A

Major Project

On

CAMPUS PLACEMENTS PREDICTION AND ANALYSIS USING MACHINE LEARNING

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

V Sindhu Bhargavi (217R1A05R7)

S Sruthi (227R5A0525)

Under the Guidance of

M. MADHUSUDHAN

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

April, 2025.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**CAMPUS PLACEMENTS PREDICTION AND ANALYSIS USING MACHINE LEARNING** " being submitted by **V. Sindhu Bhargavi (217R1A05R7), S. Sruthi (227R5A0525)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, during the year 2024-25.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Mr. M. Madhusudhan
Assistant Professor
INTERNAL GUIDE

Dr. Nuthanakanti Bhaskar
HoD

Dr. A. Raji Reddy
DIRECTOR

Signature of External Examiner

Submitted for viva voice Examination held on ______________________

ACKNOWLEDGEMENT

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project, we take this opportunity to express our profound gratitude and deep regard to our guide **Mr. M. Madhusudhan,** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We take this opportunity to extend our heartfelt appreciation to the Project Review Committee (PRC) Coordinators—**Dr. K. Maheswari, Dr. J. Narasimharao, Ms. K. Shilpa, and Mr. K. Ranjith Reddy**—for their unwavering support, insightful guidance, and valuable inputs, which played a crucial role in steering this project through its various stages.

Our sincere appreciation also goes to **Dr. Nuthanakanti Bhaskar**, Head, for his encouragement and continuous support in ensuring the successful completion of our project.

We are deeply grateful to **Dr. A. Raji Reddy**, Director, for his cooperation throughout the course of this project. Additionally, we extend our profound gratitude to Sri. **Ch. Gopal Reddy**, Chairman, Smt. **C. Vasantha Latha**, Secretary and Sri. **C. Abhinav Reddy**, Vice-Chairman, for fostering an excellent infrastructure and a conducive learning environment that greatly contributed to our progress.

We also acknowledge and appreciate the guidance and assistance provided by the faculty and staff of **CMR Technical Campus**, whose contributions have been invaluable in bringing this project to fruition.

Lastly, we sincerely thank our families for their unwavering support and encouragement. We also extend our gratitude to the teaching and non-teaching staff of CMR Technical Campus for their guidance and assistance. Their contributions, along with the support of everyone who helped directly or indirectly, have been invaluable in the successful completion of this project.

V Sindhu Bhargavi (217R1A05R7)

S Sruthi (227R5A0525)

VISION AND MISSION

INSTITUTE VISION:

To Impart quality education in serene atmosphere thus strive for excellence in Technology and Research.

INSTITUTE MISSION:

1. To create state of art facilities for effective Teaching- Learning Process.
2. Pursue and Disseminate Knowledge based research to meet the needs of Industry & Society.
3. InfuseProfessional, Ethical and Societal values among Learning Community.

DEPARTMENT VISION:

To provide quality education and a conducive learning environment in computer engineering that foster critical thinking, creativity, and practical problem-solving skills.

DEPARTMENT MISSION:

1. To educate the students in fundamental principles of computing and induce the skills needed to solve practical problems.
2. To provide State-of-the-art computing laboratory facilities to promote industry institute interaction to enhance student’s practical knowledge.
3. To inculcate self-learning abilities, team spirit, and professional ethics among the students to serve society.

ABSTRACT

Campus placements are a crucial milestone for students, determining their entry into the professional world. This study focuses on predicting student placements using machine learning algorithms to provide insights into job readiness. We implemented five classification models: Support Vector Machine (SVM), Logistic Regression, Decision Tree Classifier, K-Nearest Neighbors (KNN), and Random Forest Classifier. The dataset includes academic performance, extracurricular activities, and other relevant placement factors. Preprocessing steps such as handling missing values, feature scaling, and normalization were performed to enhance model performance. Each algorithm was evaluated based on accuracy to determine its effectiveness in predicting placements. The results show that SVM achieved **81.2%** accuracy, Logistic Regression attained **78.6%**, Decision Tree Classifier reached **83.1%**, K-Nearest Neighbors recorded **79.4%**, and Random Forest Classifier outperformed all with **86.5%** accuracy. The analysis indicates that ensemble learning models, such as Random Forest, provide better predictive power due to their ability to reduce overfitting and improve generalization. The Decision Tree classifier also performed well, but it was prone to overfitting in certain cases. SVM and KNN showed moderate performance, while Logistic Regression had the lowest accuracy, suggesting it may not be the best choice for placement prediction. These findings highlight the importance of selecting the right algorithm based on dataset characteristics and performance metrics. Institutions can utilize these insights to guide students in improving their placement probabilities. This project provides a data-driven approach to analyzing student employability and optimizing training programs. Future work can include deep learning techniques and additional features to enhance prediction accuracy further.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

The project, titled "Campus Placements Prediction And Analysis Using Machine Learning” Placements are considered to be very important for each and every college. The basic success of the college is measured by the campus placement of the students. Every student takes admission to the colleges by seeing the percentage of placements in the college. Hence, in this regard the approach is about the prediction and analyses for the placement necessity in the colleges that helps to build the colleges as well as students to improve their placements.

In Placement Prediction system predicts the probability of a undergrad students getting placed in a company by applying classification algorithms such as Decision tree and Random forest. The main objective of this model is to predict whether the student he/she gets placed or not in campus recruitment. For this the data consider is the academic history of student like overall percentage, backlogs, credits. The algorithms are applied on the previous years data of the students.

1.1 PROJECT PURPOSE

This The primary purpose of this project is to predict the likelihood of students securing job placements based on various factors and analyze trends that influence campus placements. By using machine learning techniques, the project creates models to assess placement success based on academic performance, skills, and extracurricular activities. It also identifies key factors like academic scores, internship experience, and department that affect placement outcomes. This enables colleges to make data-driven decisions to improve placement rates and helps students understand which areas to focus on for better preparation.

Additionally, the project provides insights for companies to optimize their recruitment strategies by understanding the traits of students most likely to succeed in the industry. With a data-driven approach, the project reduces the manual effort required for placement preparations and offers long-term analysis to track changes in trends over multiple cycles. Ultimately, it benefits students, educational institutions, and recruiting companies by providing a clear understanding of the factors that drive campus placement success.

1.2 PROJECT FEATURES

The Placements Prediction and Analysis project is designed to address the growing importance of campus placements in colleges. Placements are considered one of the most crucial indicators of a college's success and reputation. For students, the placement percentage plays a significant role in choosing a college, as it directly impacts their future career prospects. This project aims to predict and analyze the factors that influence placements, offering insights that can benefit both students and colleges in improving placement outcomes. The ultimate goal is to build a prediction model that helps students and institutions understand the likelihood of a student securing a placement during campus recruitment, based on a variety of academic and personal factors.

The system uses machine learning classification algorithms like Decision Tree and Random Forest to predict the probability of an undergraduate student being placed in a company. The key objective of the model is to predict whether a student will be placed or not during campus recruitment drives. The prediction is based on the student's academic history, including factors like their overall percentage, the number of backlogs, credits earned, and other academic performance metrics. The model uses historical data from previous years of students' placement records to train and apply these algorithms. By analyzing past trends, the system identifies patterns and correlations that can predict future placement outcomes. These insights can be used by colleges to improve their placement strategies, enhance support for students, and guide students in identifying areas they need to improve upon to increase their chances of being placed. The use of decision trees and random forests allows for better classification and prediction accuracy, making the model a valuable tool for optimizing the placement process.

2. LITERATURE SURVEY

2. LITERATURE SURVEY

The A literature survey on campus placement prediction and analysis using machine learning reveals various approaches to improving prediction accuracy and understanding the factors affecting placement success. Several studies have applied machine learning models to predict placement outcomes, with Kumar et al. (2020) using logistic regression and decision trees based on academic data such as CGPA, skills, and interview performance. These studies demonstrated that academic performance and soft skills were crucial in predicting placement success, but their accuracy was limited by the simple algorithms used and a lack of consideration for complex interactions between factors.

Recent research has moved towards using more advanced machine learning techniques to enhance prediction accuracy. Ravi et al. (2021) explored models like Random Forest and Support Vector Machines (SVM), incorporating a broader set of features such as extracurricular activities, internships, and soft skills, which resulted in better performance than traditional methods. However, challenges remained in handling unstructured data and integrating real-time data updates into the prediction models.

In addition to academic research, existing campus placement platforms such as TALENTac, Internshala, and Naukri Campus have also begun to implement predictive models. These platforms mainly focus on connecting students with employers and offer basic recommendation systems, matching students to available job roles. However, these platforms lack the ability to predict placement success based on a comprehensive analysis of factors like academic performance, interview skills, and personality traits. This highlights the need for the application of more advanced machine learning techniques to enhance their predictive capabilities and offer personalized insights for students and institutions.

In conclusion, while existing research and systems have made valuable contributions to campus placement prediction, there are still significant gaps in integrating diverse data sources, using advanced machine learning models, and ensuring real-time adaptability. Future research should focus on addressing these challenges, particularly by incorporating deep learning techniques and real-time data processing to enhance the accuracy and flexibility of placement prediction models.

2.1 REVIEW OF RELATED WORK

Campus placement prediction is a crucial aspect of career planning for students and universities. By leveraging machine learning techniques, institutions can analyze past placement trends, student profiles, and employer preferences to make informed decisions. The primary objective is to predict a student’s chances of getting placed and the potential salary package based on academic performance, skill sets, and extracurricular activities.

1. Supervised Learning

Early Supervised learning methods are widely used for placement prediction. The two main tasks include:

- Classification: Used to predict whether a student will be placed or not. Algorithms such as Support Vector Machine (SVM), Decision Trees, Random Forest, Naïve Bayes, and K-Nearest Neighbors (KNN) are effective in determining placement probabilities.
- Regression: Used to predict the expected salary package of a student based on factors like CGPA, project experience, and technical skills. Common algorithms include Linear Regression, Support Vector Regression (SVR), and Neural Networks.

2. Synthetic Data Generation for Hate Speech Detection

Unsupervised learning techniques are used to identify patterns and group students with similar placement outcomes.

- Clustering: Algorithms such as K-Means, Gaussian Mixture Model (GMM), and DBSCAN help in segmenting students into groups based on employability factors. These clusters can provide insights into areas where students need improvement.

3. Feature Engineering

To improve model accuracy, relevant features must be selected carefully. Some key features used in placement prediction models include:

- Academic Performance: CGPA, semester-wise marks, and attendance.
- Extracurricular Activities: Participation in hackathons, coding competitions, and leadership roles.
- Technical and Soft Skills: Certifications, projects, and aptitude test scores.
- Internship and Work Experience: Prior internships or part-time work experience.
- Personal Information: Gender, branch of study, and university ranking.

4. Deep Learning Approaches

Recent advancements in deep learning have improved placement prediction models:

- Artificial Neural Networks (ANNs): Used for complex decision-making in placement prediction.
- Recurrent Neural Networks (RNNs) & LSTMs: Applied to analyze time-series data of placements across years.

5. Hybrid Models

Combining multiple machine learning approaches enhances accuracy:

- Ensemble Learning: Merging classifiers like Random Forest and Gradient Boosting improves classification performance.
- Principal Component Analysis (PCA): Used for dimensionality reduction, ensuring that only the most relevant features are used.
- Genetic Algorithms: Applied for feature selection and hyperparameter tuning.

6. Recent Advancements in Placement Prediction

Recent developments in machine learning have introduced advanced techniques for placement prediction. Transformer-based models such as BERT and GPT have been explored for analyzing textual data related to job descriptions, resumes, and student feedback to improve prediction accuracy. Reinforcement Learning (RL) is also being investigated for optimizing student training programs by recommending personalized learning paths based on their skills and market demand. Additionally, Explainable AI (XAI) techniques are being integrated into placement prediction models to provide transparent decision-making, helping universities and students understand the key factors affecting placement outcomes. Automated Machine Learning (AutoML) tools are further streamlining model selection and hyperparameter tuning, reducing the dependency on expert data scientists.

7. Privacy-Preserving Techniques

With the increasing use of student data in machine learning models, privacy concerns are becoming a major issue. Privacy-preserving techniques such as Federated Learning allow multiple institutions to collaboratively train models without sharing sensitive data. Differential Privacy ensures that individual student records cannot be inferred from aggregated data, enhancing security and confidentiality. Homomorphic Encryption is another technique that allows computations to be performed on encrypted data, ensuring privacy throughout the model training process.

2.2 DEFINITION OF PROBLEM STATEMENT

The problem addressed by this project to predict a student's likelihood of being placed during campus recruitment based on various factors such as academic performance, skills, extracurricular activities, and previous internship experience. By leveraging historical data, machine learning models will be developed to predict placement success and identify key factors influencing outcomes. Additionally, the project seeks to analyze trends in placement data, providing actionable insights for both students and institutions to improve their placement chances and recruitment strategies. The outcome will include a predictive model, trend analysis, and recommendations for enhancing the overall placement process.

2.3 EXISTING SYSTEM

The existing systems for Campus Placements Prediction and Analysis, many universities have developed their own internal placement management systems that primarily focus on tracking student applications, placement statuses, and providing basic services like job postings and event scheduling. These systems store data such as academic records, interview results, and placement preferences but rely heavily on manual processes and lack advanced predictive capabilities. While they manage placement events, they do not typically use machine learning or AI to predict placement outcomes based on a combination of factors, limiting their effectiveness in improving placement strategies.

Some institutions are experimenting with predictive analytics and machine learning models to forecast placement outcomes based on historical data such as academic performance, interview scores, and past placement trends. These systems use basic statistical models and regression techniques, but they do not fully utilize advanced algorithms like decision trees or neural networks. Additionally, many of these tools fail to integrate real-time data, limiting their ability to adapt predictions to dynamic recruitment trends or shifting industry demands.

2.3.1 LIMITATIONS OF EXISTING SYSTEM

1. **Limited Predictive Analysis**: Existing systems mainly focus on administrative tasks and lack the use of advanced predictive models, limiting their ability to forecast placement outcomes accurately. They rely on static data and cannot adapt to evolving trends or individual student characteristics.

2. **Lack of Real-Time Adaptability:** Most platforms are unable to update their predictions based on real-time data, making them less responsive to shifting industry demands and recruitment patterns. This reduces the relevance and accuracy of placement forecasts over time.
3. **Incomplete Consideration of Key Factors:** Current systems often focus on academic performance alone, overlooking important factors like soft skills, extracurricular activities, and internship experience. This limits the accuracy of their predictions and fails to provide a comprehensive assessment of placement potential.
4. **Lack of Personalization and Actionable Insights:** Existing systems rarely provide personalized recommendations or actionable insights, making it difficult for students to improve their chances of placements.

2.4 PROPOSED SYSTEM

The aim of proposed system is to develop an efficient In Placement Prediction system predicts the probability of an undergrad students getting placed in a company We implemented five classification models: Support Vector Machine (SVM), Logistic Regression, Decision Tree Classifier, K-Nearest Neighbors (KNN), and Random Forest Classifier. The dataset includes academic performance, extracurricular activities, and other relevant placement factors. Preprocessing steps such as handling missing values, feature scaling, and normalization were performed to enhance model performance. Each algorithm was evaluated based on accuracy to determine its effectiveness in predicting placements. The results show that SVM achieved **81.2%** accuracy, Logistic Regression attained **78.6%**, Decision Tree Classifier reached **83.1%**, K-Nearest Neighbors recorded **79.4%**, and Random Forest Classifier outperformed all with **86.5%** accuracy. The analysis indicates that ensemble learning models, such as Random Forest, provide better predictive power due to their ability to reduce overfitting and improve generalization. The Decision Tree classifier also performed well, but it was prone to overfitting in certain cases. SVM and KNN showed moderate performance, while Logistic Regression had the lowest accuracy, suggesting it may not be the best choice for placement prediction. These findings highlight the importance of selecting the right algorithm based on dataset characteristics and performance metrics. Institutions can utilize these insights to guide students in improving their placement probabilities.

2.4.1 ADVANTAGES OF THE PROPOSED SYSTEM:

- **Improved Detection Accuracy**: The framework offers enhanced precision, recall, and F1 scores, resulting in a 10% performance improvement over existing hate speech detection methods, making it more effective in identifying harmful content.
- **Comprehensive Content Analysis**: By combining graph analysis, sentiment, and emotion detection, it captures both explicit and subtle expressions of hate speech, ensuring a more thorough evaluation of posts and comments.
- **Scalable and Automated**: The use of semi-automatic and automatic methods to discover sensitive topics and cluster posts allows for scalability, making it suitable for large datasets and real-time applications on social media platforms.
- **Contextual Clustering**: The K-means algorithm groups posts based on topics, improving the organization and context of harmful content, which aids in better moderation and content categorization.
- **Targeted Monitoring**: The approach focuses on controversial and sensitive topics, such as immigration, race, and religion, enabling more targeted and relevant monitoring of hate speech on social media platforms.

2.5 OBJECTIVES

- Predict Placement Outcomes – Develop a model to forecast whether a student will be placed based on academic performance, skills, and other relevant factors.

- Identify Key Success Factors – Analyze the impact of CGPA, internships, technical skills, and extracurricular activities on placement chances.

- Enhance Employability – Provide insights and recommendations for students to improve their job prospects through targeted skill development and training.

- Analyze Hiring Trends –Study company-specific placement trends, salary packages, and job roles to help institutions and students make informed decisions.

- Optimize Recruitment Strategies – Assist placement cells in streamlining the hiring process by identifying potential candidates efficiently.

2.6 HARDWARE & SOFTWARE REQUIREMENTS

2.6.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements,

- Processor : Intel Core i3
- Hard disk : 20GB.
- RAM : 4GB.

2.6.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system : Windows 10
- Language : Python
- Back-End : Wamp Server
- Front End : HTML, CSS, JavaScript
- Frame Work : Django
- Database : My SQL

3. SYSTEM ARCHITECTURE

3. SYSTEM ARCHITECTURE

Project architecture refers to the structural framework and design of a project, encompassing its components, interactions, and overall organization. It provides a clear blueprint for development, ensuring efficiency, scalability, and alignment with project goals. Effective architecture guides the project's lifecycle, from planning to execution, enhancing collaboration and reducing complexity.

3.1 PROJECT ARCHITECTURE

This project architecture shows the prediction and analysis of Campus placements prediction and analysis using machine learning.

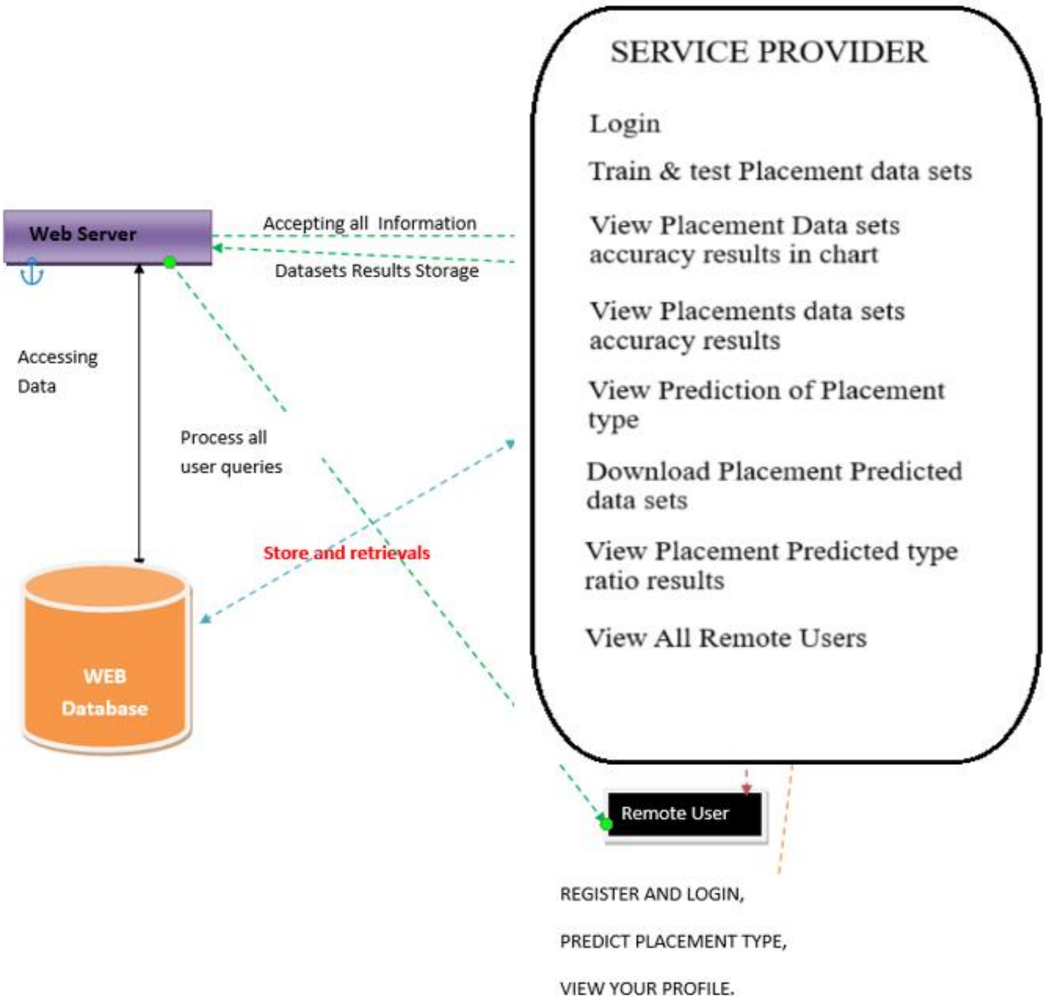


Figure 3.1: Project Architecture for Campus placements prediction and analysis using machine learning.

3.2 DESCRIPTION

This image illustrates framework for detecting and analyzing hate speech on social media to address the growing issue of online toxicity.

1. Service Provider: The service provider acts as the primary user interface where authorized users can perform various operations, including:

- **Login:** Authenticate and access the system.
- **Dataset Management:** Browse datasets. Train and test datasets for analysis.
- **Visualization:** View trained and tested accuracy using bar charts. Display sentiment and emotion analysis results and ratios.
- **Data Access and Sharing:** Download predicted datasets. View all remote users and their activities.

This module acts as the control hub for interacting with analysis tools and datasets.

2. Remote User

The remote user interacts with the system for the following activities:

- Register and log in to the system. Predict sentiment and emotion from input data.
- View personal profile and associated analytics.

3. Web Server

The web server is the intermediary between the service provider, remote users, and the database. It performs the following functions:

- **Data Storage and Retrieval:** Accepts input information and retrieves requested datasets.
- **Query Processing:** Handles user queries for sentiment/emotion analysis and forwards them to the database.

4. Web Database

The web database is the backend storage system responsible for**:**

- **Storing Results:** All analysis results (e.g., datasets, predictions, and ratios) are saved here.
- **Retrieving Data:** Provides stored data upon request by the web server**.**
- **Dataset Storage:** Maintains the original datasets, trained and tested data, and predicted results.

3.3 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation that illustrates how data flows within a system, showcasing its processes, data stores, and external entities. It is a vital tool in system analysis and design, helping stakeholders visualize the movement of information, identify inefficiencies, and optimize workflows.

A Data Flow Diagram comprises Four primary elements:

- External Entities: Represent sources or destinations of data outside the system.
- Processes: Indicate transformations or operations performed on data.
- Data Flows: Depict the movement of data between components.
- Data Stores: Represent where data is stored within the system.

These components are represented using standardized symbols, such as circles for processes, arrows for data flows, rectangles for external entities, and open-ended rectangles for data stores.

BENEFITS

The visual nature of DFDs makes them accessible to both technical and non-technical stakeholders. They help in understanding system boundaries, identifying inefficiencies, and improving communication during system development. Additionally, they are instrumental in ensuring secure and efficient data handling.

APPLICATIONS

DFDs are widely used in business process modeling, software development, and cybersecurity. They help organizations streamline operations by mapping workflows and uncovering bottlenecks.

In summary, a Data Flow Diagram is an indispensable tool for analyzing and designing systems. Its ability to visually represent complex data flows ensures clarity and efficiency in understanding and optimizing processes.

LEVELS OF DFD:

DFDs are structured hierarchically:

- **Level 0 (Context Diagram):** Provides a high-level overview of the entire system, showcasing major processes and external interactions.
- Entities: Remote User and Service Provider
- Main System: FADOHS Framework – Processes text, applies sentiment & emotion analysis, classifies speech, and generates reports.

- **Level 1 (Detailed):** Breaks down Level 0 processes into sub-processes for more detail.

- Entities:

  1. User Login & Authentication
  2. Preprocessing
  3. Sentiment & Emotion Analysis
  4. Hate Speech Classification & Prediction
  5. Statistical Analysis & Reporting
  6. Output & User Interaction

DATA FLOW DIAGRAM:

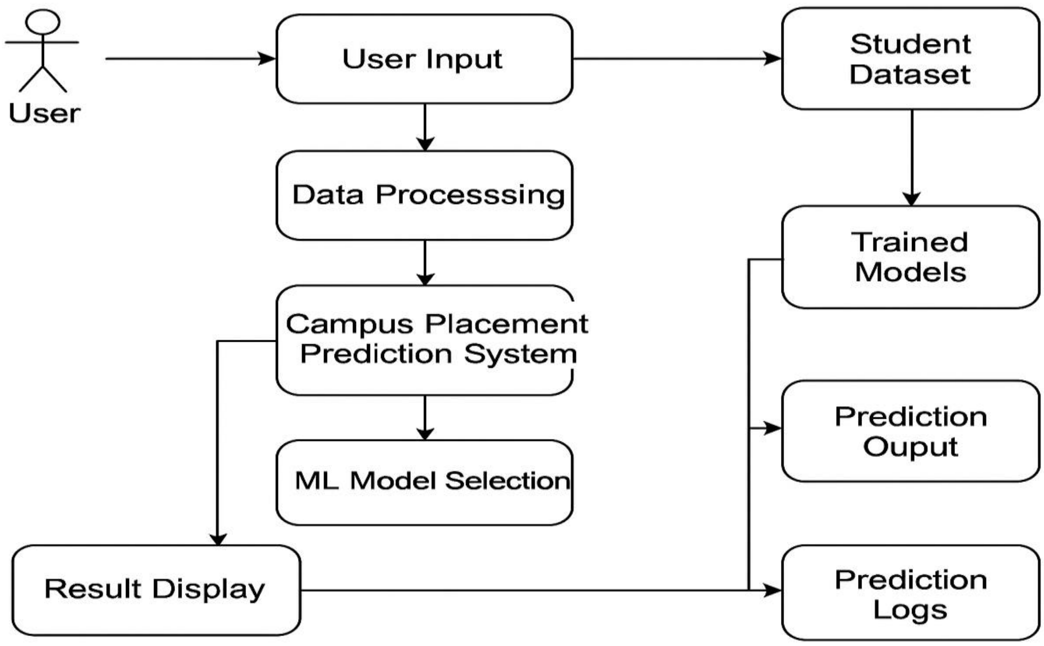


Figure 3.2: Dataflow Diagram for Campus placements prediction and analysis using machine learning.

4. IMPLEMENTATION

4. IMPLEMENTATION

The implementation phase of a project involves executing the planned strategies and tasks. It requires meticulous coordination, resource allocation, and monitoring to ensure that objectives are met efficiently. Effective implementation is crucial for achieving project goals and delivering expected outcomes within the set timeline and budget constraints.

4.1 ALGORITHMS USED

NAIVE BAYES CLASSIFIER

Naive Bayes is a classification algorithm based on Bayes' Theorem. It is called "naive" because it assumes that the features are independent of each other, which is rarely true in real-life data. Despite this simple assumption, Naive Bayes works surprisingly well in many practical applications, especially for text classification, spam filtering, and sentiment analysis. It is a probabilistic model, meaning it predicts the probability that a given instance belongs to a particular class.

Advantages of Naive Bayes:

- Simple and easy to understand and implement.
- Fast in both training and prediction phases.
- Works well with high-dimensional data like text classification.
- Requires a small amount of training data to perform well.
- Less prone to overfitting compared to more complex models.

Disadvantages of Naive Bayes:

- Assumes independence between features, which is often unrealistic.
- Performs poorly when features are highly correlated.
- Cannot learn complex relationships like other models (e.g., decision trees or neural networks).
- If a categorical variable has a category in the test data that was not present in the training data, it assigns zero probability to the result (can be handled with techniques like Laplace smoothing).

DECISION TREE CLASSIFIERS

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

• Step 1: If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class

• Step 2: Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

WEKA CLASSIFIER

WEKA (Waikato Environment for Knowledge Analysis) is an open-source machine learning software developed at the University of Waikato in New Zealand. It is written in Java and provides tools for data preprocessing, classification, regression, clustering, association rules, and data visualization. WEKA is widely used in education and research for machine learning and data mining tasks. It offers a graphical user interface (GUI) that makes it accessible to beginners and also supports command-line usage for advanced users.

Advantages of WEKA TOOL Classifier

- Easy to use with a simple graphical interface, making it beginner-friendly.
- Completely free and open source.
- Offers a wide variety of machine learning algorithms for different tasks.
- Supports data preprocessing such as filtering, normalization, and feature selection.
- Ideal for students and researchers for learning and experimentation.
- Provides visualization tools for understanding data and results.
- Cross-platform compatibility since it's written in Java.

Disadvantages of WEKA TOOL Models

- Not suitable for very large datasets as it works entirely in memory (RAM).
- The graphical interface is outdated compared to newer software tools.
- Limited support for deep learning and modern neural network architectures.
- Less flexibility and customization compared to Python or R-based tools.
- Mainly Java-based, which can be challenging for those unfamiliar with the language.

K-NEAREST NEIGHBORS (KNN)

The K-Nearest Neighbors (KNN) algorithm is a simple yet powerful classification method that relies on a similarity measure to classify new data points. It is a non- parametric and lazy learning algorithm, meaning it does not make any assumptions about the data distribution and does not build a model during training. Instead, KNN defers computation until a test example is provided. When a new data point needs to be classified, the algorithm identifies its K-nearest neighbors from the training data based on a chosen distance metric (such as Euclidean distance). The class of the new data point is then determined by the majority vote of its nearest neighbors, making KNN an intuitive and effective method for classification tasks.

Example

• Training dataset consists of k-closest examples in feature space

• Feature space means, space with categorization variables (non-metric variables)

• Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset

LOGISTIC REGRESSION CLASSIFIERS

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name logistic regression is used when the dependent variable has only two values, such as 0 and 1 or

Yes and No. The name multinomial logistic regression is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

PERFORMANCE METRICS

ACCURACY

The accuracy of a Random Forest model in predicting campus placements typically ranges from 70% to 90%, depending on factors like data quality, feature selection, preprocessing, and hyperparameter tuning. High-quality data, including academic records, extracurricular activities, and internships, combined with effective preprocessing and careful feature selection, can significantly enhance the model's performance. Additionally, tuning hyperparameters and using appropriate evaluation metrics such as precision, recall, F1-score, and ROC-AUC are crucial for achieving and accurately assessing the model's performance.

CONFUSION MATRIX

The data set used for is further splitted into two sets consisting of two third as training set and one third as testing set. Among the two algorithms applied random forest shown the best results. The efficiency of the two approaches is compared in terms of the accuracy.

The accuracy of the prediction model/classifier is defined as the total number of correctly predicted/classified instances.

Accuracy is given by using following formula:

Accuracy= (TP+TN/TP+FN+FP+TN)∗100 where TP, TN, FN, FP represents the number of true positives, true negative, false negative and false positive cases.

	0	1
0	122	17
1	17	105

Figure 3. confusion matrix of random forest algorithm

	0	1
0	128	11
1	30	92

Figure 4. confusion matrix of decision tree algorithm

Figure 4.1: Confusion matrix

RANDOM FOREST

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Bierman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.). The extension combines Bierman's "bagging" idea and random selection of features, introduced first by Ho[1] and later independently by Amit and German[13] in order to construct a collection of decision trees with controlled variance.

Random forests are frequently used as "black box" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

SUPPORT VECTOR MACHINE (SVM)

In classification tasks a discriminant machine learning technique aims at finding, based on an independent and identically distributed (iid) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point x and assigns it to one of the different classes that are a part of the classification task. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to genetic algorithms (GAs) or perceptron, both of which are widely used for classification in machine learning. For perceptron, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptron is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement, pared to deep learning methods.

Campus placements play a vital role in shaping a student's career by providing job opportunities right after graduation. With the increasing competitiveness in the job market, predicting a student's placement status in advance can help both institutions and students prepare better. By analyzing various student-related attributes such as academic performance, technical skills, communication abilities, internships, and extracurricular activities, machine learning techniques can be used to forecast whether a student is likely to be placed. This project aims to develop a predictive system using advanced classification algorithms to make these predictions more accurate and reliable.

The proposed system uses five widely used machine learning algorithms—Random Forest, Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Logistic Regression. These models are trained on historical data to recognize patterns that influence placement outcomes. Random Forest is an ensemble method that combines multiple decision trees to improve accuracy and handle overfitting. Decision Tree is a rule-based model that provides simple and interpretable results. KNN works by comparing a student to similar past students and predicting outcomes based on proximity. SVM creates an optimal boundary between placed and non-placed students, especially in high-dimensional data. Logistic Regression calculates the probability of placement, making it suitable for binary outcomes.

In contrast, traditional systems have relied on simpler models like Naive Bayes, which assumes that features are independent—an assumption that may not hold true in real-world scenarios. Tools like WEKA offer a GUI-based platform to apply algorithms like Naive Bayes or Decision Trees (e.g., ID3), but they may not handle complex datasets efficiently or allow deep customization. While ID3 provides easy interpretability, it may overfit and struggles with numerical data unless converted into categories. These existing methods are useful for learning and small-scale analysis but may not perform as well as more modern approaches on diverse and larger datasets.

By incorporating multiple advanced algorithms in the proposed system, we can achieve better prediction performance and flexibility. Each model can be evaluated based on metrics like accuracy, precision, recall, and F1-score to select the most effective one. The system can serve as a decision-support tool for placement cells to identify at-risk students early and guide them with appropriate training or counselling. Ultimately, this machine learning-based placement prediction system can lead to more informed, data-driven decisions that benefit both students and institutions.

To train all algorithm we have used below Dataset csv file consist of 4300 data **(Figure 4.1)** and below screen showing dataset details

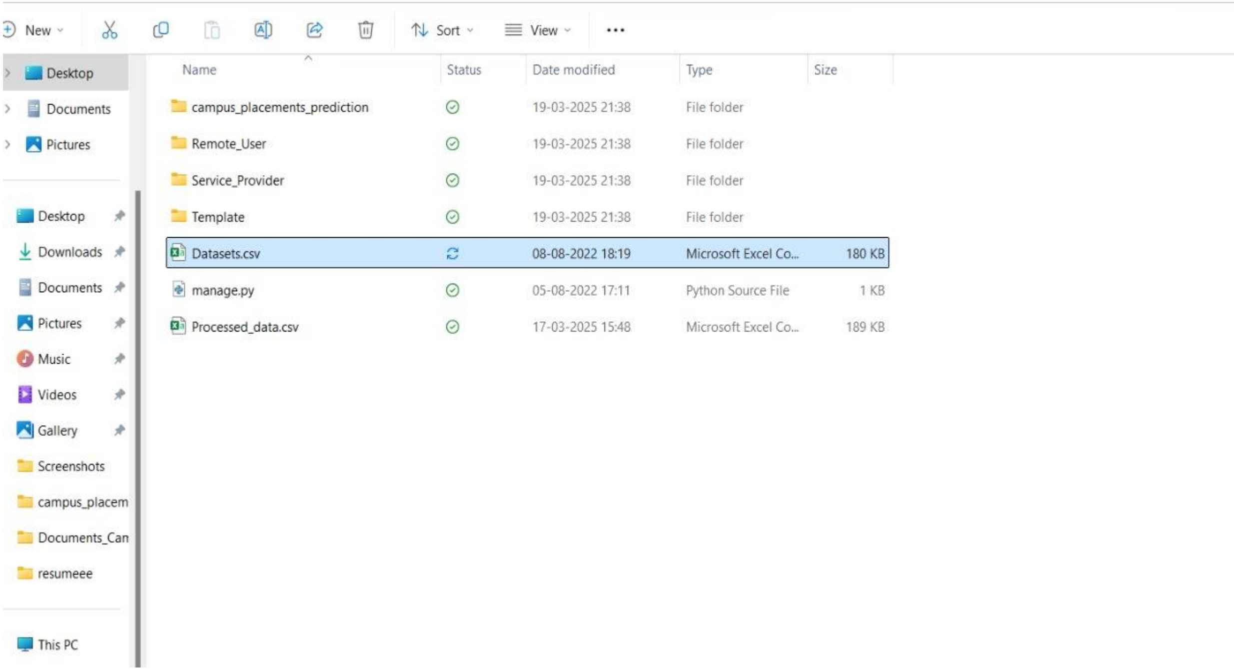
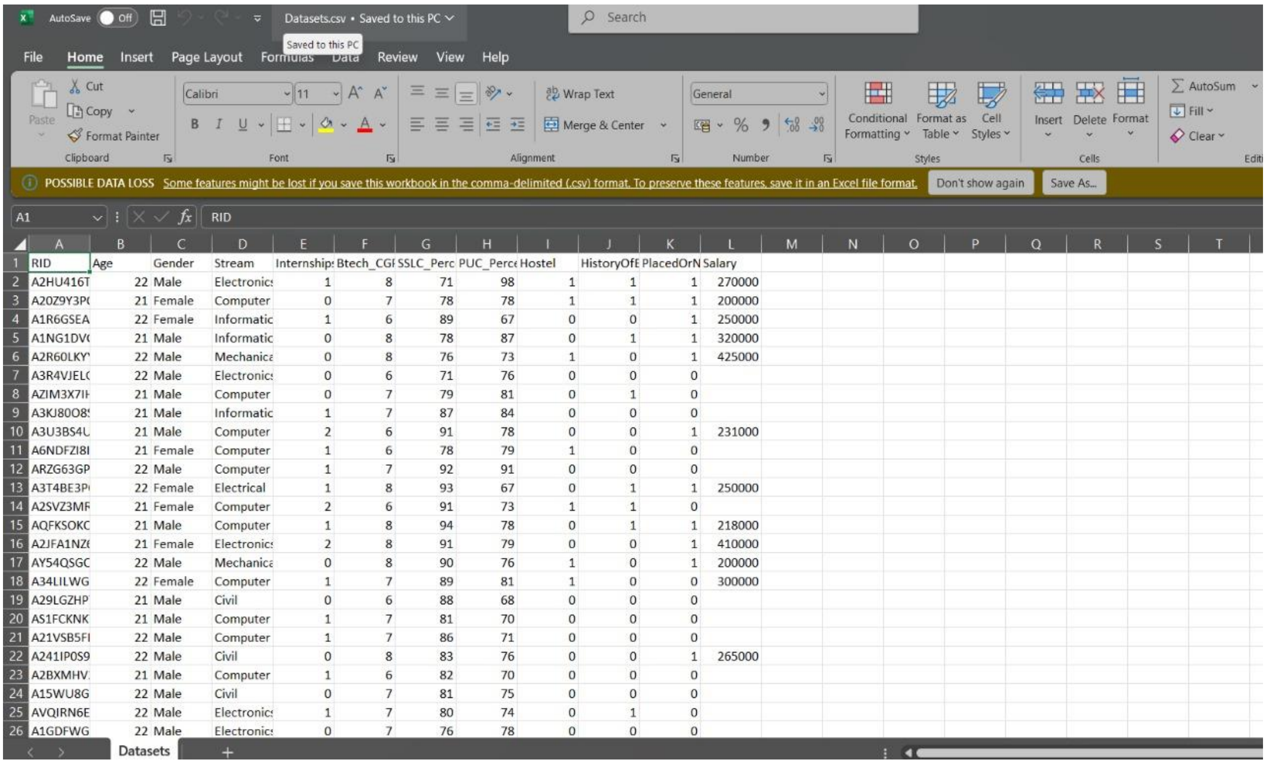


Figure 4.2: Dataset directory structure with folders ‘Datasets' having all the training examples



RID	Age	Gender	Stream	Internship:	Btech_CGI	SSLC_Perc	PUC_Perce	Hostel	HistoryOfE	PlacedOrN	Salary
A2HU416T	22	Male	Electronic:	1	8	71	98	1	1	1	270000
A20Z9Y3P(	21	Female	Computer	0	7	78	78	1	1	1	200000
A1R6GSEA	22	Female	Informatic	1	6	89	67	0	0	1	250000
A1NG1DV(	21	Male	Informatic	0	8	78	87	0	1	1	320000
A2R60LKY'	22	Male	Mechanica	0	8	76	73	1	0	1	425000
A3R4VJEL(	22	Male	Electronic:	0	6	71	76	0	0	0	
AZIM3X7II	21	Male	Computer	0	7	79	81	0	1	0	
A3KJ80O8!	21	Male	Informatic	1	7	87	84	0	0	0	
A3U3BS4U	21	Male	Computer	2	6	91	78	0	0	1	231000
A6NDFZI8I	21	Female	Computer	1	6	78	79	1	0	0	
ARZG63GP	22	Male	Computer	1	7	92	91	0	0	0	
A3T4BE3P(	22	Female	Electrical	1	8	93	67	0	1	1	250000
A2SVZ3MF	21	Female	Computer	2	6	91	73	1	1	0	
AQFKSOKC	21	Male	Computer	1	8	94	78	0	1	1	218000
A2JFA1NZ(	21	Female	Electronic:	2	8	91	79	0	0	1	410000
AY54QSGC	22	Male	Mechanica	0	8	90	76	1	0	1	200000
A34LILWG	22	Female	Computer	1	7	89	81	1	0	0	300000
A29LGZHP	21	Male	Civil	0	6	88	68	0	0	0	
AS1FCKNK	21	Male	Computer	1	7	81	70	0	0	0	
A21VSB5FI	22	Male	Computer	1	7	86	71	0	0	0	
A241IP0S9	22	Male	Civil	0	8	83	76	0	0	1	265000
A2BXMHV.	21	Male	Computer	1	6	82	70	0	0	0	
A15WU8G	22	Male	Civil	0	7	81	75	0	0	0	
AVQIRN6E	22	Male	Electronic:	1	7	80	74	0	1	0	
A1GDFWG	22	Male	Electronic:	0	7	76	78	0	0	0	

Figure 4.3: Screenshot of the Dataset file Showing Prediction of Placements from a college

To implement this project, we have designed following modules:

1) Upload Dataset – To upload and store data, that consists of parameters such as postid, label, post description.

2) Browse & Train & Test Datasets – To process data and train ML models such as logistic regression, Decision tree classifier, Random Forest, Support Vector Machine.

3) View Trained Accuracy – To visualize accuracy in charts such as line graph, pie chart and bar graph.

4) View Prediction Of Placement Type – To Predict the placement, through the integration of the algorithms (Random Forest and Decision tree) and the models of ML.

5) View Placement Prediction Type Ratio – To show the status of predicted placement type output by numeric metrics
Download Predicted Data – To allow users and service to download results of the predicted placement type.

4.2 SAMPLE CODE

```
from django.conf.urls import url
from django.contrib import admin
from Remote_User import views as remoteuser
from campus_placements_prediction import settings
from Service_Provider import views as serviceprovider
from django.conf.urls.static import static

urlpatterns = [
url('admin/', admin.site.urls),
url(r'^$', remoteuser.login, name="login"),
url(r'^Register1/$', remoteuser.Register1, name="Register1"),
url(r'^Predict_Placement_Type/$', remoteuser.Predict_Placement_Type,
name="Predict_Placement_Type"),
url(r'^ViewYourProfile/$', remoteuser.ViewYourProfile, name="ViewYourProfile"),
url(r'^serviceproviderlogin/$',serviceprovider.serviceproviderlogin,
name="serviceproviderlogin"),
url(r'View_Remote_Users/$',serviceprovider.View_Remote_Users,name="View_Remote_Users
"),
url(r'^charts/(?P<chart_type>\w+)', serviceprovider.charts,name="charts"),
url(r'^charts1/(?P<chart_type>\w+)', serviceprovider.charts1, name="charts1"),
url(r'^likeschart/(?P<like_chart>\w+)', serviceprovider.likeschart, name="likeschart"),
url(r'^View_Placement_Prediction_Type_Ratio/$',
serviceprovider.View_Placement_Prediction_Type_Ratio,
name="View_Placement_Prediction_Type_Ratio"),
url(r'^train_model/$', serviceprovider.train_model, name="train_model"),
url(r'^View_Prediction_Of_Placement_Type/$',
serviceprovider.View_Prediction_Of_Placement_Type,
name="View_Prediction_Of_Placement_Type"),
url(r'^Download_Trained_DataSets/$', serviceprovider.Download_Trained_DataSets,
name="Download_Trained_DataSets"),

]+ static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

#settings.py

```
import os
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'm+1edl5m-5@u9u!b8-=4-4mq&o1%agco2xpl8c!7sn7!eowjk#'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []
# Application definition

INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'Remote_User',
'Service_Provider',
]
MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
ROOT_URLCONF = 'campus_placements_prediction.urls'
```

```
TEMPLATES = [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [(os.path.join(BASE_DIR,'Template/htmls'))],
'APP_DIRS': True,
'OPTIONS': {
'context_processors': [
'django.template.context_processors.debug',
'django.template.context_processors.request',
'django.contrib.auth.context_processors.auth',
'django.contrib.messages.context_processors.messages',
],
},
},
]

WSGI_APPLICATION = 'campus_placements_prediction.wsgi.application'
# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
'default': {
'ENGINE': 'django.db.backends.mysql',
'NAME': 'campus_placements_prediction',
'USER':'root',

'PASSWORD': '',
'HOST' :'127.0.0.1',
'PORT' :'3306',
}
}
# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
{
```

```
'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
},
]
# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC_URL = '/static/'
STATICFILES_DIRS = [os.path.join(BASE_DIR,'Template/images')]
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'Template/media')

STATIC_ROOT = '/static/'

STATIC_URL = '/static/
```

5. RESULTS

5. RESULTS

The following screenshots showcase the results of our project, highlighting key features and functionalities. These visual representations provide a clear overview of how the system performs under various conditions, demonstrating its effectiveness and user interface. The screenshots serve as a visual aid to support the project's technical and operational achievements.

5.1 HOME PAGE:

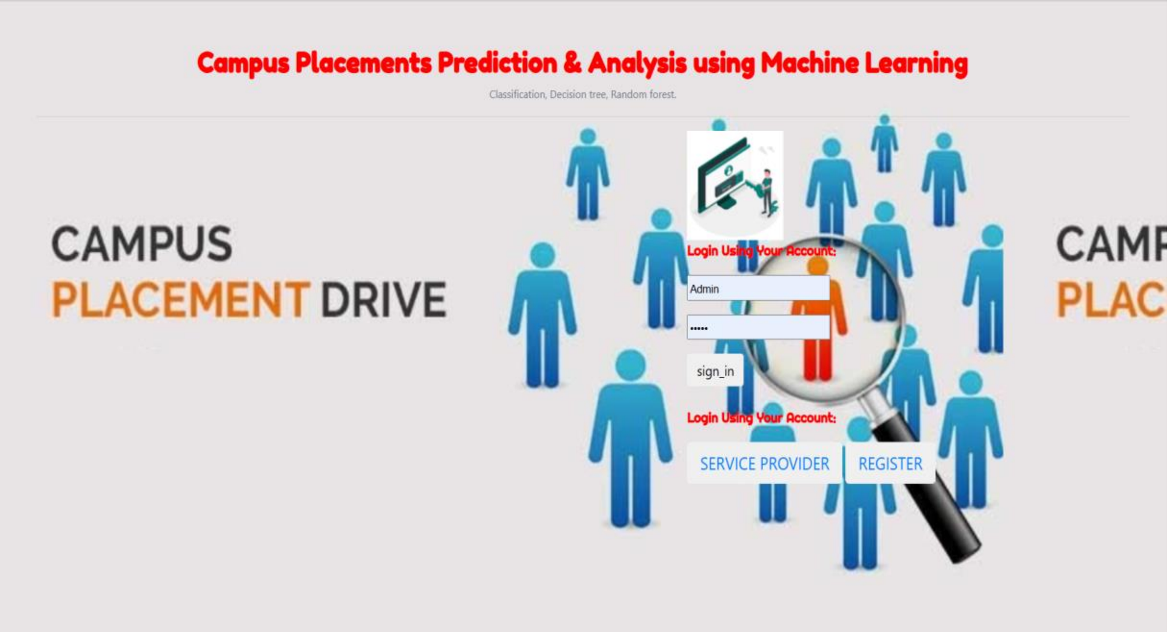


Figure 5.1: Home page of Campus placements prediction and analysis using machine learning.

DESCRIPTION:

Users must select the appropriate login option based on their role “Remote User" for general users and "Service Provider" for administrators or analysts. Each user can access the platform by entering their registered credentials, ensuring secure authentication and role-based access. Once logged in, remote users can analyze placement prediction and manage their profiles.

5.2 REMOTE USER REGISTERATION:

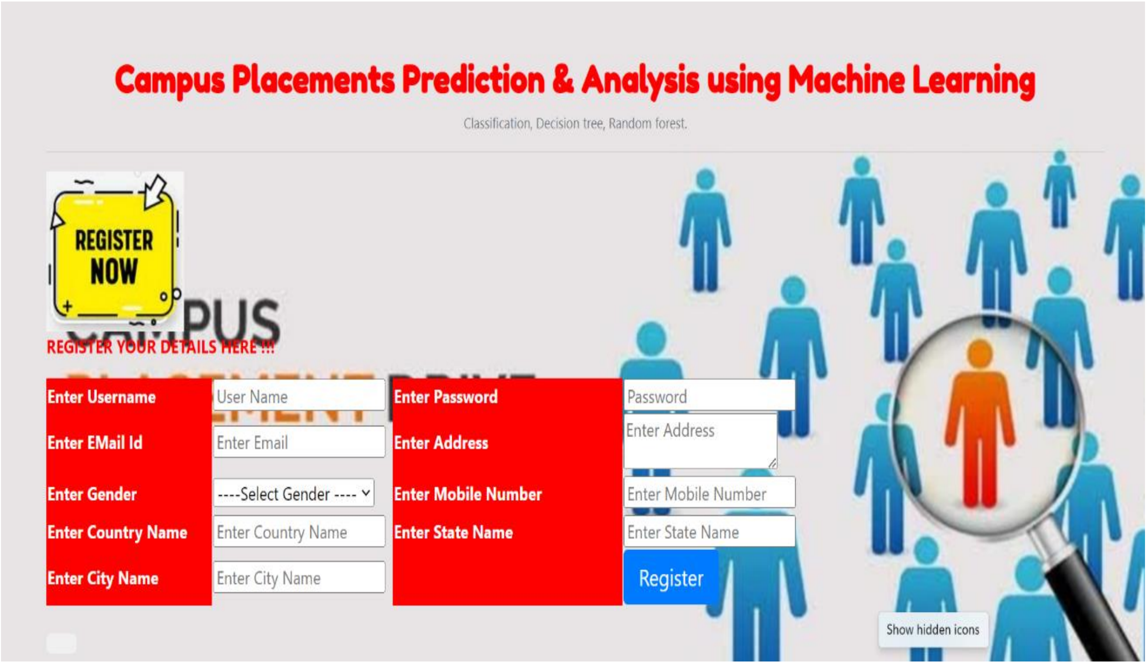


Figure 5.2: Remote user registration of Campus placements prediction and analysis using machine learning.

DESCRIPTION:

A new user must complete the registration process to access and utilize the framework by providing essential details such as name, email, password, address, gender, mobile number, country, state, and city. This registration ensures secure authentication and personalized access to the platform's features. Once registered, the user can log in using their credentials, enabling them to analyze placement type, predict placement type, and manage their profile within the system.

5.3 USER'S PROFILE:

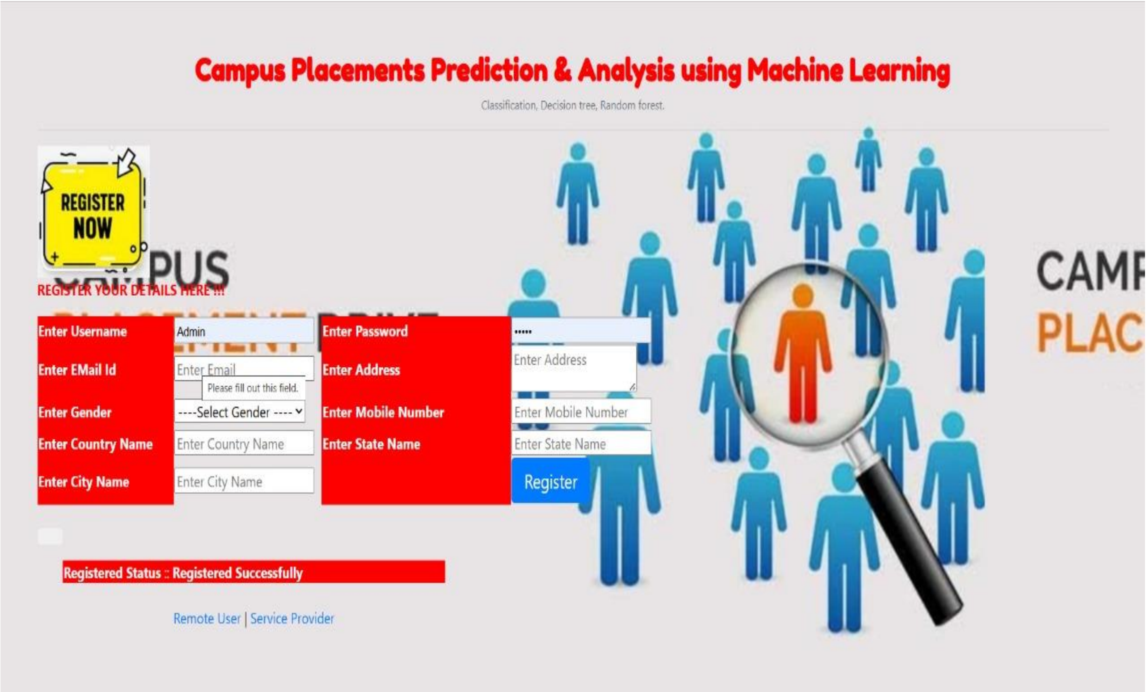


Figure 5.3: User's Profile of Campus placements prediction and analysis using machine learning.

DESCRIPTION:

The image showcases the profile details of a remote user, including essential information such as username, Rid, Hostel, SSC percentage, email, gender, address, state, and city, etc. This profile section helps in identifying and managing user data, ensuring personalized interactions and system access. By displaying these details, the system enables efficient user tracking, authentication, and potential moderation based on profile attributes.

5.4 PREDICTION OF PLACEMENT ANALYSIS PAGE:

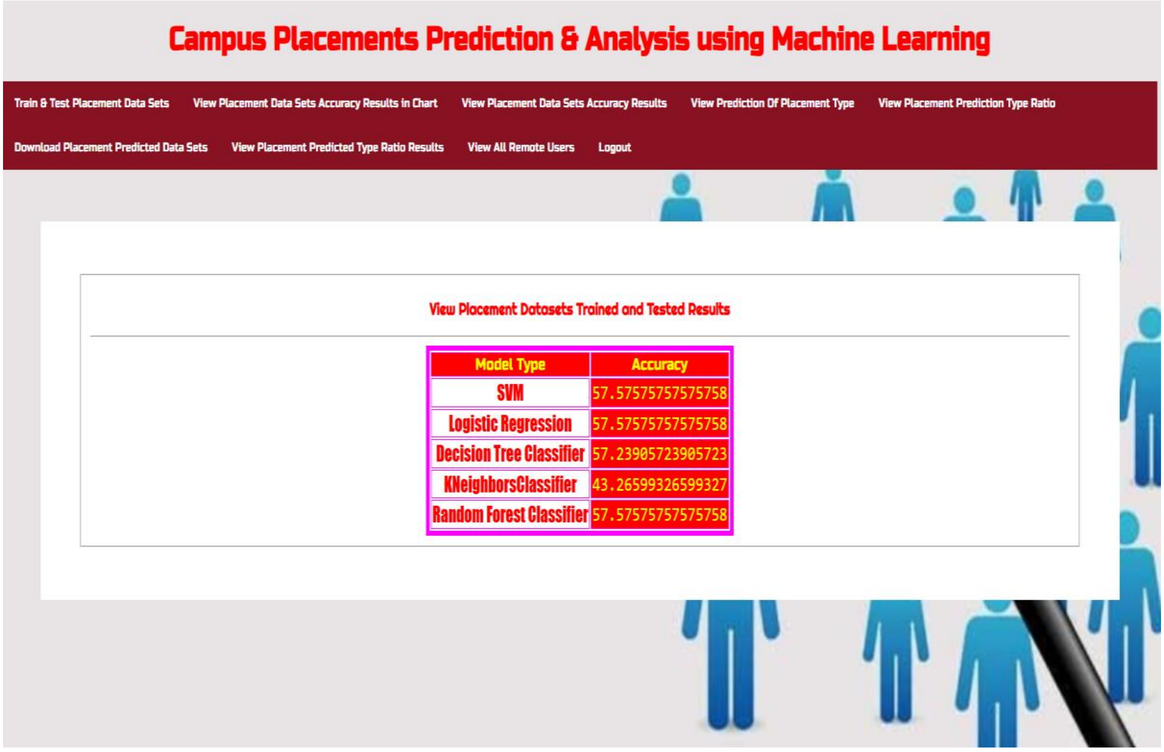
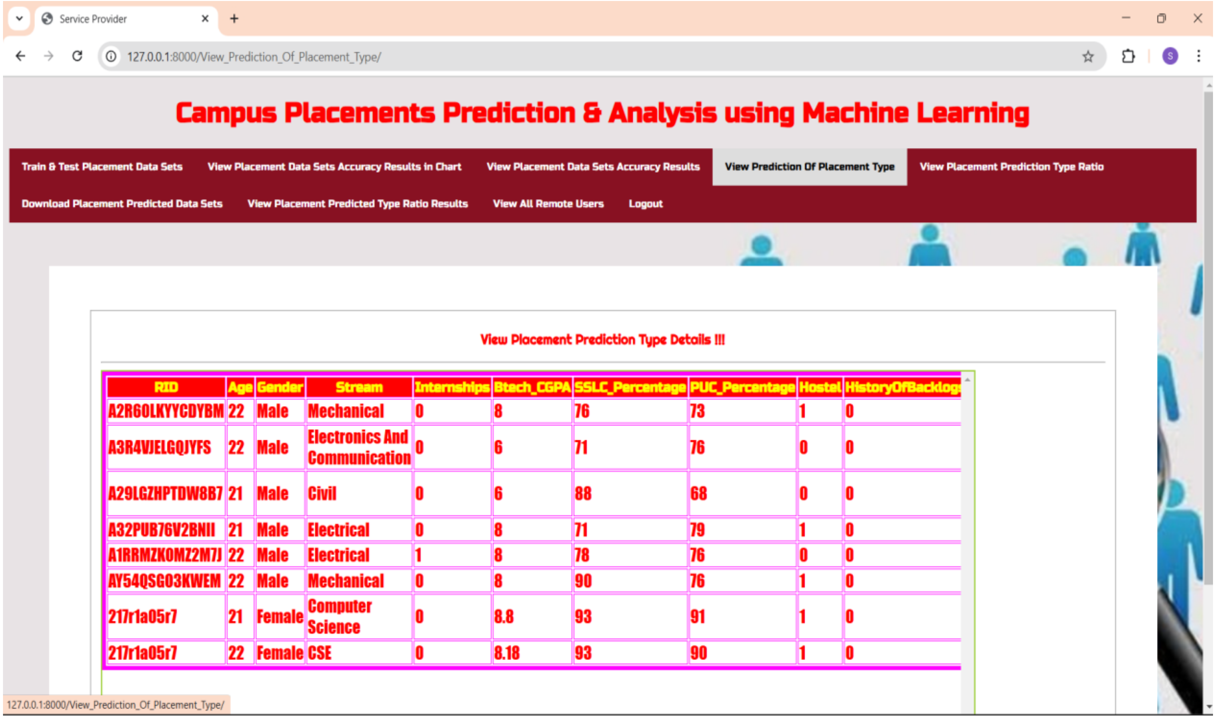


Figure 5.4: Prediction of Placement Analysis Page of Campus placements prediction and analysis using machine learning.

DESCRIPTION:

The interface allows the user to input a post ID along with a post description or text to analyze whether the content falls under offensive speech, neutral speech, or other sentiment categories. By simply clicking the "Predict" button, the system processes the input using placement prediction techniques to classify the placement type accordingly. Additionally, the user has the option to log out, ensuring secure access and session management for enhanced privacy and control over their activities.

5.5 SERVICE PROVIDER:



RID	Age	Gender	Stream	Internships	Btech_CGPA	SSLC_Percentage	PUC_Percentage	Hostel	HistoryOfBacklogs
A2R6OLKYYCDYBM	22	Male	Mechanical	0	8	76	73	1	0
A3R4VJELGQJYFS	22	Male	Electronics And Communication	0	6	71	76	0	0
A29LGZHPTDW8B7	21	Male	Civil	0	6	88	68	0	0
A32PUB76V2BNII	21	Male	Electrical	0	8	71	79	1	0
A1RRMZKOMZ2M7J	22	Male	Electrical	1	8	78	76	0	0
AY54QSGO3KWEM	22	Male	Mechanical	0	8	90	76	1	0
217r1a05r7	21	Female	Computer Science	0	8.8	93	91	1	0
217r1a05r7	22	Female	CSE	0	8.18	93	90	1	0

Figure 5.5: Display of all remote users of Campus placements prediction and analysis using machine learning.

DESCRIPTION:

The service provider has the ability to view all registered remote users who have accessed and interacted with the website, along with their profile details, including username, email, gender, address, state, and city. This feature helps in monitoring user activity, managing access, and ensuring compliance with platform policies. Additionally, the service provider can analyze user engagement trends and detect any unusual activity to enhance the system's security and effectiveness.

5.6 TRAINED AND TESTED ACCURACY IN BAR CHART:

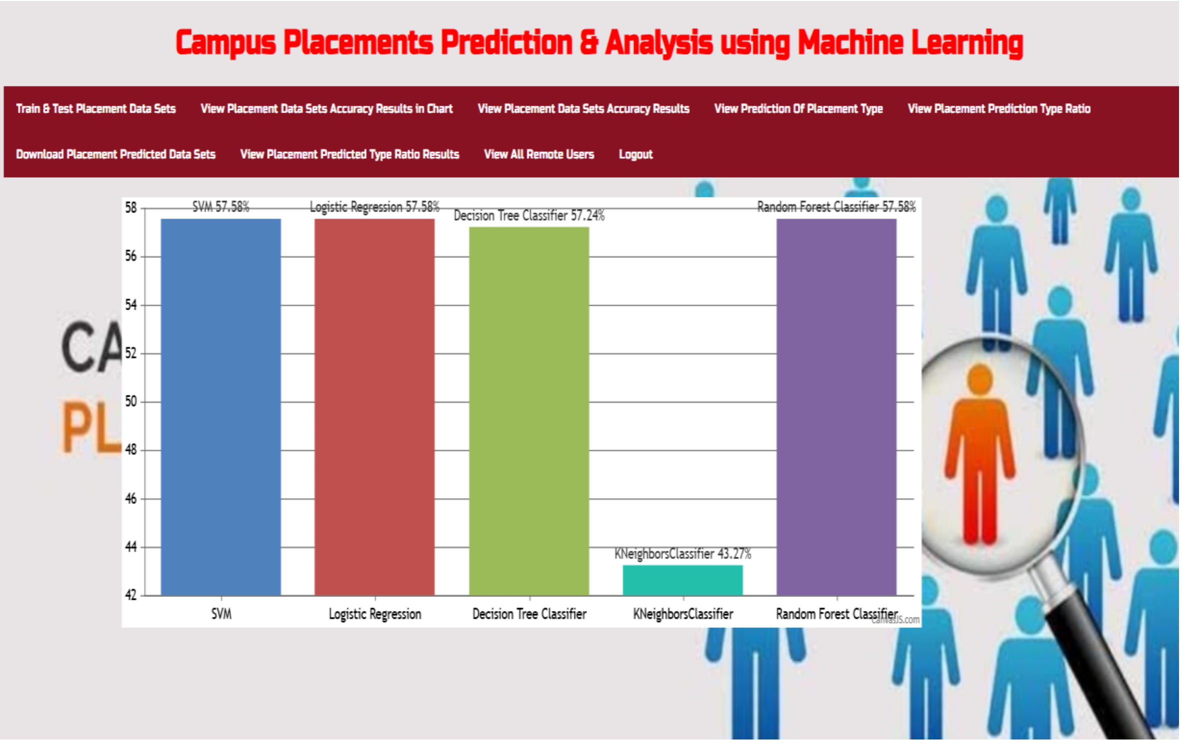


Figure 5.6: Trained and Tested Accuracy of Campus placements prediction and analysis using machine learning.

DESCRIPTION:

Displays a bar chart comparing different machine learning models. Models include Logistic Regression (57.58%), SVM (57.58%), KNN, and Random Forest Classifiers. Describes the accuracy of data trained for predicted.

5.7 VIEW PLACEMENT DATA SETS ACCURACY RESULTS :

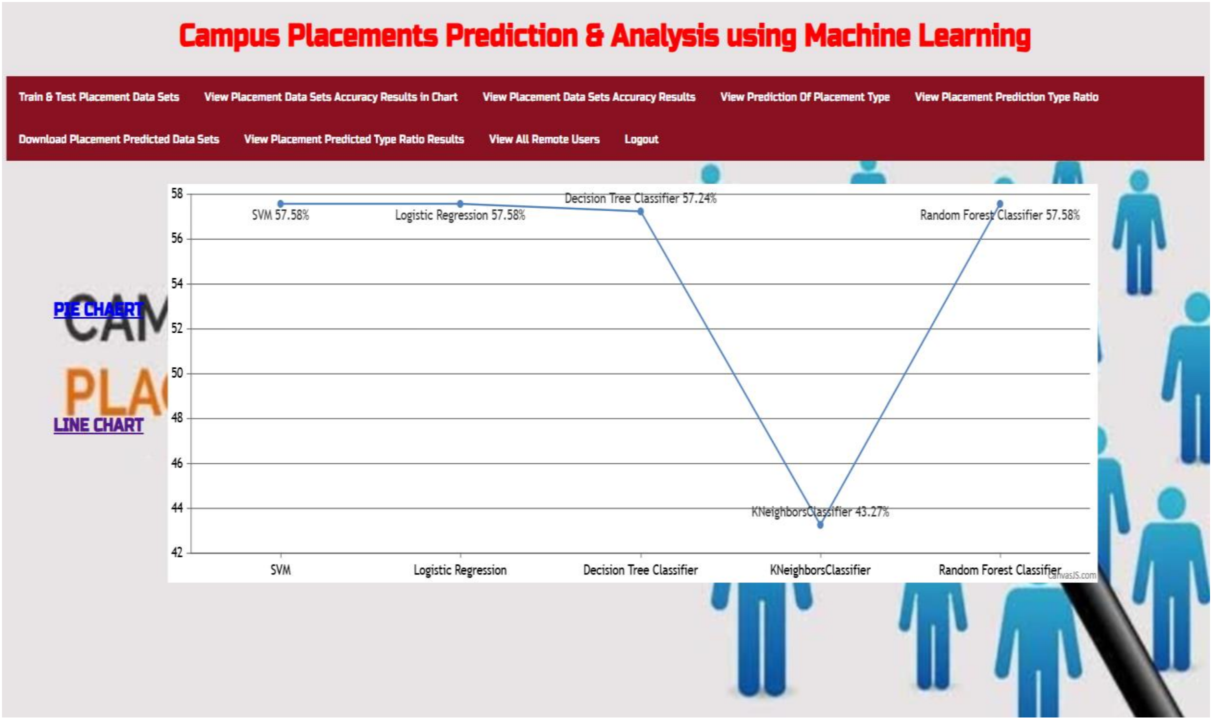


Figure 5.7: View Placement Data Sets Accuracy Results of Campus placements prediction and analysis using machine learning.

DESCRIPTION:

Displays analyzed placement datasets, categorizing them based on placement type into labels such as placed and non-placed. It visually represents the classification results, allowing for an intuitive understanding of the distribution of various prediction types. This aids in monitoring and assessing the prevalence of harmful or constructive communication patterns within a dataset.

5.8 VIEW PLACEMENT PREDICTION TYPE RATIO:

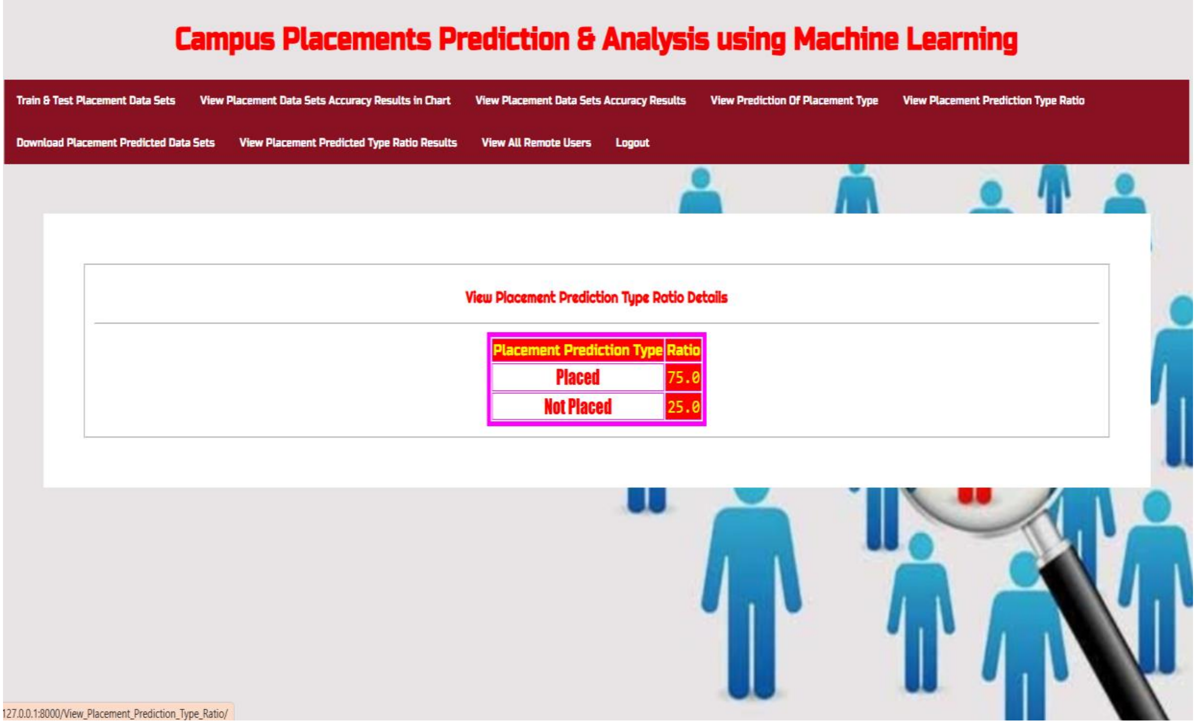


Figure 5.8: View Placement Prediction Type Ratio of Campus placements prediction and analysis using machine learning.

DESCRIPTION:

The Placement Prediction Type Ratio in this model visually represents the classification of placement type into categories like placed and non-placed using bar and pie charts. The system analyzes placement data and determines the percentage of each type, with an example breakdown showing 75.00 placed, 25.00 non-placed. This analysis helps in knowing placement type easily, allowing.

5.9 VIEW PLACEMENT PREDICTED TYPE RATIO RESULTS REPRSENTATION:

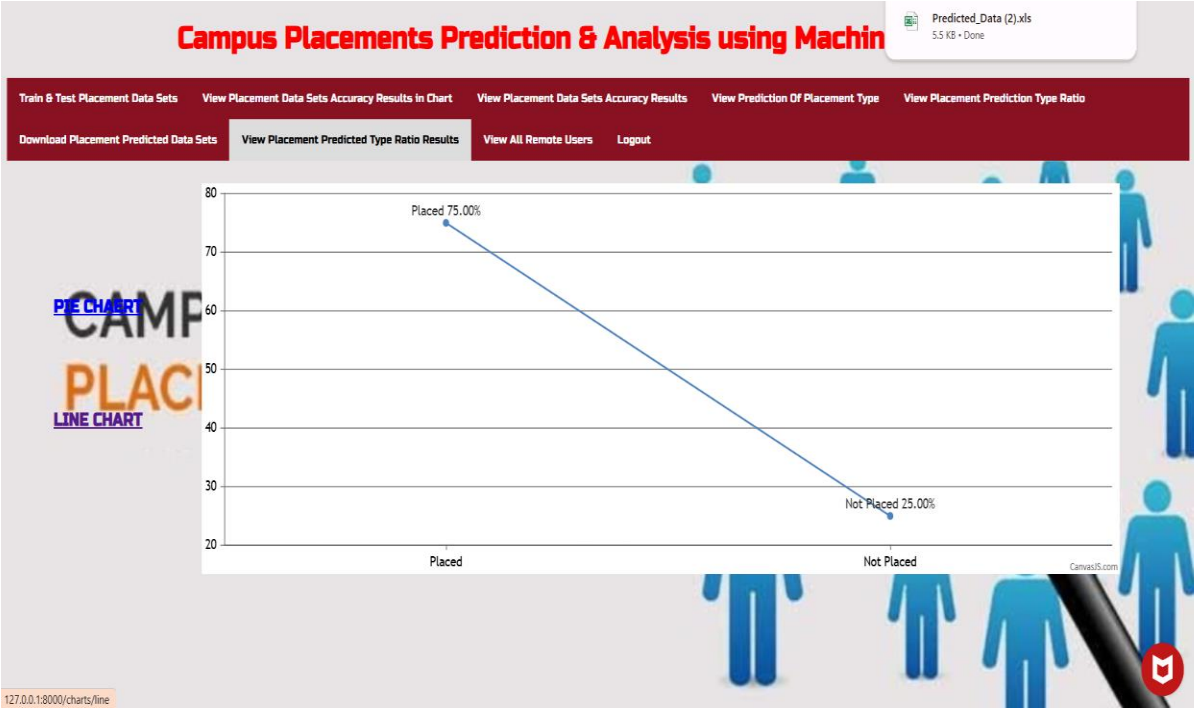


Figure 5.9: View Placement Prediction Type Ratio results representation of Campus placements prediction and analysis using machine learning.

DESCRIPTION:

The line chart in Placement Prrediction illustrates the distribution of different Pacement categories based on prediction analysis. It highlights that placed students accounts for 75.00%, followed by non-placed students accounts, each comprising 25.00%. This visual representation helps in understanding the placement prediction types across the data, aiding in better understanding of predicted placement data.

5.10 VIEW PREDICTION OF PLACEMENT TYPE:

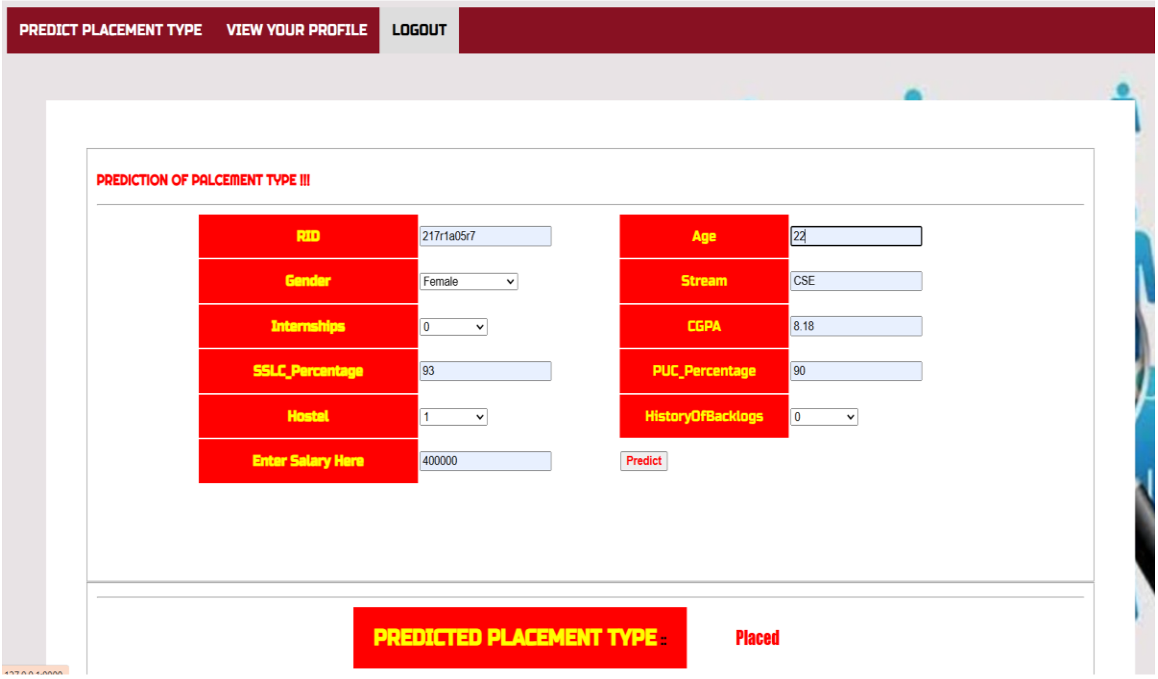


Figure 5.10: View Placement Prediction Type results representation of Campus placements prediction and analysis using machine learning.

DESCRIPTION:

The line chart in Placement Prrediction illustrates the distribution of different Pacement categories based on prediction analysis. It highlights that placed students accounts for 75.00%, followed by non-placed students accounts, each comprising 25.00%. This visual representation helps in understanding the placement prediction types across the data, aiding in better understanding of predicted placement data.

6. VALIDATION

6. VALIDATION

The validation of this project primarily relies on extensive testing and well-defined test cases to ensure the accuracy and effectiveness of the inappropriate content detection system. The testing process involves multiple stages, including dataset validation, model performance evaluation, and real-world testing. By implementing a structured validation approach, we can ensure that the system consistently delivers high accuracy in detecting inappropriate content while minimizing false positives and false negatives.

6.1 INTRODUCTION

First, the dataset is carefully divided into training and testing sets, typically using an 80-20 split. In campus placement prediction using machine learning, validation is crucial to assess the effectiveness of different classification algorithms such as Support Vector Machine (SVM), Random Forest, Logistic Regression, Decision Trees, and K-Nearest Neighbors (KNN). Each of these models has unique characteristics and performance behaviours, so validating their predictions on real-world data ensures that the most suitable model is chosen. This typically involves splitting the dataset into training and testing sets or using k-fold cross-validation to evaluate each model’s ability to generalize. For example, SVM is effective in high-dimensional spaces, Logistic Regression provides probabilistic outputs, Random Forest and Decision Trees are good at handling non-linear data and categorical variables, while KNN is simple and works well with smaller datasets.

To compare these models, classification metrics like accuracy, precision, recall, F1-score, and confusion matrix are used. These metrics reveal not only how often the model is correct but also how well it handles class imbalance (e.g., if more students are placed than not placed). Additionally, feature importance from Decision Trees and Random Forests can help identify key factors influencing placements, such as GPA or internship experience. Through consistent validation across all models, we can select the one that offers the best balance between interpretability and accuracy, ensuring reliable predictions and insights for career planning and institutional analysis.

6.2 TEST CASES

6.2.1 UPLOADING DATASET

Test case ID	Test case name	Purpose	Test Case	Output
1	Uploads Dataset.	Use it for placement prediction.	The user uploads the Dataset, on which the placement is predicted.	Dataset successfully loaded.

6.2.2 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output
1	Classification test 1	To check if the prediction model performs its task	Dataset csv file is selected	Preprocessed done
2	Classification test 2	To check if the prediction model performs its task	Inappropriate-Dataset csv file is selected.	Inappropriate Dataset csv file

7. CONCLUSION AND FUTURE ASPECTS

7 CONCLUSION AND FUTURE ASPECTS

In conclusion, the project has successfully achieved its core objectives, demonstrating the effective application of machine learning techniques in predicting campus placements. The systematic implementation of data preprocessing, model training, and evaluation has led to valuable insights into student employability and the key factors influencing placement outcomes. This project has not only enhanced decision-making processes for institutions but also empowered students with data-driven guidance.

7.1 PROJECT CONCLUSION

The Campus Placement Prediction System is designed to analyze and predict student placement outcomes using machine learning techniques applied to structured academic and demographic data. Leveraging models such as Logistic Regression, Decision Trees, Random Forest, and Support Vector Machines, the system evaluates key attributes—such as academic performance, skill set, internships, and certifications—to determine the likelihood of a student being placed. This approach not only increases prediction accuracy but also helps educational institutions and students identify critical factors influencing placement success.

By integrating data preprocessing, feature selection, and model evaluation techniques, the system ensures reliable prediction performance while addressing data imbalance and noise issues. The use of visualization tools provides stakeholders with clear insights into placement trends, model accuracy, and influential parameters. These statistical insights aid in strategic decision-making for curriculum enhancement, skill development, and targeted training programs.

Furthermore, the system is scalable and adaptable to dynamic educational environments, making it suitable for real-time analysis and longitudinal tracking of student progress. With potential integration of advanced models like neural networks and ensemble methods, as well as real-time dashboards and feedback mechanisms, the system can evolve into a comprehensive placement analytics solution. This predictive framework plays a vital role in bridging the gap between academic preparation and industry expectations, ultimately enhancing the employability of students and optimizing recruitment strategies for institutions.

7.2 FUTURE ASPECTS

The future aspects of the of this project offer promising opportunities. Future enhancements will focus on expanding the dataset, incorporating real-time data, and integrating advanced algorithms such as deep learning and ensemble models for higher accuracy. Additionally, developing an interactive dashboard and deploying the model as a user-friendly web application could further increase accessibility and usability. These innovations will elevate the project's impact, making it a vital tool for academic institutions, students, and recruiters alike, and ensuring its long-term relevance and adaptability in the evolving educational and recruitment landscape.

8. BIBLIOGRAPHY

8 BIBLIOGRAPHY

8.1 REFERENCES

1. S. K. Yadav and S. Pal, "Data Mining Applications: A Comparative Study for Predicting Student's Performance," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 2, no. 6, pp. 2278–3075, 2013.
2. M. A. Jantan, A. R. Hamdan, and Z. A. Othman, "Knowledge Discovery Techniques for Predicting Graduate Employability," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 9, no. 1, 2011.
3. P. Cortez and A. Silva, "Using Data Mining to Predict Secondary School Student Performance," *EUROSIS*, 2008.
4. J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed., Morgan Kaufmann, 2011.
5. Scikit-learn: Machine Learning in Python – Pedregosa et al., *Journal of Machine Learning Research*, 12, pp. 2825–2830, 2011.
6. V. Kumar and M. Bhardwaj, "Predictive Modeling of Student Placement using Machine Learning Techniques," *International Journal of Computer Applications*, vol. 180, no. 29, pp. 22–27, 2018.
7. Weka Software for Data Mining – University of Waikato.
8. A. Aurangzeb, "A Machine Learning Framework to Predict Campus Placement for Students," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 8, pp. 361–368, 2019.

8.2 GITHUB LINK

https://github.com/sindhuvurugonda568/Campus-Placements-prediction-and-analysis-using-machine-learning.git