Low Level Design (LLD)

Campus Placement Prediction

Revision Number: 1.0
Last date of revision: 30/07/2023

Dev Reshamiya

Document Version Control

Date Issued	Version	Description	Author
30 July 2023	1.1	First Draft	Dev Reshamiya

Contents

Abstract

The Placement of students is one of the most important objective of an educational institution. Reputation and yearly admissions of an institution invariably depend on the placements it provides it students with. That is why all the institutions, arduously, strive to strengthen their placement department so as to improve their institution on a whole. Any assistance in this particular area will have a positive impact on an institution's ability to place its students. This will always be helpful to both the students, as well as the institution. The main goal is to predict whether the student will be recruited in campus placements or not based on the available factors in the dataset. By analyzing historical placement data, this project identifies key factors that contribute to successful placements and those that hinder the process. Machine learning algorithms are employed to predict the likelihood of a student being placed based on various attributes such as academic performance, extracurricular activities, and skillset. These predictive models assist placement officers in identifying students who may require additional support and personalized guidance.

1 Introduction

1.1 Why this Low-Level Design Document?

The low-level design for the campus placement system is crucial for its successful implementation. It will provide a detailed plan for building the system that predicts whether a person will be placed and their expected salary based on various features. The design will encompass various aspects, including data collection, preprocessing, model building, and user interface development.

Data Collection:

- Specify the sources of data, including student profiles, academic records, and other relevant information.
- Define the data format and data storage mechanisms.

Data Preprocessing:

- Describe the steps for handling missing data, outliers, and data normalization.
- Define the encoding process for categorical variables and feature scaling.

Model Building:

- Select the appropriate machine learning models, such as logistic regression, linear regression, and random forest, for prediction tasks.
- Specify hyperparameter tuning and cross-validation strategies for model optimization.

User Interface Development:

- Design an intuitive and user-friendly interface for inputting student data.
- Implement functionality to visualize prediction results and display the accuracy of the models.

Integration and Deployment:

- Outline the process of integrating the frontend with the backend.
- Specify the server requirements for hosting the application.

Error Handling:

- Define error messages and responses for incorrect or missing inputs.
- Implement measures to handle server errors and prevent data leakage.

1.2 Scope

The scope of the campus placement system includes the development of a robust and accurate predictive model that can assess the likelihood of a student being placed in a company based on their academic and other relevant attributes. The system will also predict the expected salary for placed students. Additionally, the system will provide a user-friendly and intuitive interface for students and placement coordinators to input and access data. The scope further involves implementing error handling, security measures, and testing to ensure the system's reliability and data privacy. The system will cater to the needs of educational institutions and companies seeking to optimize their campus recruitment processes and enhance their decision-making.

1.3 Constraints

Data Availability: The success of the project relies on the availability of high-quality and relevant data related to student profiles, academic performance, and placement outcomes. Limited or inaccurate data could impact the model's accuracy and effectiveness. 2. Model Accuracy: The machine learning models used for predicting placement status and salary are based on historical data and are subject to certain limitations. The accuracy of the models might not be 100%, and there could be cases where predictions may not align with actual outcomes. 3. Ethical Considerations: The project involves handling sensitive information about students, such as academic performance and personal details. Ensuring data privacy and complying with ethical guidelines are critical aspects that need to be strictly followed. 4. Deployment and Hosting: Deploying the web application requires hosting the backend server and the frontend on suitable platforms. This could incur costs, and ensuring smooth hosting and maintenance are important considerations. 5. User Engagement: Encouraging students to use the platform and provide accurate information is essential for the success of the project. Ensuring a user-friendly interface and clear communication of the benefits are crucial to drive user engagement. 6. Scaling and Performance: As the number of users and data increases, the web application should be capable of handling the load efficiently. Ensuring scalability and optimizing the performance of the application are important factors to consider

1.4 Risks

1 Data Quality and Integrity: The accuracy and quality of the data used for building the predictive models are crucial. Incomplete or inaccurate data can lead to misleading predictions and affect the overall performance of the system.
2 Model Overfitting: There is a risk of overfitting the models to the training data, which might lead to poor generalization on unseen data. It is essential to carefully tune hyperparameters and apply regularization techniques to mitigate this risk.
3 Ethical and Bias Concerns: Predictive models may unintentionally introduce biases based on gender, race, or other sensitive attributes. Ensuring fairness and avoiding discriminatory practices is vital to maintaining ethical standards in the campus placement process.
4 Model Interpretability: Complex machine learning models may lack transparency, making it challenging to understand the factors influencing predictions. It is essential to strike a balance between model accuracy and interpretability to gain stakeholders' trust.

4.1 Out of Scope

- Campus Recruitment Drives: The project does not involve organizing or managing campus recruitment drives. It focuses solely on predicting students' placement status and expected salary based on historical data.
- Placement Process Management: The project does not handle the overall management of the placement process, such as scheduling interviews, coordinating with companies, or conducting pre-placement talks.
- Student Counseling: Providing individualized career counseling or advice to students based on the prediction results is not within the scope of this project.

5 Technical specifications

2.1 Dataset

Disease	Finalized	Source
Placement	yes	https://www.kaggle.com/c/ml-with-python-course-project/data

2.1.1 Dataset overview

- The campus placement dataset is a collection of information related to students' academic and placement records. It includes various attributes that provide insights into the students' educational background, performance, and the final outcome of their campus placements. The dataset is used to develop predictive models to determine the likelihood of a student getting placed and to estimate their expected salary.

- Features: Some of the features in the dataset may include:

  - Gender: The gender of the student (e.g., male or female).
  - SSC Percentage: The percentage marks obtained in the Secondary School Certificate (SSC) exam.
  - HSC Percentage: The percentage marks obtained in the Higher Secondary Certificate (HSC) exam.
  - Degree Percentage: The percentage marks obtained in the undergraduate degree program.
  - Specialization: The field of specialization in the undergraduate program (e.g., Engineering, Commerce, Science).
  - Work Experience: Whether the student has prior work experience (yes or no).
  - MBA Percentage: The percentage marks obtained in the Master of Business Administration (MBA) program.
  - Status: The outcome of the campus placement process (placed or not placed).
  - Salary: The salary offered to the student upon placement.

2.2 Predicting Placement

- The system displays a form to input values.
- User fills the form and submit
- System Preprocess the data
- Prediction are made using model
- Prediction are displayed in the interface

2.3 Logging

- The System identifies at what step logging required
- The System logs each and every system flow.

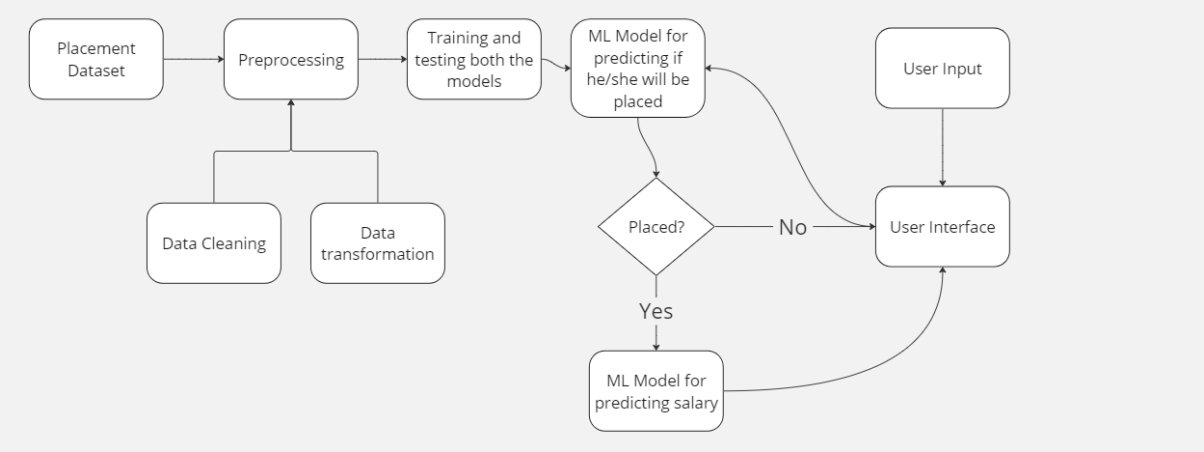
6 Technology stack

Front End	HTML/CSS/JS/React
Backend	Python Flask

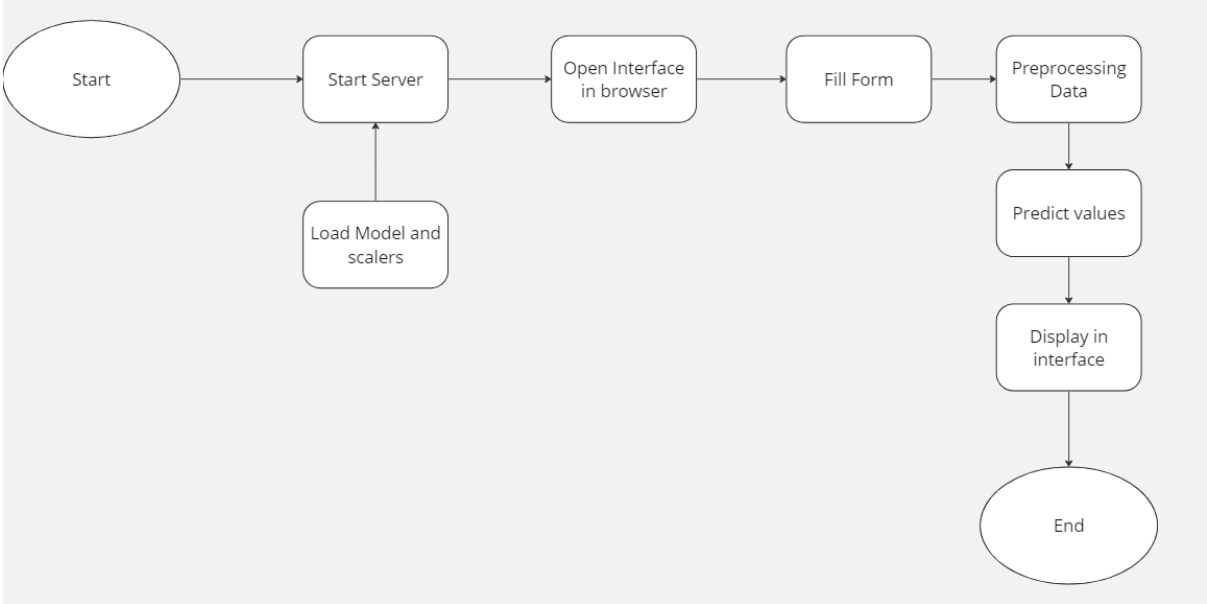
7 Proposed Solution

The proposed solution aims to develop a comprehensive campus placement prediction system that utilizes machine learning algorithms to forecast a student's likelihood of getting placed and estimate their expected salary upon successful placement. The system will be designed with Python's Flask framework for the backend and ReactJS for the frontend, ensuring an efficient and interactive user interface. Preprocessing steps like outlier removal, feature engineering, and standard scaling will be employed to enhance model performance. The trained logistic regression model will predict placement status, while the linear regression model will forecast salary. The web application will offer a simple form-based user interface, allowing students to input their academic and personal details. Upon submission, the models will process the data and display the results, providing valuable insights for both students and recruiters.

8 Model training/validation workflow



9 User I/O workflow



10 Key performance indicators (KPI)

1 Prediction Accuracy: The accuracy of the placement status and salary predictions made by the app's machine learning models is a crucial KPI. It measures how well the app's models are performing in correctly predicting whether a candidate will be placed and estimating their expected salary.

2 Response Time: The time taken by the app to respond to user actions, such as form submission and displaying prediction results, is an essential KPI. Lower response times indicate better app performance and a more enjoyable user experience.

3 User Engagement: User engagement metrics, such as the number of form submissions and user interactions with the app, help measure how actively users are using the app and how interested they are in the prediction results.

4 Error Rate: The percentage of errors encountered during the app's operation is an important KPI. Lower error rates indicate better app stability and reliability.