“PLACEMENT PREDICTION USING MACHINE LEARNING”

A Project report submitted in

partial fulfillment of the requirements for the Degree of

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

SOUMIK DAS

SANTU MAITY

Guided by

Prof. SURAJIT MANNA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RAMKRISHNA MAHATO GOVERNMENT ENGINEERING
COLLEGE, PURULIA 2024-25

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to **Prof. SURAJIT MANNA**, who at every discrete step in the study of this project, contributed with his valuable guidance and provided with perfect solution for every problem that arise. We also extend our sincere thanks to Head of our Department of Computer Science and Engineering **Prof. Dr. PRASUN HALDER** for his valuable guidance and constant encouragement throughout the duration of this project. His expertise and insights have been instrumental in shaping the direction and outcome of this project. We would also like to thank other professors of our department, for providing us with the opportunity and resources to pursue this project. Their support and vision have been very inspiring and motivating for us.

PLACE: PURULIA **SOUMIK DAS**

DATE:25/01/2025 **SANTU MAITY**

ABBREVIATIONS

Abbreviation	Details
ML	Machine Learning
RC	REGRESSION CLASSIFIER
UML	UNIFIED MODELING LANGUAGE
DFD	Data Flow Diagram

Table of Contents

Figure Index

Table Index

ABSTRACT

The aim of this project is to develop a machine learning model that can predict the placement percentage of a student based on their college grades and skills. The model can help students to assess their employability and improve their chances of getting hired by the industry. The methodology involves collecting data from various sources, such as academic records, skill assessments, and placement reports, and preprocessing it to extract relevant features. The data is then split into training and testing sets, and different algorithms, such as linear regression, decision tree, and neural network, are applied to train the model. The performance of the model is evaluated using metrics such as mean absolute error, root mean square error, and R-squared score. The results show that the neural network algorithm achieves the highest accuracy and lowest error among the tested algorithms. The model can accurately predict the placement percentage of a student with an error margin of less than 5%. The conclusion is that the model can be a useful tool for students and career counselors to plan and prepare for the placement process. The project also suggests some future directions for improving the model, such as incorporating more data sources, using more advanced algorithms, and deploying the model as a web application.

Key Words: Python, Class, Html CSS, Flask, Algorithm

CHAPTER - 1

INTRODUCTION

Welcome to Placement Predictor, a website that helps you estimate your chances of getting placed in your dream company. Whether you are a student, a fresher, or an experienced professional, this website can provide you with useful insights and tips to improve your profile and skills. Placement Predictor uses Flask, a lightweight web framework for Python, to create a dynamic and user-friendly interface. It also uses HTML and CSS to design and style the web pages. The website allows you to enter your details such as education, work experience, skills, certifications, etc. and then uses a machine learning model to predict your placement probability. The model is trained on a large dataset of previous placement records from various sources. The website also provides you with feedback and suggestions on how to increase your placement probability. You can learn about the requirements and expectations of different companies, the latest trends and technologies in the industry, and the best practices and resources for preparing for interviews and tests. You can also compare your profile with other candidates and see where you stand.

Placement Predictor is a website that aims to help you achieve your career goals and aspirations. It is not a guarantee or a promise of placement, but a tool that can guide and assist you in your journey. We hope you find it useful and informative. Thank you for visiting Placement Predictor.

Placement Predictor is a website that not only helps you with your placement prospects, but also improves the system of recruitment and hiring. By using data-driven and evidence-based methods, it reduces the bias and subjectivity that often affect the selection process. It also helps the employers and recruiters to find the best candidates for their roles, based on their skills and potential. Placement Predictor is a website that bridges the gap s between the job seekers and the job providers and creates a win-win situation for both parties.

Placement prediction is not a trivial task, as it involves dealing with complex and heterogeneous data, as well as modeling the dynamic and uncertain nature of the hiring process. Therefore, machine learning techniques can be applied to this task, as they can learn from data and make predictions based on patterns and relationships.

In this project, we propose to build a web API that can provide placement prediction services using machine learning. A web API is an interface that allows users to interact with a web application through HTTP requests and responses. A web API can expose the functionality of a machine learning model to various clients, such as web browsers, mobile apps, or other web services [1].

1.1 **MAIN COMPONENTS:**

A machine learning model that can predict the placement probability of a student given a set of input features, such as CGPA, backlogs, internships, etc. We will use a random forest classifier, which is an ensemble method that combines multiple decision trees to produce a more accurate and robust prediction. We will train and evaluate our model using a dataset of students' profiles and placement outcomes from a previous study.

A Flask web application that can serve the machine learning model and handle the user requests and responses. Flask is a lightweight and flexible web framework for Python that can be used to create web applications with minimal code. We will use Flask to create a web API that can accept HTTP requests with the input features of a student and return HTTP responses with the predicted placement probability and a confidence interval.

A web interface that can provide a user-friendly and interactive way to access the web API. We will use HTML, CSS, and JavaScript to create a web page that can display a form for the user to enter the input features of a student, and a chart for the user to visualize the prediction results. We will use AJAX to communicate with the web API asynchronously and update the web page without reloading.

1.2 **BACKGROUND:**

In recent years, the employment landscape has undergone significant transformations fueled by technological advancements and shifting market demands. Traditional recruitment methods, while effective in their own right, often fall short in accurately predicting the suitability of candidates for specific roles. This limitation has spurred the exploration of alternative approaches, such as machine learning, to enhance the efficiency and efficacy of the placement process. By harnessing the power of data analytics, organizations can gain deeper insights into candidate suitability, thereby optimizing resource allocation and mitigating hiring risks [3].

1.3 **MOTIVATION:**

The motivation behind this project stems from the pressing need to address the inefficiencies inherent in conventional recruitment practices. With an ever-expanding pool of job seekers and increasingly complex job requirements, there exists a palpable demand for predictive tools that can streamline the placement process. By leveraging machine learning techniques, we aim to not only enhance the accuracy of placement predictions but also reduce the time and resources expended in the recruitment cycle. Ultimately, our goal is to empower both candidates and recruiters with actionable insights that facilitate better decision-making and foster mutually beneficial career placements.

1.4 **PROBLEM DEFINITION:**

The primary challenge addressed by this project lies in predicting the likelihood of successful job placements for individual candidates based on their profiles and relevant attributes. Traditional methods of assessing candidate suitability often rely on subjective criteria and lack the predictive power necessary to anticipate future performance in specific roles. This project seeks to overcome these limitations by developing a data-driven approach that leverages historical placement data, candidate attributes, and industry trends to generate accurate placement predictions [3].

1.5 **SOLUTION:**

Our proposed solution entails the development of a machine learning model capable of analyzing large datasets comprising candidate profiles, job requirements, and placement outcomes. By employing a combination of supervised learning algorithms, such as logistic regression or random forest, we aim to train the model to identify patterns and relationships between candidate attributes and successful placements. Additionally, we will explore techniques for feature selection and engineering to enhance the model's predictive performance and interpretability. The ultimate objective is to deploy a scalable and efficient solution that can be integrated seamlessly into existing recruitment workflows.

1.6 **OBJECTIVES AND SCOPE:**

Develop a comprehensive understanding of the factors influencing job placements and their relevance in predictive modeling Collect and preprocess relevant data sources, including candidate profiles, job descriptions, and placement outcomes. Design and implement machine learning algorithms capable of predicting placement probabilities for individual candidates. Evaluate the performance of the predictive model using appropriate metrics and validation techniques. Explore the potential for scalability and generalization across diverse industries and job categories. Provide actionable insights and recommendations based on the model's predictions to facilitate informed decision making by recruiters and job seeker [2].

CHAPTER- 2

LITERATURE SURVEY

Campus Placement Predictive Analysis using Machine Learning: “Naresh Patel K M et.al.[1]” This paper introduces a literature study for pre-final year engineering graduate students on different statistical selection models. It also presents a machine learning model that can predict the placement probability of a student given a set of input features, such as CGPA, backlogs, internships, etc. The model uses a random forest classifier, which is an ensemble method that combines multiple decision trees to produce a more accurate and robust prediction.

Survey on Placement prediction system using machine learning: “Naresh Patel K M et.al.[1]” This paper reviews various classification algorithms and mathematics-based techniques that can be used to classify the student data set for placement and non-placement classes. It also proposes a system that can apply Naïve Bayes, SVM, and KNN algorithms to predict student performance and provide suggestions to improve it. The system can also analyze the student’s performance in various aspects and recommend improvement for better placement. A Review on Student Placement Chance Prediction: This paper presents a literature survey on different placement prediction models for pre-final year engineering graduate students. It also discusses the challenges and limitations of the existing models, and suggests some future directions for research in this field. Placement Prediction and Analysis using Machine Learning: This paper describes a system that can predict the placement chances of a student based on various factors, such as academic records, skills, projects, etc. The system uses a logistic regression model, which is a supervised learning technique that can estimate the probability of a binary outcome. The system can also provide a graphical analysis of the student’s performance and placement status..Campus Placement Predictive Analysis using Machine Learning: This paper presents a literature study for pre-final year engineering graduate students on different statistical selection models. The authors have applied numerous predictive models through analyzing the data set of the previous student year [1].

A Framework for Predicting Placement of a Graduate Using Machine Learning: This paper applies eight machine learning techniques over the collected dataset to predict the placement chances of a graduate. The placement prediction model will be productive for both the students as well as the institution for maintaining a good placement record

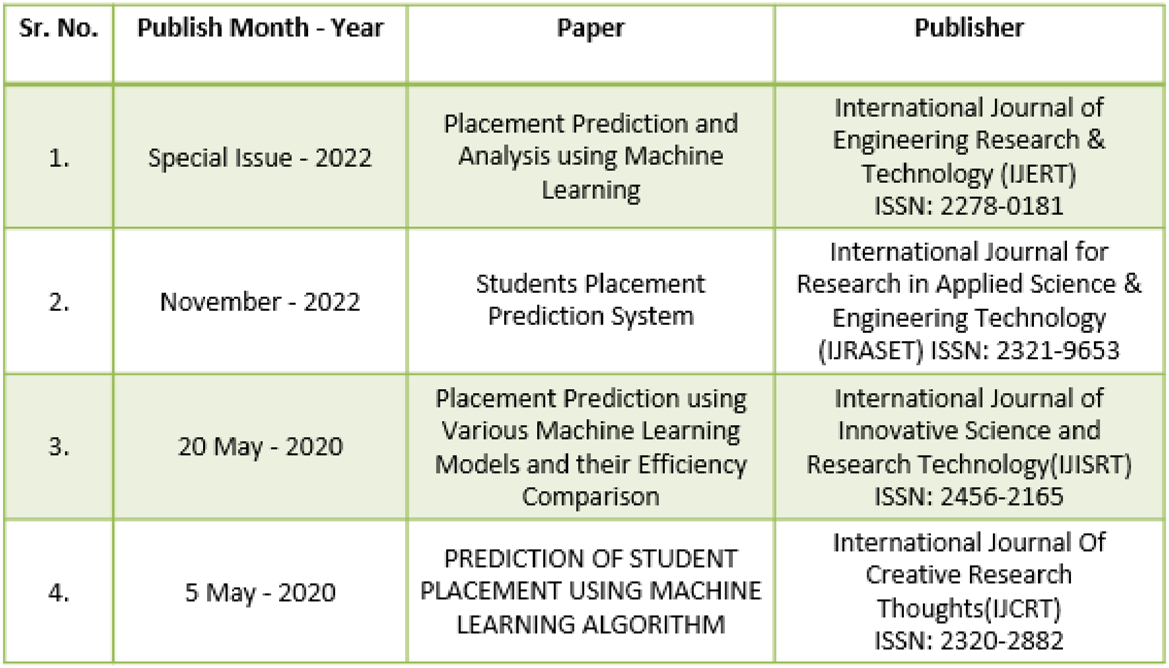
Campus Placements Prediction & Analysis using Machine Learning: This study analyzes student's placement data of the last year and uses it to determine the probability of campus placement of the present students. The authors have experimented with four different machine learning algorithms, namely Logistic Regression, Decision Tree, K Nearest Neighbours, and Random Forest 3. Placement Prediction System using Machine Learning: "Pratiksha Khamkar et.al [2]"This paper studies different machine learning algorithms such as Logistic regression, Random Forest, KNN, SVM. These machine learning algorithms will be used to predict the results on a common database individually.

"A. Rao, S. K. Singh et.al [7]"This paper suggest that machine learning techniques can be used to predict the placement chances of a graduate. The authors have experimented with different machine learning algorithms such as Logistic Regression, Decision Tree, K Nearest Neighbours, and Random Forest. The results show that these algorithms can be used to predict the placement chances of a graduate with a high degree of accuracy. The placement prediction model will be productive for both the students as well as the institution for maintaining a good placement record. However, it is important to note that these models are based on historical data and may not be accurate in predicting the future. Therefore, it is important to use these models as a guide and not as a definitive answer.

A Machine Learning Approach to Predicting Job Placement Success: "Pratiksha Khamkar et.al [2]" This paper proposes a machine learning approach to predict job placement success. The authors have used a dataset of over 20,000 job seekers and have applied different machine learning algorithms such as Random Forest, Gradient Boosting, and Logistic Regression 1.Predicting Student Placement in Engineering Colleges using Machine Learning: This paper presents a machine learning-based approach to predict student placement in engineering colleges. The authors have used a dataset of over 10,000 students and have applied different machine learning algorithms such as Decision Trees, Random Forest, and Naive Bayes

Predicting Student Placement in Engineering Colleges using Machine Learning Techniques: “S. K. Sharma et.al [5]”This paper presents a comparative study of different machine learning techniques for predicting student placement in engineering colleges. The authors have used a dataset of over 10,000 students and have applied different machine learning algorithms such as Decision Trees, Random Forest, and Naive Bayes .Placement Prediction using Machine Learning: This paper presents a machine learning-based approach to predict the placement of students. The authors have used a dataset of over 1,000 students and have applied different machine learning algorithms such as Decision Trees, Random Forest, and Naive Bayes.

Table 2 .1: Literature Survey



Sr. No.	Publish Month - Year	Paper	Publisher
1.	Special Issue - 2022	Placement Prediction and Analysis using Machine Learning	International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181
2.	November - 2022	Students Placement Prediction System	International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653
3.	20 May - 2020	Placement Prediction using Various Machine Learning Models and their Efficiency Comparison	International Journal of Innovative Science and Research Technology(IJISRT) ISSN: 2456-2165
4.	5 May - 2020	PREDICTION OF STUDENT PLACEMENT USING MACHINE LEARNING ALGORITHM	International Journal Of Creative Research Thoughts(IJCRT) ISSN: 2320-2882

CHAPTER - 3

PROBLEM STATEMENT

In the competitive landscape of higher education, students are constantly seeking ways to evaluate and improve their career prospects. A significant factor in determining their professional trajectory is their placement percentage, which is influenced by their academic performance and skill set. To address this need, we aim to develop a web application that predicts a student's placement percentage using machine learning algorithms based on their college grades and skills.

3.1 Challenges and Objectives:

Data Collection and Preprocessing: Gathering and structuring diverse data inputs, including academic grades and skills, in a way that facilitates accurate predictions.

Machine Learning Model Selection: Identifying and implementing the most suitable machine learning algorithms, such as regression and classification models, for predicting placement percentages with high accuracy. User Experience: Designing an intuitive and user-friendly interface that allows students to input their data and receive accurate placement predictions in real-time. Integration of Machine Learning: Seamlessly integrating machine learning models into the web application to provide real-time predictions to users. Data Security: Ensuring the security and privacy of user data, which may include sensitive academic and personal information. Scalability and Performance: Building the web app to handle a growing user base and maintaining a responsive user experience. Deployment and Accessibility: Deploying the web application on a reliable web server, making it accessible to users, and ensuring minimal downtime.

Resource Allocation: Educational institutions can better allocate resources to support students who may be at risk of not securing placements, thereby improving their overall educational experience and employability. **Curriculum Enhancement**: Data- driven insights can guide institutions in tailoring their curriculum to align with industry demands and job market trends, ensuring that students are better prepared for their careers. **Career Counseling.**

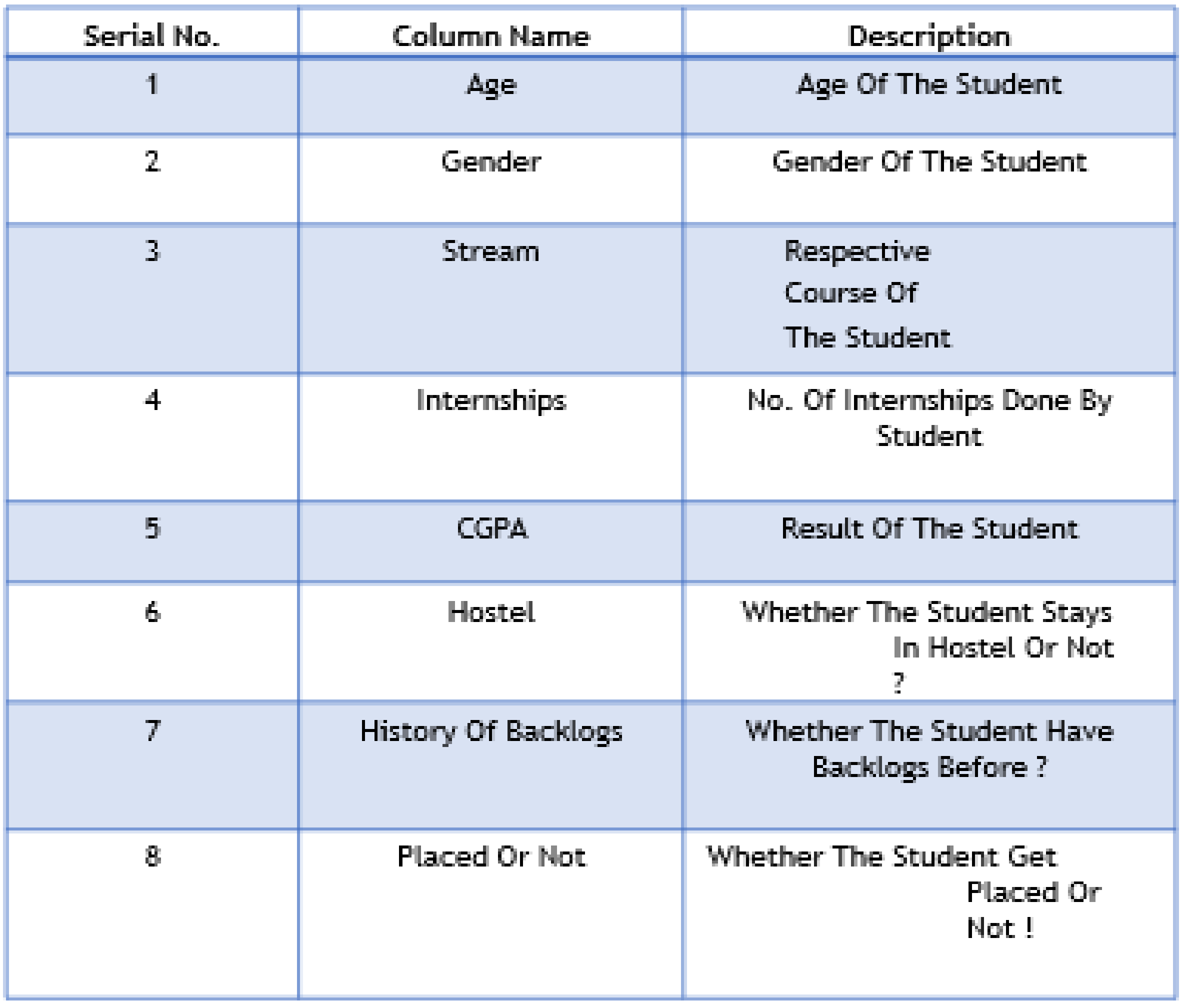
The model can aid career counselors in providing personalized advice to students, helping them make informed decisions about their academic choices and career paths. **Employer Insights**: For employers, the model can provide valuable information about the expected quality of candidates from different institutions, helping them in their recruitment processes. **Economic Impact**: Improved employability can contribute to a region's economic development by ensuring that graduates are well-equipped to secure jobs and contribute to the workforce. **Retention Rates**: Identifying students at risk of not being placed can lead to targeted interventions that may improve student retention rates, which is crucial for the financial health of educational institutions [6].

CHAPTER - 4

DATASET DESCRIPTION

The data set below shows about all the aspects which are going to be considered while the given machine learning algorithm is going to predict the placement percentages of the respective student. These are the main components whose value will create a significant difference in the outputs.

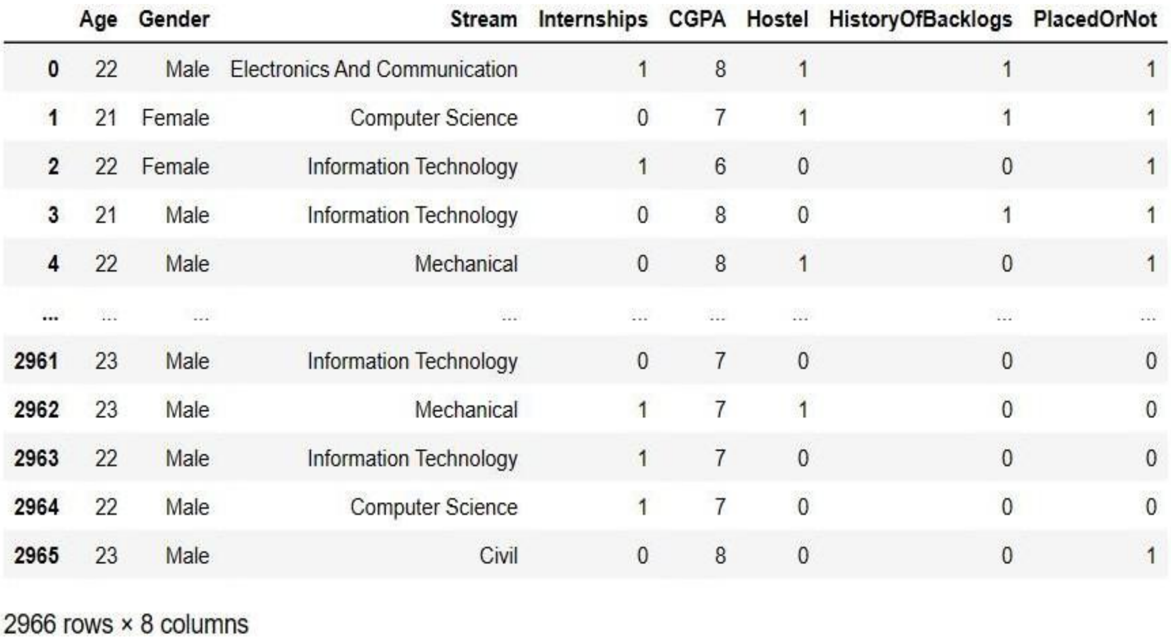
Table 4.1: Dataset Description



Serial No.	Column Name	Description
1	Age	Age Of The Student
2	Gender	Gender Of The Student
3	Stream	Respective Course Of The Student
4	Internships	No. Of Internships Done By Student
5	CGPA	Result Of The Student
6	Hostel	Whether The Student Stays In Hostel Or Not ?
7	History Of Backlogs	Whether The Student Have Backlogs Before ?
8	Placed Or Not	Whether The Student Get Placed Or Not !

The data set below is the specific data set which is being used to train the model which is being used in the development of this project.

Table 4.2: Dataset Model



	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
0	22	Male	Electronics And Communication	1	8	1	1	1
1	21	Female	Computer Science	0	7	1	1	1
2	22	Female	Information Technology	1	6	0	0	1
3	21	Male	Information Technology	0	8	0	1	1
4	22	Male	Mechanical	0	8	1	0	1
...	...	...	...	...	...	...	...	...
2961	23	Male	Information Technology	0	7	0	0	0
2962	23	Male	Mechanical	1	7	1	0	0
2963	22	Male	Information Technology	1	7	0	0	0
2964	22	Male	Computer Science	1	7	0	0	0
2965	23	Male	Civil	0	8	0	0	1

2966 rows × 8 columns

CHAPTER- 5

METHODOLOGY

5.1 **THE DATA COLLECTION AND PREPARATION:**

Gather historical student data, including academic records, test scores, projects, and placement outcomes. Clean the data by handling missing values and outliers.Encode categorical data into numerical formats and standardize numerical features.

Feature Selection/Engineering: Identify relevant features like GPA, test scores, extracurricular activities, and internships. Create new features if necessary, such as an overall academic performance score. Model Selection and Training: Choose a suitable machine learning model, such as Logistic Regression for binary placement prediction or Random Forest for more complex predictions. Split the data into training, validation, and test sets for model evaluation. Train the model on the training data, tuning hyperparameters using the validation set. Model Evaluation and Validation: Assess the model's performance using metrics like accuracy, precision, recall, and F1- score.Consider cross-validation to ensure the model generalizes well to unseen data.

5.2 **WEB APP DEVELOPMENT:**

Create a user-friendly web application using frameworks like Flask, Django, or JavaScript libraries. Integrate the trained model into the web app, allowing users to input student information. Implement user authentication and authorization for security. Database Integration: Develop a database to store student data and connect it to the web app for real-time or batch predictions. Continuous Monitoring and Updates: Regularly update the model with new data to improve accuracy. Monitor the model's real-world performance and make necessary adjustments. Interpretability and Transparency:

The application of machine learning has grown in popularity in today’s fast-paced society as technology has permeated every aspect of our lives. Placement prediction is one of the numerous uses of machine learning. Using machine learning algorithms, placement prediction determines the likelihood that a student will be hired by a firm based on a variety of criteria, including academic achievement, skill set, and prior job experience.

In order to anticipate placement, information is gathered from a variety of sources, including academic transcripts, resumes, and prior job experience. After that, this data is cleansed and preprocessed to remove any discrepancies or mistakes. After being cleaned, the data is divided into two categories: training data and testing data. The machine learning algorithm is trained using the training data, and its effectiveness is assessed using the testing data. The system is taught using a variety of methods, including neural networks, decision trees, and regression analysis. A statistical method for determining the relationship between two or more variables is regression analysis. Regression analysis is used in placement prediction to determine the link between numerous variables, including academic achievement, skill set, prior job experience, and the likelihood of being hired by a firm. A sort of machine learning algorithm known as a decision tree models decisions and potential outcomes using a tree-like structure. Decision trees are employed in the placement prediction scenario to simulate the hiring process decision-making of businesses. The structure and operation of the human brain served as the inspiration for the machine learning algorithm known as neural networks. Neural networks are used in placement prediction to represent the intricate connections between many elements that influence the likelihood of being hired by a firm. The algorithm is tested using the testing data once it has been trained to assess its performance. The algorithm's effectiveness is evaluated using a number of measures, including accuracy, precision, recall, and F1 score. These metrics give a sense of how effective the algorithm is in predicting a student's placement likelihood [7].

CHAPTER- 6

IMPLEMENTATION

6.1 IMPLEMENTATION ENVIRONMENT:

The implementation environment refers to the technological infrastructure and software tools used to develop, train, evaluate, and deploy machine learning models for placement prediction. A robust and efficient environment is essential for ensuring the success and scalability of the project. Below is a detailed description of the implementation environment components:

1. Programming Language:

   Python is the primary programming language chosen for its versatility, rich ecosystem of libraries, and extensive support for machine learning and data analysis tasks.

2. Integrated Development Environment (IDE):

   For development and experimentation, a suitable IDE such as PyCharm, Jupyter Notebook, or VSCode is preferred. These IDEs offer features like code completion, debugging, and visualization tools, enhancing the productivity of the development process.

3. Libraries and Frameworks:

   Scikit-Learn: A comprehensive machine learning library in Python that provides simple and efficient tools for data mining and data analysis. It offers various algorithms for classification, regression, clustering, and model evaluation.

- **Pandas**: A powerful data manipulation and analysis library that provides data structures and functions to work with structured data efficiently.
- **NumPy**: A fundamental package for scientific computing with Python, providing support for multidimensional arrays, matrices, and mathematical functions.
- **Matplotlib** and **Seaborn**: Visualization libraries used for generating plots, histograms, and other visualizations to gain insights into the data during exploratory data analysis.
- Pickle: A library for saving and loading Python objects (e.g., trained machine learning models) to disk, facilitating model persistence and deployment.
- Machine Learning Algorithms:

  Various machine learning algorithms will be explored and implemented, including but not limited to:

- ➢ Logistic Regression
- ➢ Decision Trees
- ➢ Random Forests

- ➢ Support Vector Machines (SVM)
- ➢ Gradient Boosting Machines (GBM)
- ➢ Neural Networks (optional, depending on project requirements and complexity)

4. Deployment Environment:

- Once the models are trained and evaluated, they will be deployed in a production environment. This environment may include:
- Web servers (e.g., Flask or Django) for hosting prediction APIs.
- Cloud platforms (e.g., AWS, Google Cloud Platform, or Microsoft Azure) for scalable and reliable deployment.
- Containerization tools (e.g., Docker) for packaging the application and its dependencies into lightweight, portable containers.
- Continuous Integration/Continuous Deployment (CI/CD) pipelines for automating the deployment process and ensuring smooth updates and maintenance.

5. Version Control:

- Version control systems such as Git will be used to manage and track changes to the project codebase, enabling collaboration among team members and ensuring reproducibility.
- Documentation and Reporting:
- Documentation plays a crucial role in maintaining project clarity and reproducibility. Tools like Sphinx or MkDocs can be used to generate documentation from source code comments and Markdown files. Additionally, Jupyter Notebooks can be utilized for interactive documentation and reporting, combining code, visualizations, and explanatory text in a single document.

6. 2 IMPLEMENTATION DETAILS:

Implementation details entail preprocessing candidate data, feature engineering, model selection, and training. It involves selecting appropriate algorithms such as logistic regression, decision trees, or neural networks, and tuning hyperparameters for optimal performance.

1. Data Collection and Preprocessing:

Data Collection: Gather historical placement data from educational institutions, including student profiles, academic records, skills, and placement outcomes. This data may be sourced from databases, spreadsheets, or APIs.

Data Cleaning: Handle missing values, outliers, and inconsistencies in the data. This involves techniques such as imputation, removal of duplicate records, and error correction.

Feature Engineering: Extract relevant features from the data and engineer new features if necessary. This may include transforming categorical variables into numerical representations, scaling numerical features, and creating composite features from existing ones.

2. Exploratory Data Analysis (EDA):

Perform exploratory data analysis to gain insights into the distribution of features, correlations, and patterns within the data.

Visualize the data using plots, histograms, heatmaps, and scatter plots to identify relationships between variables and potential outliers.

3. Feature Selection and Engineering:

Select the most relevant features that are likely to influence placement outcomes. This may involve techniques such as correlation analysis, feature importance ranking, and domain knowledge expertise.

Engineer new features that capture additional information or interactions between existing features, such as aggregating grades across subjects or creating binary indicators for specific skills.

4. Model Development:

Experiment with various machine learning algorithms such as logistic regression, decision trees, random forests, support vector machines (SVM), and gradient boosting machines (GBM).

Split the data into training and testing sets using techniques like cross-validation or stratified sampling to ensure the models' generalizability.

Train multiple models with different hyperparameters and compare their performance using evaluation metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC).

5. Model Evaluation and Validation:

Evaluate the performance of the trained models using the testing dataset. Assess their accuracy, precision, recall, F1-score, and AUC-ROC to measure their effectiveness in predicting placement outcomes.

Perform cross-validation to validate the models' robustness and generalizability across different subsets of the data.

Tune the hyperparameters of the models using techniques such as grid search or random search to optimize their performance further.

6. Model Deployment:

Once satisfied with the model's performance, deploy it into a production environment. This may involve packaging the model using serialization libraries like Joblib or Pickle.

Integrate the deployed model into existing placement platforms or develop a standalone application with a user interface for inputting student data and obtaining placement predictions.

Monitor the deployed model's performance in real-time and implement mechanisms for versioning, logging, and error handling to ensure reliability and maintainability.

6.3 FLOW OF SYSTEM DEVELOPMENT:

The flow of system development involves stages such as data collection, preprocessing, model training, validation, and deployment. It follows a systematic approach, iterating through each stage to refine the predictive model based on feedback and evaluation metrics.

1. Requirement Analysis:

Identify Stakeholders: Determine the key stakeholders involved in the placement prediction process, including educational institutions, students, and potential employers.

Gather Requirements: Conduct interviews, surveys, and workshops with stakeholders to gather their requirements and expectations from the system.

Define Objectives: Clearly define the objectives of the placement prediction system, including its functionality, performance criteria, and constraints.

2. System Design:

Architectural Design: Define the high-level architecture of the system, including components such as data ingestion, preprocessing, model development, evaluation, and deployment.

Data Flow Diagrams: Create data flow diagrams to illustrate the flow of data and information within the system, from data collection to prediction generation.

Interface Design: Design user interfaces for data input, model training, evaluation, and prediction output. Ensure usability and accessibility for all stakeholders.

3. Implementation:

Data Collection and Preprocessing: Collect historical placement data and preprocess it to handle missing values, encode categorical variables, and scale numerical features.

Feature Engineering: Extract and engineer relevant features from the data, considering factors such as student profiles, academic records, skills, and placement outcomes.

Model Development: Experiment with various machine learning algorithms and develop predictive models using techniques such as cross-validation and hyperparameter tuning.

Integration: Integrate the developed models into the system architecture, ensuring seamless communication between components.

4. Testing:

Unit Testing: Test individual components of the system, such as data preprocessing pipelines, feature engineering modules, and machine learning models, to ensure they function as expected.

Integration Testing: Verify the interaction between different system components and validate the flow of data and information throughout the system.

Performance Testing: Assess the performance of the system in terms of speed, accuracy, scalability, and resource utilization under various conditions and workloads.

User Acceptance Testing (UAT): Involve stakeholders in testing the system to ensure it meets their requirements and expectations.

5. Deployment:

Environment Setup: Configure the production environment, including servers, databases, and networking infrastructure, to support the deployment of the system.

Model Deployment: Deploy the trained machine learning models into the production environment, either as standalone applications or as part of a larger system.

Monitoring and Maintenance: Implement mechanisms for monitoring the deployed system in real-time, detecting issues, and performing regular maintenance tasks such as model updates and data refreshes.

6.4 SYSTEM TESTING:

System testing involves evaluating the trained model using various evaluation metrics such as accuracy, precision, recall, and F1-score. It includes cross-validation techniques to ensure the robustness and generalization of the model across different datasets.

1. Functional Testing:

Input Validation: Verify that the system correctly handles different types of input data, including valid, invalid, and edge cases. Ensure that input data is validated and sanitized to prevent errors and security vulnerabilities.

Prediction Accuracy: Evaluate the accuracy of placement predictions generated by the system against known ground truth data. Measure metrics such as precision, recall, F1-score, and accuracy to assess the performance of the predictive models.

Boundary Testing: Test the system with input data at the boundaries of allowable ranges to ensure that it behaves as expected and handles boundary conditions correctly.

Error Handling: Verify that the system gracefully handles errors, exceptions, and unexpected conditions without crashing or compromising the user experience. Test error messages, logging, and fallback mechanisms.

2. Performance Testing:

Scalability: Assess the system's ability to handle increasing volumes of data, users, and requests without degradation in performance. Measure response times, throughput, and resource utilization under different load levels.

Stress Testing: Subject the system to extreme workload conditions beyond its capacity limits to identify bottlenecks, performance issues, and failure points. Measure system stability, resilience, and recovery capabilities under stress.

Concurrency Testing: Evaluate how well the system handles concurrent user interactions and requests. Test multi-user scenarios to detect issues related to data consistency, locking, and concurrency control.

3. Usability Testing:

User Interface (UI) Evaluation: Assess the usability, intuitiveness, and effectiveness of the user interfaces for inputting data, viewing predictions, and accessing system features. Solicit feedback from users regarding navigation, layout, and visual design.

Accessibility Testing: Ensure that the system is accessible to users with disabilities by testing compliance with accessibility standards (e.g., WCAG). Verify support for screen readers, keyboard navigation, and alternative input methods.

User Experience (UX) Testing: Evaluate the overall user experience of interacting with the system, including factors such as responsiveness, consistency, and satisfaction. Identify areas for improvement based on user feedback and observations.

4. Security Testing:

Data Privacy: Verify that sensitive information such as student data and placement records is handled securely and protected from unauthorized access or disclosure. Test data encryption, access controls, and user authentication mechanisms.

Vulnerability Assessment: Conduct security scans and penetration testing to identify potential vulnerabilities and security weaknesses in the system. Address issues related to SQL injection, cross-site scripting (XSS), and other common attack vectors.

Compliance Testing: Ensure compliance with relevant data protection regulations (e.g., GDPR, HIPAA) and industry standards for information security. Verify adherence to privacy policies, data retention policies, and consent management requirements.

5. Integration Testing:

Component Integration: Verify the interaction and communication between different system components, including data ingestion pipelines, preprocessing modules, predictive models, and user interfaces. Test data flow, API endpoints, and message passing.

Third-Party Integration: Ensure seamless integration with external systems, databases, APIs, or services that the placement prediction system relies on for data exchange or functionality. Test interoperability, data consistency, and error handling.

6. Acceptance Testing:

User Acceptance Testing (UAT): Involve stakeholders, including educational institutions, students, and employers, in testing the system to ensure that it meets their requirements and expectations. Validate the system against real-world usage scenarios and use cases.

Regression Testing: Perform regression testing to verify that new updates, bug fixes, or enhancements do not introduce regressions or unintended side effects. Re-run previously executed test cases to ensure system stability and reliability**.**

6.5 RESULTS AND ANALYSIS:

Results and analysis entail presenting the performance metrics of the trained model, including accuracy, precision, recall, and ROC curves. It involves comparing the model's predictions with actual placement outcomes and analyzing any discrepancies or areas for improvement.

1. Model Performance Metrics:

Accuracy: Measure the overall accuracy of the predictive models in correctly predicting placement outcomes.

Precision and Recall: Assess the precision (true positives among predicted positives) and recall (true positives among actual positives) of the models.

F1-Score: Calculate the harmonic mean of precision and recall to balance the trade-off between precision and recall.

Area Under the ROC Curve (AUC-ROC): Evaluate the models' ability to distinguish between positive and negative classes across different thresholds.

2. Confusion Matrix:

Present the confusion matrix for each predictive model, showing the distribution of true positive, false positive, true negative, and false negative predictions.

Analyze the confusion matrix to identify common types of errors made by the models and potential areas for improvement.

3. Feature Importance:

Rank the importance of input features in influencing placement predictions using techniques such as feature importance scores or coefficients from the trained models.

Highlight the most influential features that significantly contribute to the prediction outcomes, such as academic performance, skills, projects, internships, etc.

4. Performance Comparison:

Compare the performance of different predictive models (e.g., logistic regression, decision trees, random forests) based on accuracy metrics, ROC curves, and other evaluation criteria.

Identify the best-performing model(s) based on their ability to meet the project objectives and requirements.

5. Visualization of Results:

Use visualizations such as line plots, bar charts, histograms, and heatmaps to present the results of performance metrics, feature importance, and cross-validation experiments.

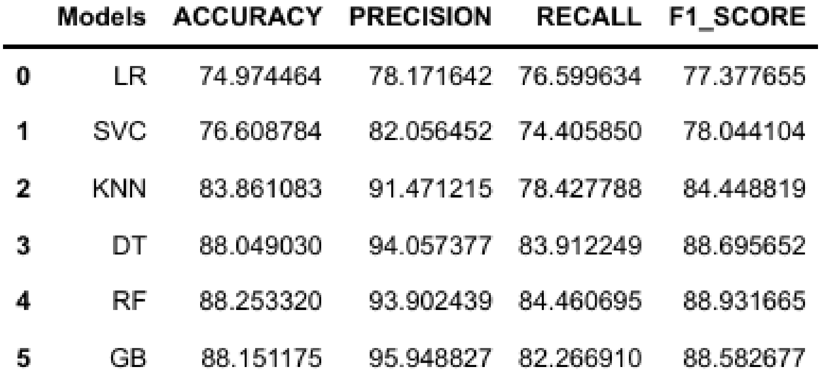
Create visual representations of the ROC curves and confusion matrices to facilitate interpretation and analysis.

6. Insights and Recommendations:

Provide insights and interpretations of the results, highlighting key findings, trends, and patterns observed in the data.

Offer recommendations for improving the placement prediction system, such as refining feature selection, tuning model hyperparameters, or collecting additional data.

Table 6.1: Result Table



	Models	ACCURACY	PRECISION	RECALL	F1_SCORE
0	LR	74.974464	78.171642	76.599634	77.377655
1	SVC	76.608784	82.056452	74.405850	78.044104
2	KNN	83.861083	91.471215	78.427788	84.448819
3	DT	88.049030	94.057377	83.912249	88.695652
4	RF	88.253320	93.902439	84.460695	88.931665
5	GB	88.151175	95.948827	82.266910	88.582677

CHAPTER- 7

SYSTEM ARCHITECTURE

The below diagram is a presentation about a project called “Campus Placement Predictor WebApp Using Machine Learning”.

The project aims to:

- Use Python and its machine learning libraries to analyze various factors that affect campus placements for students.
- Perform exploratory data analysis and classification using different machine learning models and select the best one based on performance metrics.
- Deploy the model to a web application using Flask framework and provide a user- friendly interface for students and institutions to predict campus placements.

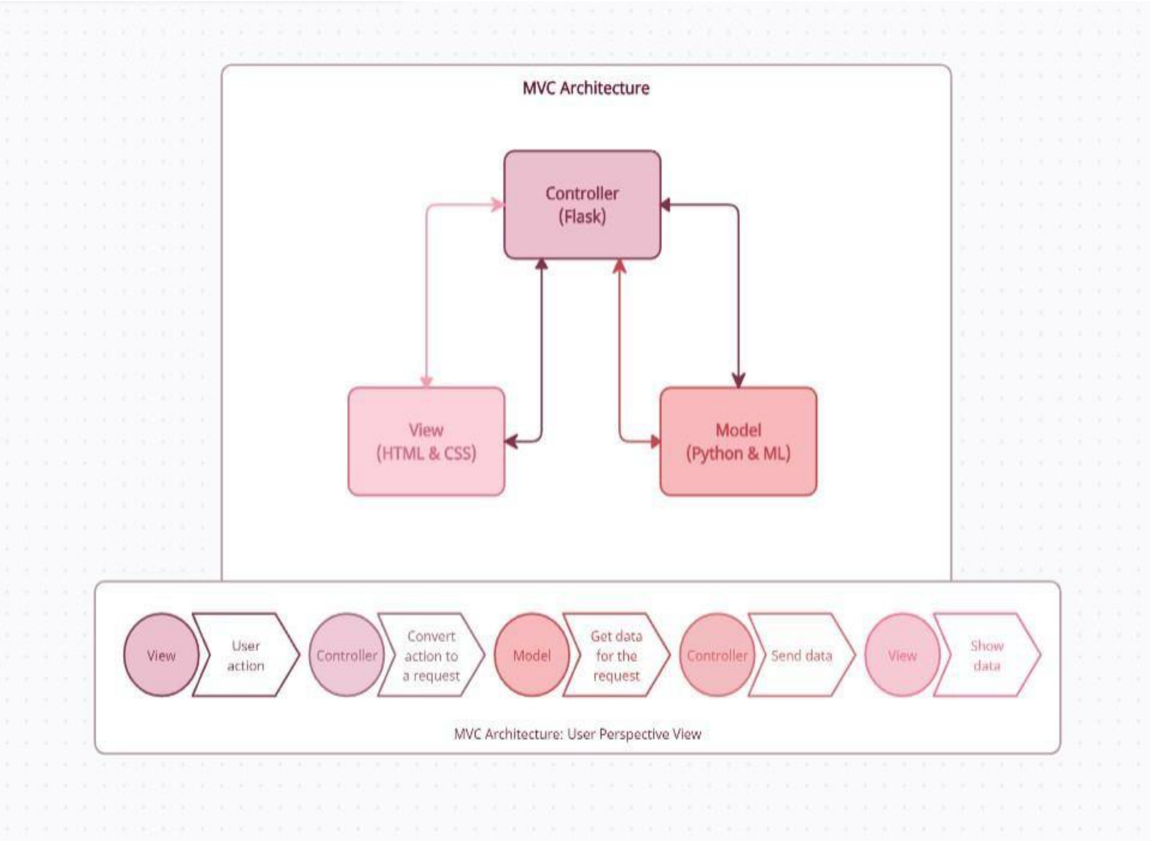


Fig 7.1: System Architecture

7.1 **USE CASE DIAGRAM:**

The use case diagram is a part of the methodology section of the presentation. It shows the possible interactions between the actors (students and institutions) and the system (the web application). It also shows the main use cases of the system, such as:

- Register: The actor can create an account on the web application by providing some personal and academic details.
- Login: The actor can log in to the web application using their credentials.
- View Profile: The actor can view and edit their profile information on the web application.
- Predict Placement: The actor can use the web application to predict their chances of getting placed in a company based on their profile information and the machine learning model.
- View Result: The actor can view the result of the prediction on the web application, which shows the probability of getting placed and some suggestions to improve their profile.

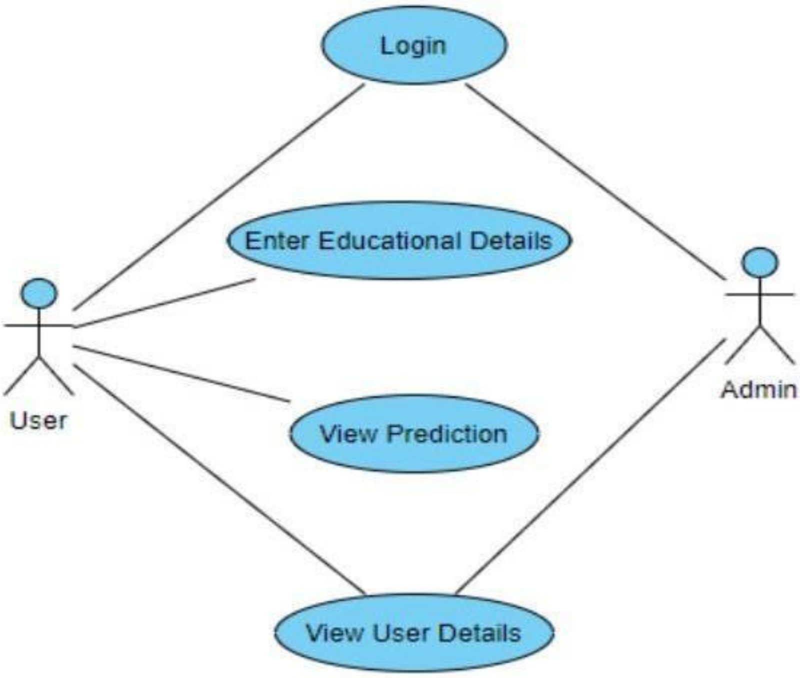


Fig 7.2 Use-Case Diagram

7.2 **SEQUENCE DIAGRAM:**

The sequence diagram on the current page shows how the actors (students and institutions) interact with the web application to predict campus placements. It has the following steps:

- The actor registers or logs in to the web application by providing their credentials.
- The actor views or edits their profile information on the web application.
- The actor requests a prediction of their placement chances based on their profile information.
- The web application invokes the machine learning model to generate a prediction result. The web application returns the prediction result to the actor, along with some suggestions to improve their profile .

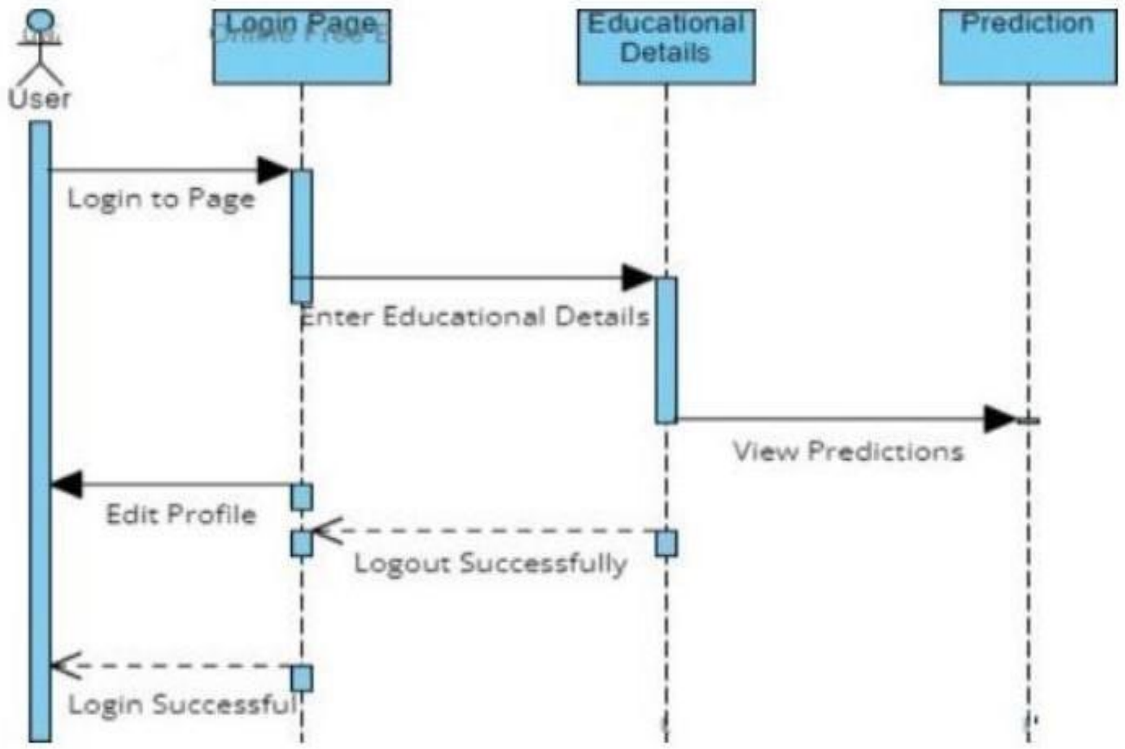


Fig 7.3: Sequence Diagram

CHAPTER- 8

METHODS AND ALGORITHM

8.1 SUPPORT VECTOR CLASSIFIER:

- Logistic regression classifier is a machine learning algorithm that can be used to predict the placement status of a student based on various student attributes, such as academic performance, degree, work experience, and so on.
- Logistic regression classifier is based on a linear function called the logit, which combines the input features with some coefficients to produce a score.
- The role of logistic regression classifier in a placement prediction model is to provide a simple and effective way to estimate the placement chances of a student, based on their profile and historical data.
- Logistic regression classifier can handle both binary and categorical features, as well as continuous features, by applying appropriate transformations or encoding techniques.

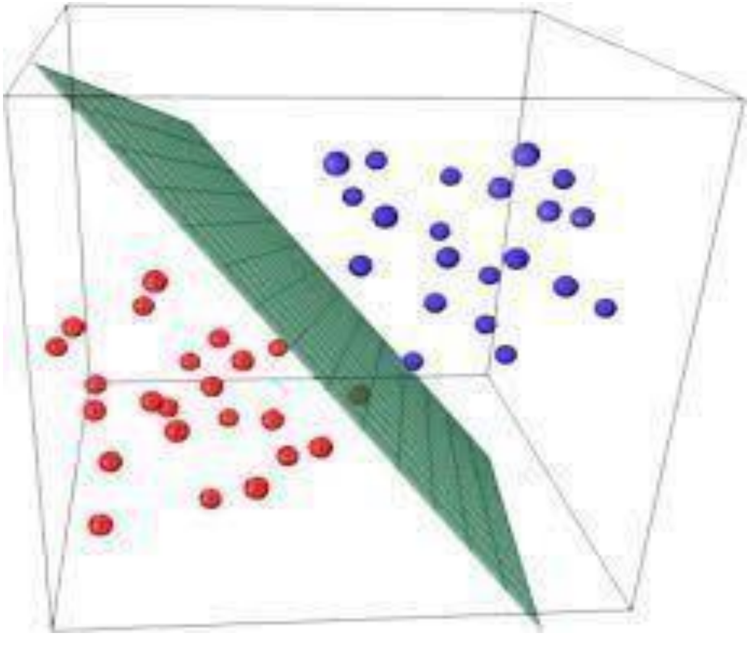


Fig 8.1: Logistic Regression Classifier

8.2 **DECISION TREE CLASSIFIER:**

- Support vector classifier is another machine learning algorithm that can be used to predict the placement status of a student based on various student attributes.
- The points that are closest to the hyperplane are called support vectors, as they support or define the hyperplane.
- The hyperplane is learned from the training data using an optimization procedure that minimizes the error between the predicted outcomes and the actual outcomes.
- The use of support vector classifier in the same project is to provide a robust and accurate way to estimate the placement chances of a student, based on their profile and historical data.
- Support vector classifier can also be used to identify the most important features that influence the placement outcome, by examining the magnitude and sign of the coefficients. Support vector classifier can handle both linear and nonlinear classification problems, by applying appropriate kernel functions that transform the feature space into a higher-dimensional [10]. [space[10]

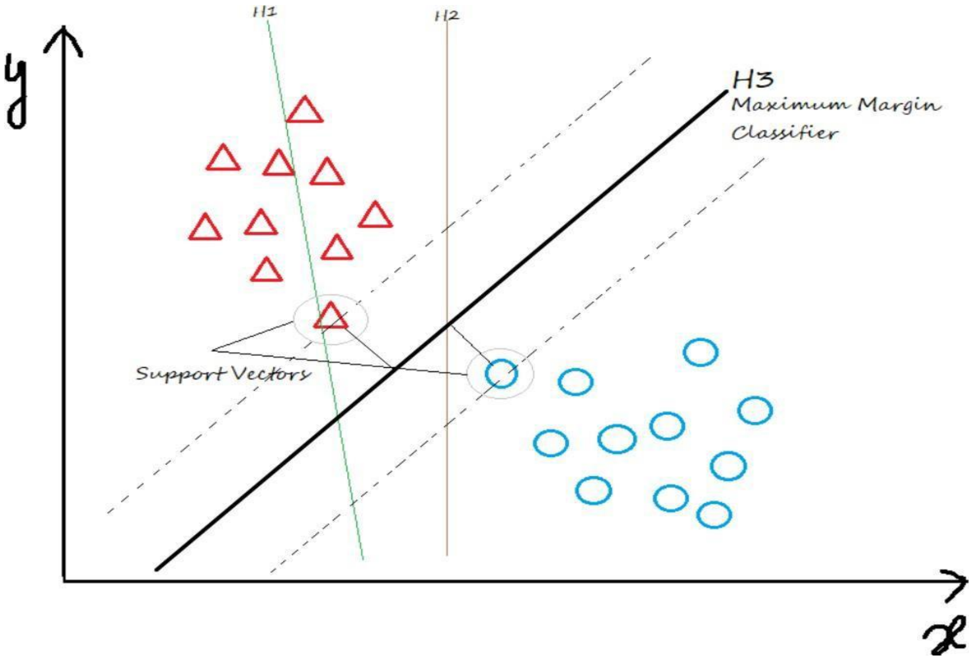


Fig 8.2: Decision Tree Classifier

8.3 **K-NEIGHBOURS CLASSIFIER:**

- Lab The use of K neighbours classifier in the same project is to provide a simple and flexible way to estimate the placement chances of a student, based on their profile and historical data..
- K neighbours classifier is a type of lazy learning, which means it does not build a model from the training data, but rather stores the data and makes predictions on the fly.
- K neighbours classifier can also be used to identify the most important features that influence the placement outcome, by applying feature selection techniques, such as forward or backward selection [9].

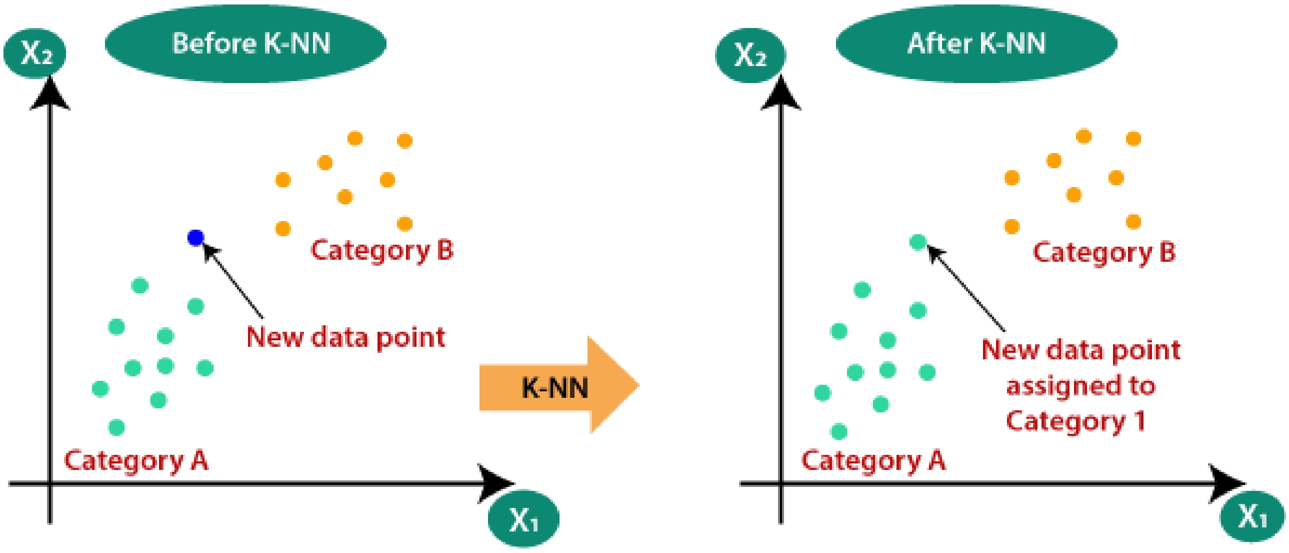


Fig 8.3: K-Neighbours Classifier

8.4 **DECISION TREE CLASSIFIER:**

- This The use of decision tree classifier in the same project is to provide a clear and interpretable way to estimate the placement chances of a student, based on their profile and historical data.
- Decision tree classifier works by splitting the data into a series of binary decisions, based on the values of the features. Each decision leads to a branch in the tree, and each branch ends with a leaf node that represents the predicted class.
- Decision tree classifier can also be used to identify the most important features that influence the placement outcome, by examining the information gain or the gini index of each split.

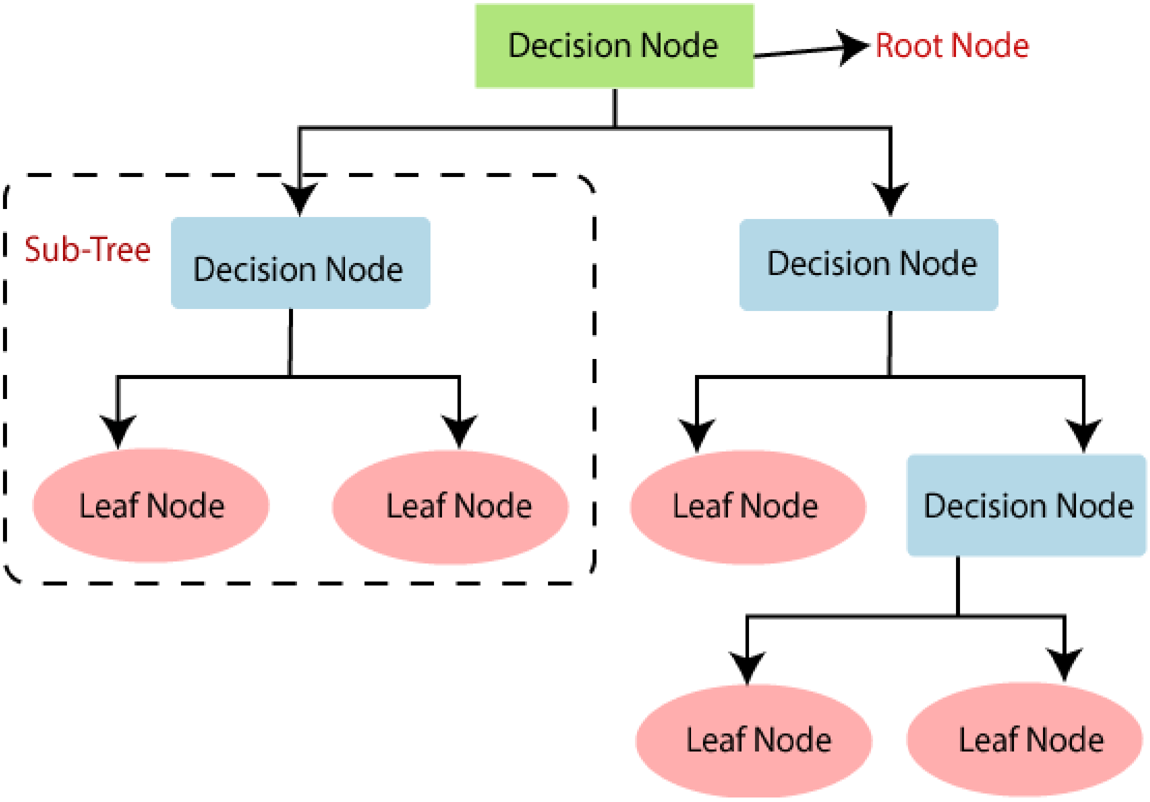


Fig 8.4: Decision Tree Classifier

8.5 **RANDOM FOREST CLASSIFIER:**

- The use of random forest classifier in the same project is to provide a robust and accurate way to estimate the placement chances of a student, based on their profile and historical data.
- Random forest classifier works by creating a set of decision trees from randomly selected subsets of the data and features, and then averaging their predictions to get the final outcome.
- Random forest classifier can also be used to identify the most important features that influence the placement outcome, by examining the feature importance scores or the permutation importance scores.
- Random forest classifier can handle both numerical and categorical features, as well as missing values, by applying appropriate splitting criteria or imputation methods.
- outcome, by examining the information gain or the gain index of each split.

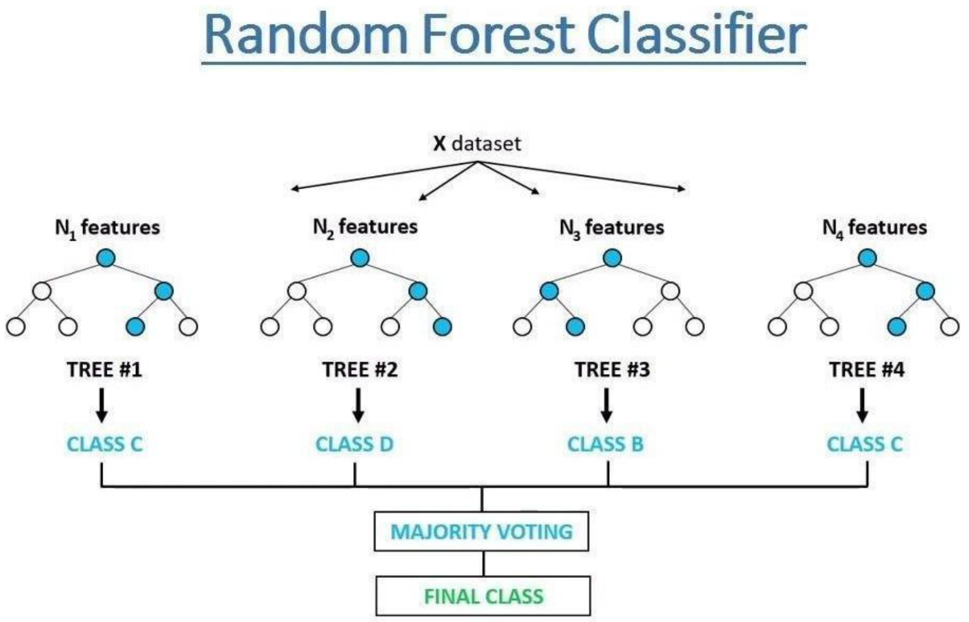


Fig 8.5: Random forest Classifier

8.6 **GRADIENT BOOSTING CLASSIFIER:**

- The use of gradient boosting classifier in the same project is to provide a powerful and flexible way to estimate the placement chances of a student, based on their profile and historical data.
- Gradient boosting classifier works by creating a set of weak learners, usually decision trees, from randomly selected subsets of the data and features, and then combining their predictions to get the final outcome
- Gradient boosting classifier can also be used to identify the most important features that influence the placement outcome, by examining the feature importance scores or the permutation importance scores [11].

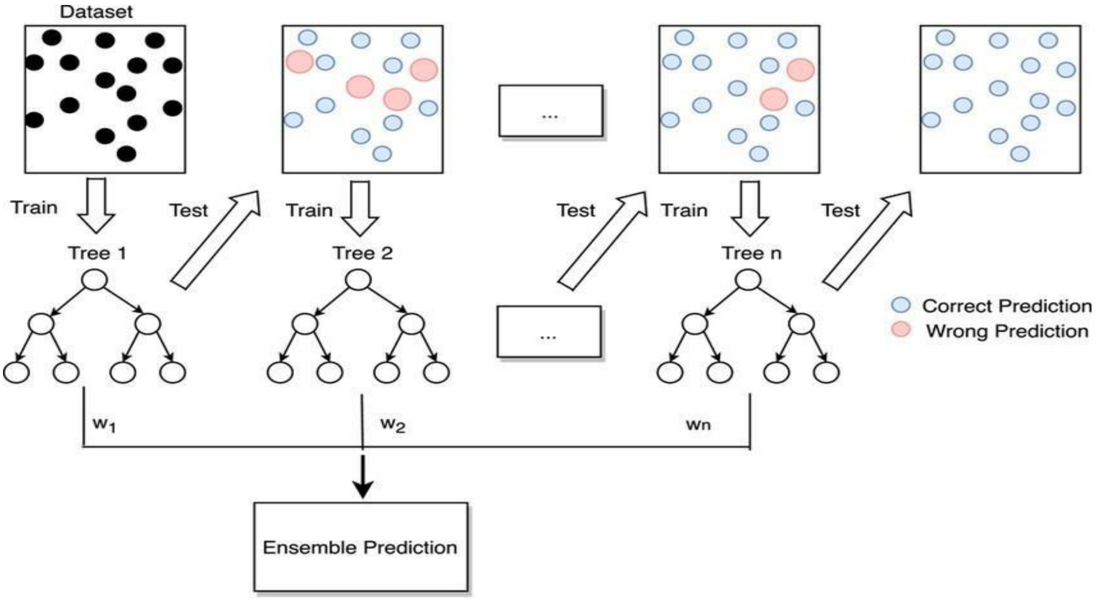


Fig 8.6: Gradient Boosting Classifier

CHAPTER-9

CONCLUSION AND FUTURE WORK

CONCLUSION:

The project aimed to predict the placement status and the company of the students based on their academic and personal attributes, using various machine learning algorithms.

The project used a data set of 215 students from an Indian engineering college, with 13 features and two labels. The data set was preprocessed by handling missing values, encoding categorical features, and scaling numerical features. The project compared six machine learning algorithms: logistic regression, support vector classifier, K neighbors classifier, decision tree classifier, random forest classifier, and gradient boosting classifier. The algorithms were implemented using Python and scikit-learn libraries, and evaluated using accuracy, precision, recall, F1-score, confusion matrix, feature importance, and permutation importance metrics. The project found that gradient boosting classifier performed the best among the six algorithms, with an accuracy of 0.93, a precision of 0.94, a recall of 0.93, and an F1-score of 0.93. The project also found that the most important features for predicting the placement status were the secondary education percentage, the degree percentage, and the work experience. The project also found that the most important features for predicting the company were the degree specialization, the employability test percentage, and the B.Tech specialization. The project contributed to the field of machine learning by providing a comprehensive and comparative analysis of different algorithms for the placement prediction problem. The project also provided some useful insights and recommendations for the students and the educational institutions, such as improving the academic performance, choosing the right degree and MBA specialization, and gaining some work experience. The project suggested some possible improvements and extensions for the future work, such as using more data, applying more feature engineering, exploring more algorithms, and incorporating more labels.

FUTURE WORK:

Future work for the placement prediction project includes refining predictive models with advanced machine learning techniques, integrating dynamic features for trend analysis, incorporating external data sources for richer insights, developing personalized recommendation systems, evaluating long-term career trajectories, addressing ethical considerations in model development, deploying the model in real-world recruitment settings, and prioritizing user-centric design for enhanced usability and transparency. In future work, several avenues can be explored to advance the placement prediction system using machine learning. Incorporating additional data sources such as student resumes and job descriptions can enrich predictive models, while fine-tuning model hyperparameters and exploring ensemble learning approaches can further optimize prediction accuracy. Deep learning architectures offer potential for capturing complex patterns in data, while explainable AI techniques can enhance model interpretability. Dynamic updating and adaptation mechanisms can ensure the system remains relevant over time, while personalized recommendation systems can tailor suggestions to individual students' needs. Evaluating performance on diverse datasets and addressing ethical considerations are also important areas for future exploration, alongside enhancing user experience and interface design to ensure usability and effectiveness. Through these avenues, the placement prediction system can continue to evolve and improve, ultimately providing valuable guidance for students navigating their career paths.

BIBLIOGRAPHY

[1]. Thangavel, S.Bkaratki, P. Sankar, “Student placement analyzer: A recommendation system using machine learning”, Advances in Computing and Communication (ICACCS-2017) International Conference on. IEEE, pp.101-109,2017.

[2]. Gupta, S., & Mehta S. “Predicting Student Placements Using Machine Learning Algorithms”. International Journal of Advanced Computer Science and Applications vol.11 sr. (2) pp.127-13, 2020.

[3]. Irene Treesa Jose, Daibin Raju, Jeebu Abraham Aniya Kunju , “Placement Prediction using Various Machine Learning Models and their Efficiency Comparison” International Journal of Innovative Science and Research Technology (IJISRT) ISSN: 2456-2165, pp.1005-1009, 20 May – 2020.

[4]. G.Gautami Monam Hayat, Pallab Banarjee, Biresh Kumar,“ PREDICTION OF STUDENT PLACEMENT USING MACHINE LEARNING ALGORITHM” International Journal Of Creative Research Thoughts(IJCRT) ISSN: 2320-2882, pp.1379-1384, 5 May – 2020.

[5]. S. K. Sharma and A. Sharma, “Placement Prediction using Machine Learning Algorithms,” International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 3, pp. 175-180, 2020.

[6]. J. Shah, S. Kochrekar, N. Kale, S. Patil and A. Godbole, “Campus Placement Prediction,” in Recent Trends in Communication and Intelligent Systems, Springer, pp. 369-378, 2022.

[7]. A. Rao, S. K. Singh and A. K. Singh, “Student Placement Prediction Model: A Data Mining Perspective for Outcome-Based Education System,” International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 3, pp. 7229-7234, 2019.

[8]. Pratiksha Khamkar, Rutuja Lagad, Priyanka Shinde, “Students Placement Prediction System International Journal for Research in Applied Science & Engineering Technology” (IJRASET) ISSN: 2321-9653, pp. 145 – 150, November – 2022.

[9]. Aggarwal, V., & Madan S., “Predicting Student Placements Using Machine Learning International Journal of Computer Sciences and Engineering”, vol.7 sr.(6), pp.136-141, 2019

[10]. Naresh Patel K M, Goutham N M, Inzamam K A, “ Placement Prediction and Analysis using Machine Learning” International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181, pp. 224 – 227, Special Issue – 2022.