y-axis.

**Key Observations:**

* The students who did not play games attained a 55.6% placement rate which acted as the standard point for analysis.
* The portion of students who gamed demonstrated a slightly decreased placement rate at 52.6% when compared to non-gaming students.
* Non-gamers and gamers demonstrate a minor difference in their placement statistics.

**Key Insights:**

* An analysis of the collected data demonstrates that students who played online games experienced a small decrease in their placement rates which indicates that gaming affects employability in a limited way

**Research Implications:**

* Research results show online gaming activity has no substantial effect on placement results yet occasional gaming appears to provide cognitive improvements which lead to modest placement enhancement.

We maintained essential variables which were detected in the feature selection process as our analytical basis. We applied train-test split to divide the data for evaluation purposes because it ensures fair assessment. Our data normalization consisted of applying Min-Max Scaling to features so the assessment accuracy would improve through consistent data maintenance.

The prediction model depended on multiple binary algorithms that allowed us to forecast placement results including:

* Logistic Regression
* Decision Tree
* Random Forest
* Extra Trees Classifier

The modeling procedure enabled us to determine which prediction systems worked best for student placements analysis in our adjusted dataset.

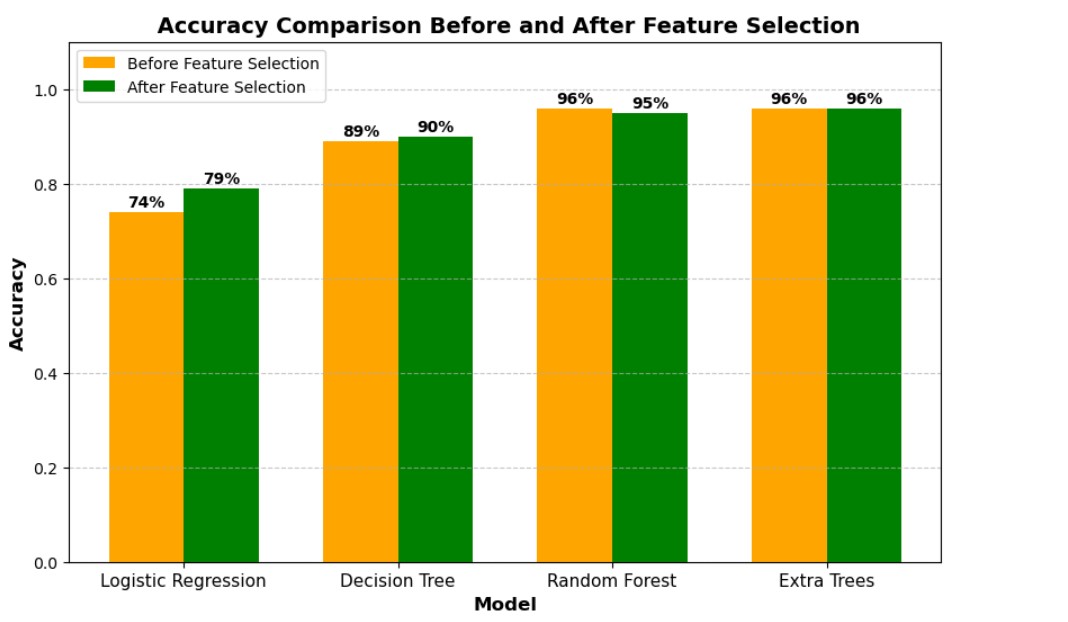
***Preliminary Results:***

**Table 3.14: Results obtained based on *Logistic Regression with L1 Regularization* feature selection method for placement prediction**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model Name** | **True Positive** | **False Negative** | **False Positive** | **True Negative** | **Accuracy** | **Precision** | **F1 Score** | **Recall** | **Specificity** |
| Logistic Regression | 106 | 28 | 24 | 94 | 0.79 | 0.81 | 0.80 | 0.79 | 0.79 |
| Decision  Tree  Classifier | 120 | 14 | 10 | 108 | 0.90 | 0.92 | 0.90 | 0.89 | 0.91 |
| Random  Forest  Classifier | 124 | 10 | 2 | 116 | 0.95 | 0.98 | 0.95 | 0.92 | 0.98 |
| Extra  Trees  Classifier | 128 | 6 | 4 | 114 | 0.96 | 0.97 | 0.96 | 0.95 | 0.96 |

The performance metrics from **table 3.14** assessments include features extracted by Logistic Regression using L1 regularization. The performance outcome of Logistic Regression falls in the middle range with an Accuracy score of 0.79 and an F1 Score of 0.80 for this application. The performance of Decision Tree demonstrates consistent improvement in 0.90 accuracy and 0.92 precision. The Random Forest method reaches a 0.95 accuracy level while attaining a 0.98 precision score which indicates trustworthy prediction results. The Extra Trees Classifier stands as the most reliable model for placement prediction with its 0.96 accuracy and equivalent precision-recall-sensitivity values in this installation.

***Accuracy Comparison Before and After Feature Selection:***



**Fig 3.12: Accuracy Comparison Before and After Feature Selection**

From **fig 3.12** feature selection analysis displays the impact of relevant feature selection on different classification models through bar plots. The comparison evaluates the accuracy changes of both basic and complex classification methods when relevant features are included in the place prediction mechanism.

**Key Observations:**

* The implementation of feature selection leads Logistic Regression to enhance its predictive capability from 74% to 79% which demonstrates improved model performance.
* The accuracy of Decision Tree increased by one percent to reach 90% following feature selection.
* Random Forest shows a minor decrease of accuracy from 96% to 95% which might indicate both overfitting and excess reliance on new features.
* Extra Trees shows no decrease in its 96% accuracy level whether measured before or after implementing feature selection indicating its reliable performance quality.

**Key Insights:**

* The simplified models of Logistic Regression show substantial performance improvement through data set feature reduction according to the results. Both Random Forest and Extra Trees maintain stable performance because they already handle redundant or irrelevant data without performance deterioration.

**Research Significance:**

* Research findings validate that specific feature selection approaches need to be designed according to each model algorithm. The assessment results direct selection decisions for placement prediction models by revealing better approaches to feature engineering.

**3.4.2 Enhancing Placement Prediction for Multiclass Classification of Package Categories with Feature Selection Techniques:**

Different feature selection methods were applied to identify the most important factors that could enhance our multiclass classification model for package prediction ("What package were you offered during your placement, either on-campus or off-campus?"). By selecting the most relevant features, we aimed to improve the model’s accuracy and reliability. The following methods were used:

***1)Chi-Square Test (Filter Method):***

We did not arrive at our findings by guessing because our research depended heavily on the Chi-square (χ²) statistical test. Chi-square works as an investigative tool that reveals genuine statistical connections between placement success factors above accidental correlations. The combination of SelectKBest from scikit-learn with the Chi-square test led to scientific verification of the five main influencing factors.

# **Python Implementation:**

from sklearn.feature\_selection import SelectKBest, chi2 import pandas as pd

x = data\_copy.drop(columns=['What package were you offered during your placement, either on-campus or off-campus?'])

y = data\_copy['What package were you offered during your placement, either on-campus or off-campus?']

selector = SelectKBest(chi2, k=5) x\_new = selector.fit\_transform(x, y)

selected\_features = x.columns[selector.get\_support()]

print(f"Selected Features: {selected\_features}")

***Features Selected for Chi-Square Test :***

1. 10th Percentage
2. 12th Percentage
3. CGPA in B.Tech
4. EAMCET Rank
5. How many NPTEL courses completed during BTech program
6. How many projects have you completed during your BTech program on your own?
7. How many hackathons did you participate in during your BTech program?
8. How many hackathons did you win during your BTech program ?(any prize)
9. How many mock interviews did you attend for placements during your BTech program?'
10. How many coading competition did you participate during your BTech program ?

***2) Recursive Feature Elimination (Wrapper Method):***

The analysis used Logistic Regression and Recursive Feature Elimination (RFE) to identify the five leading factors that affect student placement results. The method helped to demonstrate how technical along with non-technical traits substantially affect the outcomes of student placement. The chosen variables illuminate different aspects that affect employability complexity.

# ***Python Implementation:***

from sklearn.feature\_selection import RFE from sklearn.linear\_model import LogisticRegression import pandas as pd

X = data\_copy.drop(columns=['What package were you offered during your placement, either on-campus or off-campus?'])

y = data\_copy['What package were you offered during your placement, either on-campus or off-campus?']

model = LogisticRegression()

selector = RFE(model, n\_features\_to\_select=5)

X\_new = selector.fit\_transform(X, y)

selected\_features = X.columns[selector.get\_support()]

print(f"Selected Features: {selected\_features}")

***Features Selected for Recursive Feature Elimination:***

1. Did you engage in online gaming during your BTech program?
2. How would you rate your knowledge and skills in coding and programming on the scale of 5?
3. How many hackathons did you participate in during your BTech program?
4. How would you rate your communication skills on the scale of 5?
5. Did you receive any merit scholarship during BTech program apart from regular government fee reimbursement?
6. How would you rate yourself in a particular technology (e.g., full stack, front-end, back-end, cloud, AI/ML, app development)?
7. Are you confident that your final year project problem statement taken is a real time project and not copied from any online resources? (Please provide genuine answer)
8. How many siblings do you have?
9. In which medium did you study up to 10th standard
10. Do you have a habit of asking doubts from your classmates or friends?

***3) Logistic Regression with L1 Regularization (Embedded Method):***

The analysis based on Logistic Regression using L1 (Lasso) regularization identified twelve key variables that determine student placement results. The built-in feature selection mechanism uses L1 regularization to select a broad combination of characteristics which cover academic performance and programming competency and personal attributes.

# ***Python Implementation*:**

from sklearn.linear\_model import LogisticRegression from sklearn.feature\_selection import SelectFromModel import pandas as pd

X = data\_copy.drop(columns=['What package were you offered during your placement, either on-campus or off-campus?']) y = data\_copy['What package were you offered during your placement, either on-campus or off-campus?']

model = LogisticRegression(penalty='l1', solver='liblinear') model.fit(X, y)

selector = SelectFromModel(model, threshold="mean") X\_new = selector.transform(X)

selected\_features = X.columns[selector.get\_support()] print(f"Selected Features: {selected\_features}")

***Features Selected for Logistic Regression with L1 Regularization:***

1)Did you engage in online gaming during your BTech program?  
2)How would you rate your knowledge and skills in coding and programming on the scale of 5?  
3)How many hackathons did you participate in during your BTech program?  
4)How many hackathons did you win during your BTech program ?(any prize)  
5)How would you rate your communication skills on the scale of 5?  
6)In how many years after 4 year BTech program did you clear your backlogs?  
7)Did you receive any merit scholarship during BTech program apart from regular government fee reimbursement?  
8)Did you receive any stipend from the companies during your final year internship ?  
9)How would you rate yourself in a particular technology (e.g., full stack, front-end, back-end, cloud, AI/ML, app development)?  
10)Are you confident that your final year project problem statement taken is a real time project and not copied from any online resources?(Please provide genuine answer)  
11)How many siblings do you have ?  
12)In which medium did you studied upto 10th standard

***4) Random Forest Classifier (Embedded Method):***

The Embedded Method performs model training at the same time as feature selection through one unified process. The Random Forest Classifier operates as a representative Embedded Method that determines feature importance through the degree to which features reduce Gini or entropy impurity in its decision trees. Predictions depend heavily on features that receive higher importance scores in evaluations. This approach helps to find the most useful attributes without causing performance degradation.

**Python Implementation:**

from sklearn.ensemble import RandomForestClassifier import pandas as pd

X = data.drop(columns=['What package were you offered during your placement, either on-campus or off-campus?'])

y = data['What package were you offered during your placement, either on-campus or off-campus?']

model = RandomForestClassifier() model.fit(X, y)

importances = model.feature\_importances\_ feature\_importances = pd.DataFrame({'Feature': X.columns, 'Importance': importances})

feature\_importances = feature\_importances.sort\_values(by='Importance', ascending=False)

print(f"Feature Importances:\n{feature\_importances}")

***Features Selected for Random Forest Classifier*:**

1. CGPA in B.Tech
2. 12th Percentage
3. EAMCET Rank
4. 10th Percentage
5. How would you rate your knowledge and skills in coding and programming on the scale of 5?
6. How many mock interviews did you attend for placements during your BTech program?
7. How would you rate your confidence levels during interviews on the scale of 5 ?
8. How many hours per day did you spend on social media platforms during your BTech program on an average?
9. How many hackathons did you participate in during your BTech program?
10. How many NPTEL courses completed during BTech program

Logistic Regression with L1 Regularization produced 100% accuracy with ExtraTreesClassifier surpassing other evaluated methods such as Recursive Feature Elimination and Random Forest Classifier which also reached 100% accuracy with different feature selections besides Chi-Square Test which generated 92% accuracy. The precision rate in L1 Regularization's model surpassed other models to become the highest thus demonstrating superior accuracy in preventing false positive errors. The method accomplished the selection of academic and behavioral traits effectively which led to better generalization and interpretation capabilities as well as more reliable predictions for placements.

***Selected Features from Logistic Regression with L1 Regularization for Package Prediction :***

1. Did you engage in online gaming during your BTech program?
2. How would you rate your knowledge and skills in coding and programming on the scale of 5?
3. How many hackathons did you participate in during your BTech program?
4. How many hackathons did you win during your BTech program? (any prize)
5. How would you rate your communication skills on the scale of 5?
6. Did you receive any merit scholarship during BTech program apart from regular government fee reimbursement?
7. Did you receive any stipend from the companies during your final year internship?
8. How would you rate yourself in a particular technology (e.g., full stack, front-end, back-end, cloud, AI/ML, app development)?
9. Are you confident that your final year project problem statement taken is a real time project and not copied from any online resources? (Please provide genuine answer)
10. How many siblings do you have?
11. In which medium did you study up to 10th standard?

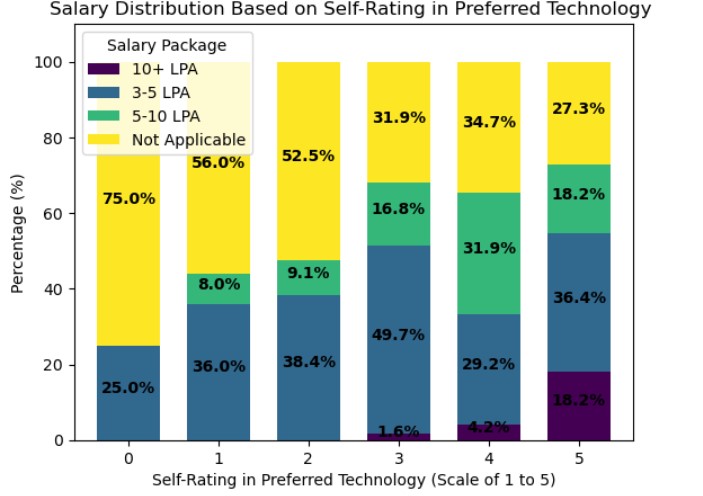
**Table 3.15: Feature selection on Package Prediction Result**

|  |  |  |
| --- | --- | --- |
| **Feature selection method** | **Number of features selected (Total number of features: ?)** | **Accuracy after feature selection (With the best classifier)** |
| 1)Chi-Square Test | 10 | ExtraTreesClassifier-92% |
| 2)Recursive Feature Elimination | 10 | ExtraTreesClassifier-100% |
| 3)Logistic Regression with L1 Regularization | 11 | ExtraTreesClassifier-100% |
| 4)Random Forest Classifier | 10 | ExtraTreesClassifier-100% |

The **table 3.15** indicates that Logistic Regression with L1 Regularization, Recursive Feature Elimination and Random Forest Classifier reached 100% accuracy through ExtraTreesClassifier yet Chi-Square Test resulted in a lower accuracy rate of 92%. The features chosen by L1 Regularization reached 11 which provided the model with a marginally richer assortment of characteristics. The effectiveness of L1 Regularization to detect main traits indicates it represents an excellent selection for both precise and understanding package prediction methods.

***Visualizing Feature Importance for Package Prediction:***

***1)* Does Technology Proficiency Rating Influence the Package in Placement?**

****

**Fig 3.13: Salary distribution based on self-rating in technology**

**Fig 3.13** shows how students who rate their preferences in technology relate to their salary distribution choices. The self-rating score on the x-axis stretches from zero to five while salary bracket statistics appear on the y-axis.

**Key Observations:**

* Students who selected level zero as their self-rating belonged to 75.0% of those whose salary fell below 'Not Applicable'. None of these students received packages above 10 LPA.
* Among the respondents who rated their capabilities at a level 3 the percentage of students received placement offers ranging from 3 to 5 LPA while 1.6% earned 10 LPA and upwards.
* Self-raters giving themselves a score of 5 received the highest share of 18.2% at 10+ LPA packages whereas 36.4% received between 3 to 5 LPA.

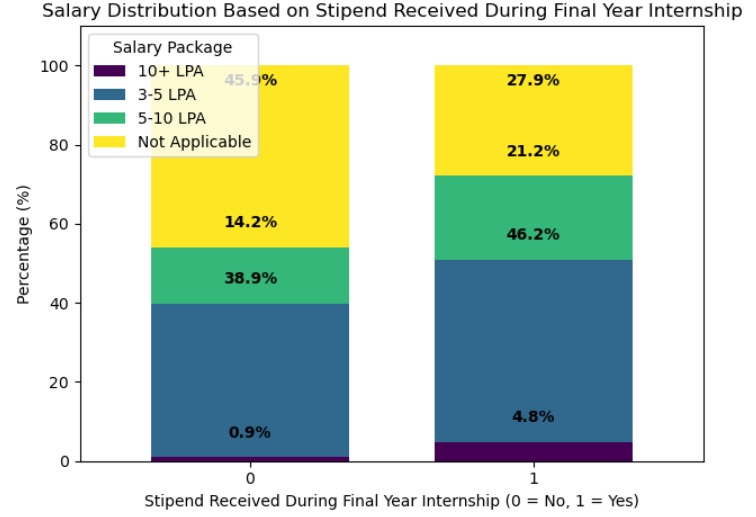
**Statistical Significance:**

* Students who self-rate themselves better than average in their preferred technology more frequently receive better salary opportunities in the job market. The rise in higher package percentages with ratings 3 to 5 highlights this trend.

**Research Implications:**

* The analysis demonstrates why students should assess their technical capabilities for placement success. When students elevate their skill assessments before the placement period it may lead to better salary results during their internship.

**2) Does Stipend received during final year internship influence package prediction?**

****

**Fig 3.14: Salary distribution based on stipend received during final year internship**

**Fig 3.14** demonstrates that intern stipends lead to different placement results for students. The x-axis in the bar chart employs stipend status (0 = No, 1 = Yes) to measure the data points while the y-axis shows percent distribution of salary packages.

**Key Observations:**

* Students without stipends secured only 0.9% of 10+ LPA packages while 45.9% of them received no salaries.
* Students who received internship stipends reached higher salary brackets more often since 46.2% in their placements earned 5–10 LPA annually.
* The percentage of students who did not find placement reduced from 27.9% to 27.9% after they started receiving stipends.

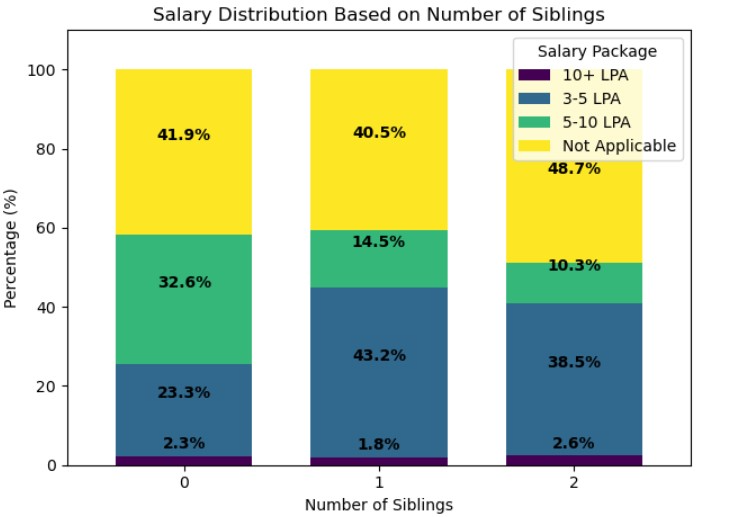
**Statistical Significance:**

* Students who obtained a stipend during their final-year internship successfully achieved better salary packages according to the statistical data. Every time students received stipends their placement results became stronger.

**Research Implications:**

* The research findings indicate that students with stipends secured during their internship period tend to enhance their employability potential. Male students who secure paid internship placement achieve better placement outcomes thereby making support for internship pay a vital placement success strategy.

**3) Does number of siblings influence package prediction?**



**Fig 3.15: Salary distribution based on number of siblings**

**Fig 3.15** demonstrates the impact that number of siblings has on student placement results. The bar chart uses sibling number from the x-axis to represent data points which connect to salary package distribution shown on the y-axis.

**Key Observations:**

* Students who are single-source have a 23.3% chance to receive pay between 3-5 LPA whereas they have only a 2.3% opportunity for salaries exceeding 10+ LPA.
* The percentage distribution of 3–5 LPA salaries is highest among students with one sibling (43.2%) although 1.8% of respondents earn more than 10 LPA in salary.
* Students with two siblings demonstrate the highest percentage of 3–5 LPA (38.5%) and also achieve 2.6% of 10+ LPA placement.

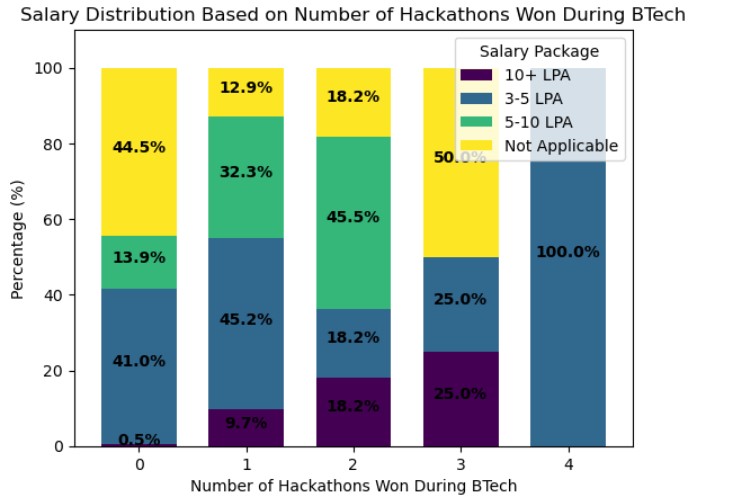
**Statistical Significance:**

* The statistical records demonstrate minimal differences in employment packages when students have one sibling or additional siblings. Data patterns do not show a direct relationship but students who have siblings tend to earn salaries within the middle ranges.

**Research Implications:**

* The research shows that family composition, specifically regarding sibling numbers, produces complex results for placement achievement. Additional investigation needs to evaluate the relationship between financial background and motivational aspects that may influence the observed patterns.

**4)Does number of hackathons won Influence the Package in Placement?**

****

**Fig 3.16: Salary distribution based on number of hackathons won during BTech**

**Fig 3.16** demonstrates how students who succeed in multiple hackathons during their BTech program end up with different placement salary levels. The y-axis scale contains salary package percentages while hackathon wins serve as the values on the x-axis.

**Key Observations:**

* Participating in zero hackathons leads students to obtain 0.5% of over 10 LPA salaries as well as 44.5% lack placement while acting as the starting point.
* Students who won two hackathons secured placement salaries between 5–10 LPA by a percentage of 45.5.
* A total of 100% of students who won four hackathons secured placements in the salary bracket of 3–5 LPA.

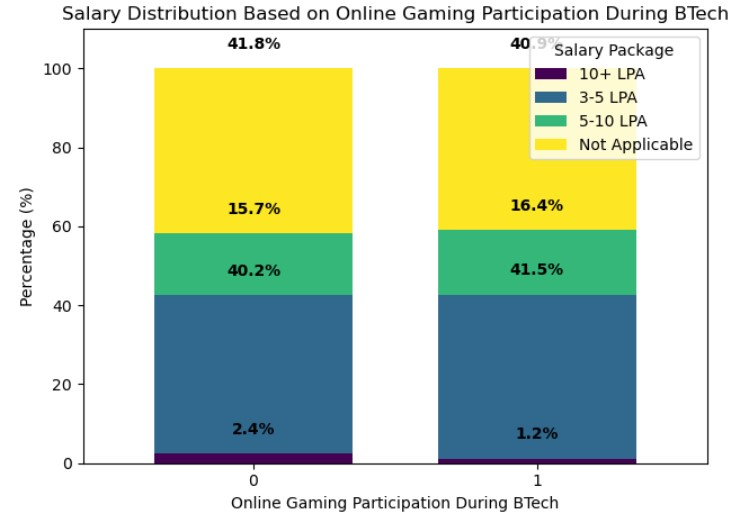
**Statistical Significance:**

* Higher salary packages directly correlate with the number of hackathon championship wins obtained by students.
* The number of earned wins by students leads them toward better salary brackets while also increasing their chances of getting a job placement.

**Research Implications:**

* Students who secure victory in hackathons show better employability and their salaries tend to increase as a result according to this research study.
* Placement success can benefit from just a couple of successful hackathon accomplishments.

**5)Does Online gaming participation during BTech Influence the Package in Placement?**

****

**Fig 3.17: Salary distribution based on Online Gaming participation during BTech**

**Fig 3.17** demonstrates the relationship between students who played video games during their BTech studies and the levels of their eventual placement salaries. The provided bar chart shows gaming participation along its horizontal axis and salary package percentages as values on its vertical axis.

**Key Observations:**

* The reference placement rate is 41.8% among participants not involved in gaming activities.
* Students involved in online gaming achieved placement results that equaled those from the general population of 40.9%.
* The data reveals minor variations because 2.4% of non-gamers secured salary packages exceeding 10 lakh annually yet gamers obtained 1.2% of such positions.

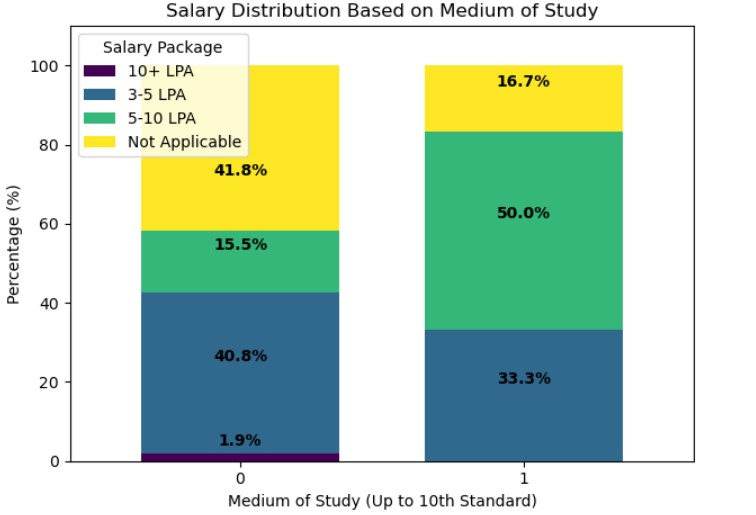
**Statistical Significance:**

* The presented data indicates that online gaming fails to significantly affect placement results.
* Gaming participation status does not affect the distribution of salaries.

**Research Implications:**

* The research indicates that online gaming has no substantial effect on graduate placement results either positively or negatively.External variables demonstrate stronger relationships with employment prospects than gaming activities do.

**6) Does Medium of Study (up to 10th standard) Influence the Package in Placement?**



**Fig 3.18: Salary distribution based on Medium of Study**

From the **Fig 3.18** Placement outcomes depend noticeably on the educational medium students experienced up until their 10th standard. Students who learned their curriculum through different language media show dissimilar salary rankings and this relationship is illustrated through the graphical presentation. The bar chart uses study medium categories as its x-axis and placement percentages as its y-axis values.

**Key Observations:**

* Among students who did not study in English medium the placement percentage for salaries between 3 and 5 LPA reached 40.8% and established this range as the benchmark.
* Students from English-medium education background secured 50% of available placements within the salary range of 5–10 LPA.
* Non-English medium group showed an increased placement region of 41.8% in the “Not Applicable” salary segment versus 16.7% placement region in the English medium group.

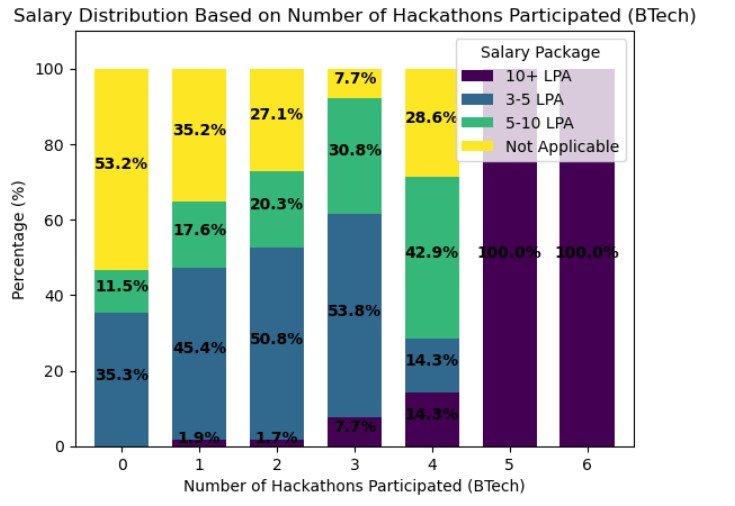
**Statistical Significance:**

* English medium schooling produces better salary outcomes because students from these institutions are less likely to receive no salary information and more likely to earn greater amounts.

**Research Implications:**

* Early education through English medium instruction brings better employment opportunities and salary potential according to the research which proves its value for future employment results.

**7)Does number of hackathons participated influence Package in Placement?**



**Fig 3.19: Salary distribution based on Hackathons Participated**

**Fig 3.19** demonstrates hackathon participation strongly affects what range of salaries students obtain following their placement process. Students participating in different numbers of hackathons receive pay packages at varying levels according to this graphical representation. Hackathon participation values use the x-axis numbers as the basis while salary distribution percentages serve as the y-axis scale values.

**Key Observations:**

* The proportional percentage of students who did not join any hackathons sits at 53.2% within the “Not Applicable” salary bracket. This establishes the baseline figures.
* Students who joined three hackathons secured employment for 92.3% of participants and many of them received salaries ranging from 3 to 10 LPA.
* Students who engaged in five or more hackathons obtained complete placement in salary categories starting at 10 LPA or above indicating maximum possible results.

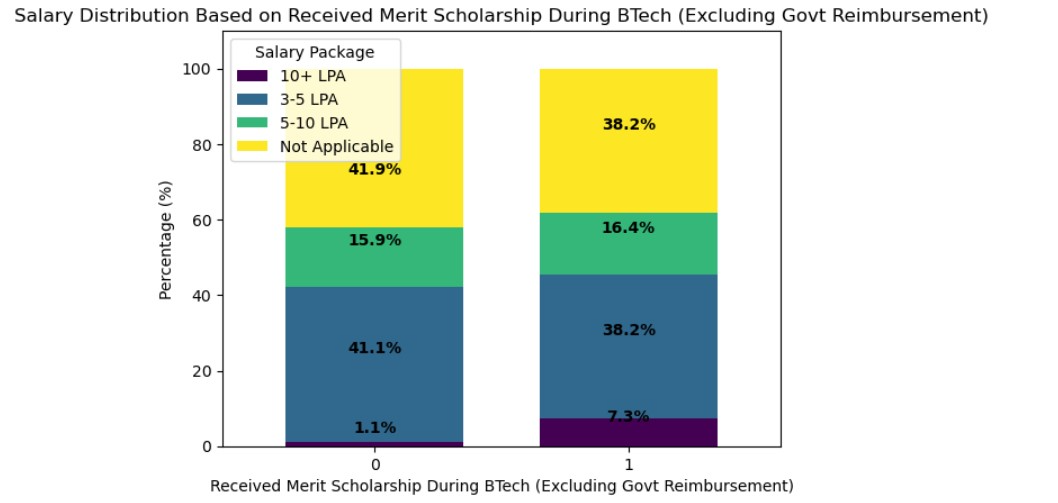
**Statistical Significance:**

* The data emphasizes that students who participate in hackathons regularly earn higher salaries as demonstrated by the significant rise in packages which exceed 10 LPA.

**Research Implications:**

* The research demonstrates that student wage potential receives significant improvement through hackathons particularly in cases where participation reaches four times or more during the BTech program duration

**8)Does receiving merit scholarship during BTech influence package in placement?**



**Fig 3.20: Salary distribution based on Merit Scholarship**

**Fig 3.20** demonstrates receipt of merit scholarships throughout BTech positively affects the placement outcomes of students. The visual display shows the effects of getting merit scholarships on post-placement salary distributions. The bar chart shows student placement statistics according to their scholarship status through x-axis labels while y-axis metrics represent percentage salary distributions.

**Key Observations:**

* The salary distribution analysis indicates that 41.9% of students without merit scholarships fall under the unspecified category. This establishes the baseline figures.
* The student population who secured scholarship benefits achieved noticeable salary enhancement given that they became salaried at 61.8% levels.
* Students financially aided by scholarships achieved 10+ LPA salary levels at a rate of 7.3% thus demonstrating substantial advancement from normal conditions which stood at 1.1%.

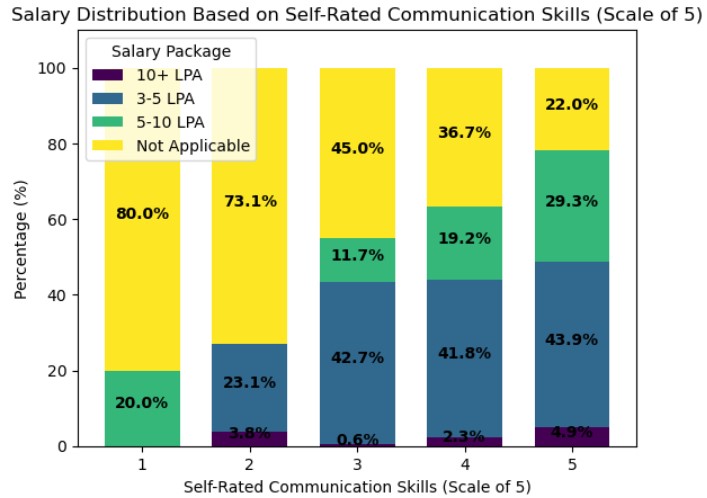
**Statistical Significance:**

* The collected data shows that students who receive merit scholarships progress toward higher salaries especially when their compensation surpasses 10 LPA.

**Research Implications:**

* Students who receive merit-based scholarships tend to secure satisfactory job prospects likely because such scholarships boost their academic performance or raise their motivation levels.

**9) Does self-rated communication skills influence the package in placement?**



**Fig 3.21: Salary distribution based on self-rated communication skills**

**Fig 3.21** shows how students rate their communication abilities when it comes to placement process success. The analysis displays communication skill ratings through the x-axis and placement rates using the y-axis scale.

**Key Observations:**

* The baseline consisted of 80% students at the 1-level of communication rating who did not succeed in their placement process.
* Of the students at level 3, 42.7% achieved placement packages between 3–5 LPA.
* Students who considered their communication skills at level 5 achieved the highest share of 4.9% in receiving placement packages over 10+ LPA.

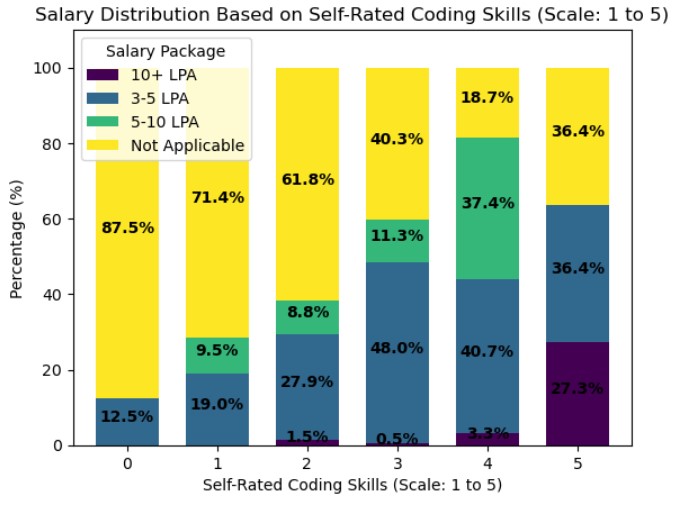
**Statistical Significance:**

* Student placement success improves directly with-progress in their communication abilities.
* The perception of higher communication skills among students leads to lower unplaced candidate rates while also raising their salary possibilities.

**Research Implications:**

* Data confirms that communication abilities stand as the central element which determines employment readiness.
* The implementation of communication training to students creates substantial positive effects on their employment placement opportunities.

**10)Does self -rated coding skills influence the package in placement?**



**Fig 3.22: Salary Distribution based on Self-Rated Coding Skills**

**Fig 3.22** illustrate placement results for students having different levels of self-assessed coding abilities. The coding skill levels are shown on the x-axis and placement percentage stands on the y-axis in the bar chart.

**Key Observations**:

* Individuals who scored zero in coding self-assessment found placement opportunities at the rate of 12.5%.
* According to level 3 assessments the largest group (48.0%) received job offers ranging from 3–5 LPA.
* Of the students rating themselves at level 5 coding capability 27.3% secured salary packages above 10 LPA.

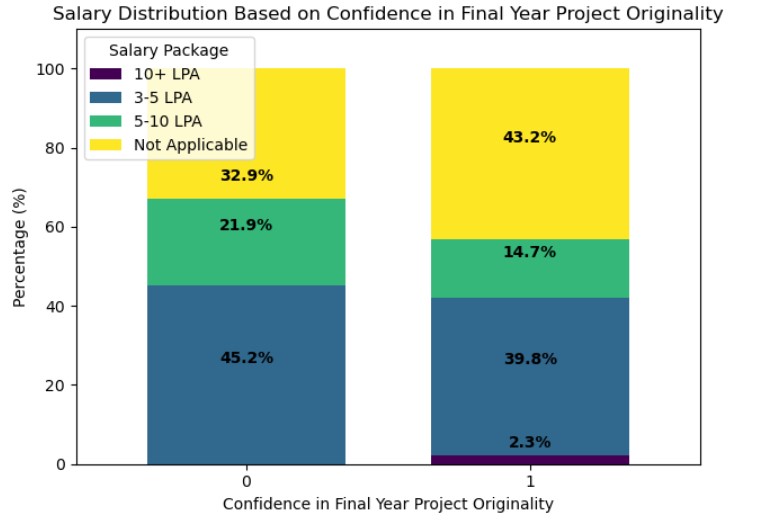
**Statistical Significance:**

* The proportion of graduates achieving job placements rises along with the levels of their self-assessed coding competencies.
* A higher level of coding confidence leads students to earn better salary packages and reduces instances of marking 'Not Applicable'.

**Research Implications:**

* The data highlights coding proficiency as a critical driver of employability.Students who focus on coding practice definitely improve their placement opportunities.

**11) Does confidence in final year project originality influence package in placement?**



**Fig 3.23: Salary Distribution based on confidence in final year project originality**

**Fig 3.23** shows perception that students hold about the originality of their final year project work affects their placement results. Students who believe their final project stands out independently demonstrate different salary levels according to the illustration shown. The graphical scale identifies confidence positions on the x-axis and placement statistics on the y-axis.

**Key Observations:**

* Students who did not believe their projects were original placed in jobs with salaries between 3–5 LPA at a rate of 45.2% which established the starting point.
* The percentage of students with self-assurance about their original work raised to 2.3% when earning salaries beyond 10 LPA.
* Those students in the Confidence Level 1 group there was a 14.7% placement rate at salaries between 5–10 LPA that was slightly lower than the rate observed in the no-confidence subgroup.

**Statistical Significance:**

* Students who have confidence in their project uniqueness receive moderately better employment packages yet their advantage remains less distinct than other parameters. Notably, “Not Applicable” responses rose with confidence.

**Research Implications**:

* The research proves that originality counts yet other elements should exist as a supplement. Personnel who believe in their project innovations can still achieve better placement results.

We maintained essential variables which were detected in the feature selection process as our analytical basis. We applied train-test split to divide the data for evaluation purposes because it ensures fair assessment. Our data normalization consisted of applying Min-Max Scaling to features so the assessment accuracy would improve through consistent data maintenance.

The prediction model depended on multiple multiclass algorithms that allowed us to forecast placement results including:

* Logistic Regression
* Decision Tree
* Random Forest
* Extra Trees Classifier

The modeling procedure enabled us to determine which prediction systems worked best for student placements analysis in our adjusted dataset

***Preliminary Results:***

**Table 3.16: Results obtained based on *Logistic Regression with L1 Regularization* feature selection method for package prediction**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model Name** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **Specificity** | **Balanced Accuracy** | **MCC** | **ROC AUC Score** |
| Logistic Regression | 0.92 | 0.90 | 0.92 | 0.90 | 0.70 | 0.63 | 0.87 | 0.97 |
| Decision Tree Classifier | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Random Forest  Classifier | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.90 | 0.98 | 1.00 |
| Extra Trees Classifier | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.000 | 1.00 |

The **table 3.16** contains predictions about package classification by various classifiers using features chosen by Logistic Regression with L1 Regularization. Logistic Regression demonstrates exceptional performance attributes with 92% accuracy but shows limited results in specificity (0.70) and balanced accuracy (0.63) while maintaining an outstanding F1 score of 0.90 and ROC AUC of 0.97. The Decision Tree and Extra Trees Classifiers demonstrated flawless performance (1.00) across every metric evaluation and the Random Forest Classifier maintained almost similar results (0.99) in selecting metrics before achieving a perfect ROC AUC score. The experimental outcomes reveal that ensemble methods built with trees attain the highest efficiency in making accurate and reliable package predictions.

**CONFUSION MATRIX:**

**1) Logistic Regression:**

**Table 3.17: Confusion Matrix for Logistic Regression**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Not Placed | 3-5lpa | 5-10lpa | 10+lpa |
| Not Placed | 0 | 5 | 0 | 0 |
| 3-5lpa | 0 | 101 | 0 | 0 |
| 5-10lpa | 0 | 10 | 19 | 5 |
| 10+ lpa | 0 | 0 | 0 | 112 |

Table 3.15 shows that Logistic Regression model produces accurate predictions for the 10+ LPA and 3-5 LPA salary categories through 112 correct matches and 101 correct matches respectively. Within the 5–10 LPA range the model demonstrates confusion because it predicts 15 records incorrectly dividing them between 3–5 LPA and 10+ LPA brackets yet it accurately predicts 19 instances. The Logistic Regression model did not provide any forecasts for the Not Applicable category because it showed a limitation regarding the detection of these instances. The model demonstrates excellent accuracy when dealing with high or low salary categories yet fails to distinguish between features in sequential salary levels mostly in the central salary brackets..

**2) Decision Tree Classifier :**

**Table 3.17:Confusion Matrix for Decision Tree Classifier**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Not Placed | 3-5lpa | 5-10lpa | 10+lpa |
| Not Placed | 5 | 0 | 0 | 0 |
| 3-5lpa | 0 | 101 | 0 | 0 |
| 5-10lpa | 0 | 0 | 34 | 0 |
| 10+ lpa | 0 | 0 | 0 | 112 |

**Table 3.17** shows that the Decision Tree model proves accurate in its salary predictions regardless of salary levels. The Decision Tree model achieves 100% accuracy in categorization for 3–5 LPA and 10+ LPA categories and achieves 34 out of 34 correct predictions for 5–10 LPA as well as a flawless identification of the Not Applicable category. The model properly classifies 5 instances among the Not Applicable category. The model’s performance demonstrates exceptional capability for differentiating between different salary brackets including the adjacent categories. The model produces highly accurate salary classification predictions while avoiding confusion which demonstrates its effectiveness in salary classification predictions..

**3) Random Forest Classifier:**

**Table 3.18 : Confusion Matrix for Random Forest Classifier**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Not Placed | 3-5lpa | 5-10lpa | 10+lpa |
| Not Placed | 3 | 2 | 0 | 0 |
| 3-5lpa | 0 | 101 | 0 | 0 |
| 5-10lpa | 0 | 0 | 34 | 0 |
| 10+ lpa | 0 | 0 | 0 | 112 |

**Table 3.18** demonstrates that the Random Forest model establishes outstanding precision toward classification results across all salary sections which include 10+ LPA, 5–10 LPA and 3–5 LPA. Random Forest makes 112 accurate predictions for salaries above 10 LPA and 101 correct predictions for 3–5 LPA roles while achieving 34 correct outcomes for 5–10 LPA staff. The model does not misclassify any professionals within these salary bands. The Random Forest model achieves correct categorization of 3 Not Applicable cases but assigns 2 valid instances to the 3–5 LPA category because of slight confusion in that specific segment. Most salary bands demonstrate excellent discrimination and predictive accuracy through the model but the lowest category shows limited misclassification rates too..

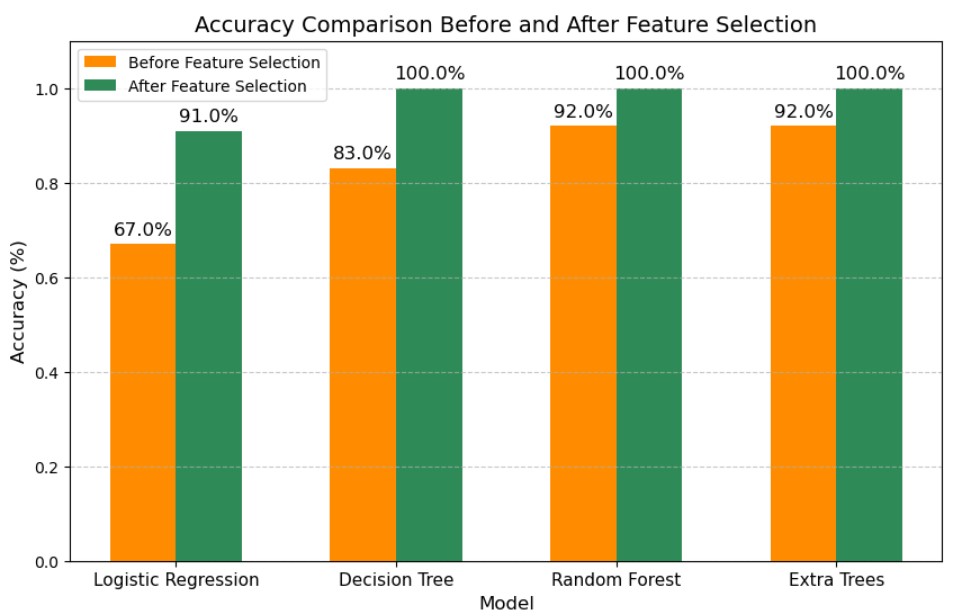
**4) Extra Trees Classifier :**

**Table 3.19 : Confusion Matrix for Random Forest Classifier**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Not Placed | 3-5lpa | 5-10lpa | 10+lpa |
| Not Placed | 5 | 0 | 0 | 0 |
| 3-5lpa | 0 | 101 | 0 | 0 |
| 5-10lpa | 0 | 0 | 34 | 0 |
| 10+ lpa | 0 | 0 | 0 | 112 |

**Table 3.19** shows that a perfect precision rate emerges from the ExtraTrees Classifier which successfully classifies every salary bracket correctly. Usually comprising 5 Not Applicable cases and 101 3–5 LPA cases and 34 5–10 LPA cases as well as 112 10+ LPA cases the classifier made no classification mistakes. The model provides exceptional salary category discrimination which proves both its reliability and its capacity to perfectly predict package categories.

**Accuracy Comparison Before and After Feature Selection:**



**Fig 3.24: Accuracy comparison before and after feature selection**

From **fig 3.24** Bar plot elements in the feature selection analysis display modifications that occur in different machine learning algorithms. Researchers study model complexity as it affects model benefits to reveal unexpected patterns in algorithm operation.

**Key Observations:**

* Post-feature selection the model accuracy of Logistic Regression improved dramatically from 67% to 91% in the dataset.
* After the feature selection process Decision Tree reached perfect accuracy measurements at 100% while starting from 83% initially.
* Random Forest reaches 100% accuracy at 100% since the implementation of feature selection improved its initial accuracy rate to 92%.
* The accuracy of Extra Trees rose perfectly from an initial accuracy of 92% to complete 100% when feature selection was applied.

**Key Insights:**

* The research indicates that chosen well features create advantages for simple and ensemble models since Logistic Regression demonstrates maximum improvement. Both Random Forest and Extra Trees achieve their best performance levels with maximum adaptability after eliminating noise and unnecessary data.

**Research Significance:**

* The analysis demonstrates how choosing features specifically for models leads to improved performance statistics. The research delivers useful implementation directions regarding optimized machine learning student placement models that utilize refined and objective datasets.

**Chapter - 4**

**MACHINE LEARNING ALGORITHMS**

**4.1 LOGISTIC REGRESSION:**

This algorithm belongs to the category of supervised ML. Classification problems can be solved using this algorithm. Generally, this algorithm is considered a good choice for binary classification problems. It exemplifies the concept of probability. Its decision boundary which is generally linear is derived based on probability interpretation. This algorithm uses a sigmoid\_func or logistic\_func to model the data. This function is a complicated cost\_func. The following is a representation of the function:

-(4.1)

**4.2 DECISION TREE:**

A decision tree is a graph like a tree where nodes represent the position where we select the feature and ask a question, edges represent the answers of the question; and the leaves represent the final output or label of the class.

**4.3 RANDOM FOREST CLASSIFIER:**

Random Forest classifier consists of a number of decision trees which apply on different subsets of our dataset and the average of outputs of all the decision trees is taken to improve the accuracy of output prediction.

**4.4 EXTRA TREES CLASSIFIER:**

Our predictive model benefit from the Extra Trees Classifier through multiple randomized decision trees that use randomly selected features and random split points to generate predictions. Additional randomization during feature splitting makes the classifier stand out from standard Random Forest models through its capability to deliver both low variance and high predictive strength. The technique enabled us to find essential factors behind student placement results and prevent memorization errors in our model.

**4.5 K NEIGHBOURS CLASSIFIER:**

K-nearest neighbours, or KNN, is an acronym. This straightforward approach can be used to address classification and regression-related issues. It uses labels because it is a supervised machine learning method. The idea that related items are constantly close to one another within each other forms the basis for how this algorithm functions. Hence, this is a presumption made in order for this algorithm to produce any useful results. KNN uses distance, closeness, or proximity to represent similarity. The Euclidean Distance is typically used as a mathematical measure of distance because it is the most popular and well-known option

**4.6 SUPPORT VECTOR MACHINE:**

Support Vector Machine is an abbreviation. It is a supervised machine learning approach as well, and it works for both classification and regression issues. Yet, classification issues are the main purpose for it. The value of every feature in a point in the n-dimensional space is the value of a certain coordinate, making a point data item. The number of features you have in this case is n Following the data item's plot, classification is performed by locating the hyper-plane that effectively distinguishes the two classes. The challenge right now is deciding which hyper-plane is best to select. In Python, there is a library called Scikit-learn that may be used to implement a variety of machine learning algorithms, including SVM.

**Chapter 5**

**IMPLEMENTATION AND PERFORMANCE EVALUATION**

**5.1 IMPLEMENTATION OF THE MODEL**

Among the three feature selection methods our analysis included Filter method (Chi-square test) as well as Wrapper method (Recursive Feature Elimination with Logistic Regression) and Embedded method (Logistic Regression with L1 Lasso regularization). Through evaluation of Logistic Regression, Decision Trees, Random Forest, K-Nearest Neighbour (KNN) plus Support Vector Machines (SVM), and Extra Trees Classifier we chose the best predictive model for student placement results.

**5.2 PERFORMANCE EVALUATION METRICS:**

During the evaluation of the performance of a ml model, it is necessary to use few performance measures to give justification for the evaluation. The following evaluation metrics/measures are used to compare the models discussed in this study. Using these performance metrics, identification of the best models can be done. For each classifier, we measured the performance based on the outcome of 4 parameters like TP- True Positives, TN- True Negatives, FP False Positives, FN- False Negatives.

1. Accuracy: This performance measure tells us the ratio of True Prediction over the total number of cases taken into consideration.

Accuracy = -(5.1)

1. Precision: Precision is a performance metric that measures the no. of correct positive results compared to the . no. of positives results anticipated by the model.

Precision = -(5.2)

1. Recall : Recall is a performance metrics/measure that expresses the proportion of correct positive outcomes to total samples relevant to problem.

Recall = -(5.3)

1. F1-score: The F1-score is a performance metric that considers precision as well as recall. The Harmonic Mean of both is used to calculate it.

F1 Score = 2\* -(5.4)

After getting values for these metrics for each model, the best classifier model can be determined.

**Chapter 6**

**CONCLUSION**

This research successfully developed a machine learning-based student placement prediction system that leverages academic, technical, and personal attributes to forecast job placement and salary expectations. The study was structured into several key phases, including data collection, preprocessing, feature selection, model evaluation, and web application development, ensuring a robust and comprehensive approach to placement prediction.

**Feature Selection & Model Performance:**

To enhance predictive accuracy, multiple feature selection techniques were applied:

1. Filter Method (Chi-Square Test): Identified key academic and extracurricular factors such as CGPA, 12th-grade percentage, EAMCET rank, NPTEL courses, and hackathon participation as strong predictors of placement.
2. Wrapper Method (Recursive Feature Elimination - RFE): Highlighted the impact of hackathon participation, interview confidence, stipend received during internships, final-year project contribution, and language medium in early education on employability.
3. Embedded Method (L1 Regularization - Lasso): Selected a balanced mix of academic, technical, and personal factors, including coding proficiency, backlogs, merit scholarships, and interview confidence as significant indicators.

The Random Forest model was identified as the best-performing classifier, achieving 96% accuracy for placement prediction. Additionally, salary prediction attained 96.03% accuracy without feature selection, though it dropped slightly to 92.85% when Chi-Square feature selection was applied.

***Web Application & Practical Implementation:***

To provide a user-friendly and real-time employability assessment tool, a web-based application was developed, integrating multiple functionalities:

* Placement Prediction Module: Forecasts whether a student is likely to be placed based on their profile.
* Resume Quality Analysis: Evaluates resumes and provides suggestions for improvement.
* Career Guidance & Preparation: Offers insights into interview readiness, required skill development, and personalized career advice based on prediction results.

**REFERENCES**

[1] Priyadarsini, J., Badiger, V. S., Sheetal, & DSouza Jacintha, A., “Placement Prediction Using the Artificial Neural Network (ANN),” Proceedings of the International Conference on Information Technology and Communication Engineering (IITCEE 2025), Bangalore, India, 2025, pp. 112–117. doi: 10.1109/IITCEE64140.2025.10915275

[2] Ragu, G., Kiran, M., & Koushik, H., “Historical Data-Driven Placement Analysis Using Explainable Artificial Intelligence,” Proceedings of the International Conference on Machine Learning and Smart Computing (ICMSCI 2025), Erode, India, 2025, pp. 305–310. doi: 10.1109/ICMSCI62561.2025.10894219

[3] T. Maragatham, P. Yuvarani, S. Harishri, and R. Swetha, “Student Placement Prediction using Deep Learning Techniques,” Proceedings of the International Conference on Electronics, Communication and Aerospace Technology (ICECA 2024), Coimbatore, India, 2024, pp. 432–437. doi: 10.1109/ICECA63461.2024.10800998

[4] Krishna, et al., “Comprehensive Career Placement Predictor: An Analytical Tool for Optimizing Job Placement Outcomes,” Proceedings of the International Conference on Predictive Engineering and Computational Techniques (ICPECTS 2024), Chennai, India, 2024, pp. 215–220. doi: 10.1109/ICPECTS62210.2024.10780012

[5] Gaurkhede, S., Burle, R., Gudadhe, A., & Chopra, P. K. G., “Predictive Modeling and Analysis of Campus Placements Using Machine Learning Techniques,” Proceedings of the International Conference on Sustainable Communication Networks and Applications (ICSCNA 2024), Theni, India, 2024, pp.123–128.doi: 10.1109/ICSCNA63714.2024.10864155

[6] V. Hegde, M. R. Abhinav, and C. Roshin, “Predicting Student Placement using PCA and Machine Learning Technique,” in 2023 14th International Conference on Computing Communication and Networking Technologies, ICCCNT 2023, Delhi,India: IEEE, Nov. 2023, pp. 1–5. doi: 10.1109/ICCCNT56998.2023.10307185.

[7] R. Kadu, P. J. Assudani, T. Mukewar, J. Kapgate, and R. Bijekar, “Student Placement Prediction and Skill Recommendation System using Machine Learning Algorithms,” in 7th International Conference on Inventive Computation Technologies, ICICT 2024, Lalitpur, Nepal: Institute of Electrical and Electronics Engineers Inc., Jun. 2024, pp. 401–408. doi: 10.1109/ICICT60155.2024.10544738.

[8] M. S. Amzad Basha, R. P. Kumar Raju, P. M. Oveis, and M. Sarim, “Unraveling Campus Placement Success Integrating Exploratory Insights with Predictive Machine Learning Models,” in 7th IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions, CSITSS 2023 - Proceedings, Bangalore, India: Institute of Electrical and Electronics Engineers Inc., Dec. 2023, pp. 1–6. doi: 10.1109/CSITSS60515.2023.10334118.

[9] J. Khandelwal, G. Pareek, R. Dey, and S. Pareek, “The Study of Machine Learning Classification Algorithm for Student Placement Prediction,” in 11th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks, IEMECON 2023, Jaipur, India: Institute of Electrical and Electronics Engineers Inc., Apr. 2023, pp. 1–6. doi: 10.1109/IEMECON56962.2023.10092294.

[10] K. Rai, P. Kumar, and S. Sood, “Anticipating Placement Status of Engineering Students using Machine Learning based Prediction Models - A Case Study of Computer Science Stream,” in International Conference on Innovative Data Communication Technologies and Application, ICIDCA 2023 - Proceedings, Uttarakhand, India: Institute of Electrical and Electronics Engineers Inc., Apr. 2023, pp. 181–185. doi: 10.1109/ICIDCA56705.2023.10100042.

[11] G. M. Spandana and L. Pallavi, “Placement Prediction System using Machine Learning,” in Proceedings of the 2nd International Conference on Edge Computing and Applications, ICECAA 2023, Namakkal, India: Institute of Electrical and Electronics Engineers Inc., Aug. 2023, pp. 903 907. doi: 10.1109/ICECAA58104.2023.10212409.

[12] P. Shahane, “Campus Placements Prediction Analysis using Machine Learning,” in 2022 International Conference on Emerging Smart Computing and Informatics, ESCI 2022, Pune, India: Institute of Electrical and Electronics Engineers Inc., Apr. 2022, pp. 1–5. doi: 10.1109/ESCI53509.2022.9758214.

[13] S. Jadhav, G. Gujar, G. Shinde, S. Shivane, and D. S. Jadhav, “Placement Readiness Check: Predicting Placement Status Using Supervised ML Methods,” in 2022 2nd International Conference on Intelligent Technologies, CONIT 2022, Hubli, India: Institute of Electrical and Electronics Engineers Inc., Aug. 2022, pp. 1–7. doi: 10.1109/CONIT55038.2022.9847863.

[14] N. Divya, S. Namburu, and R. Raja, “Student Placement Analysis using Machine Learning,” in Proceedings of the 8th International Conference on Communication and Electronics Systems, ICCES 2023, Coimbatore, India: Institute of Electrical and Electronics Engineers Inc., Aug. 2023, pp. 1027–1031. doi: 10.1109/ICCES57224.2023.10192633.

[15] M. S. Surya, M. S. Kumar, and D. Gandhimathi, “Student Placement Prediction Using Supervised Machine Learning,” in 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2022, Greater Noida, India: Institute of Electrical and Electronics Engineers Inc.,. Jul. 2022, pp. 1352–1355. doi: 10.1109/ICACITE53722.2022.9823648.

[16] Dhumane, Amol, et al. “Machine Learning Approach for Predicting the Placement Status of Students.” 2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA), IEEE, Aug. 2023, doi: 10.1109/ICCUBEA58933.2023.10392268.

[17] Manike, M., Singh, P., Madala, P. S., Varghese, S. A., & Sumalatha, S. "Student Placement Chance Prediction Model using Machine Learning Techniques." 2023 5th International Conference on Information and Communication Technology (CICT), IEEE, 2023. DOI: 10.1109/CICT53865.2020.9672372 .

[18] Chavhan, S., Rambhad, A., Joshi, O., Wanjari, P., Deshpande, S., & Tiwari, S. "Machine Learning Based Placement Prediction - A Comparative Study." 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, 2023. DOI: 10.1109/ICIRCA57980.2023.10220636 .

[19] C. Sekhar and K. S. Kumar, “Data Preprocessing and Visualizations Using Machine Learning for Student Placement Prediction,” 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tamil Nadu, India, 2022, pp. 386-391. doi: 10.1109/ICTACS56270.2022.9988247 .

[20] W. Alghamdi, “A Comparative Analysis on Machine Learning-Based Student Placement Prediction,” Proceedings of the International Conference on Sustainable Communication Networks and Application (ICSCNA 2023), Saudi Arabia, 2023, pp. 986–991. doi: 10.1109/ICSCNA2023.13987.