High-Level Design (HLD)

Campus Placement Prediction

Written By	TRISHIT NATH THAKUR
Version	1.1
Date	25-05-2023

Version	Date	Author	Comments
1.0	10/6/23	TRISHIT	Initial HLD-V 1.0
1.1	18/6/23	TRISHIT	Updated HLD-V 1.1

Abstract

One of the most crucial goals of an educational institution is student placement. An institution's reputation and yearly admissions are inextricably linked to the placements it offers its students. Institutions work very hard to place students in appropriate positions. The institution will always benefit from this. By examining the data gathered from students from the previous year, the goal is to estimate where the students will be placed for the upcoming academic year. This model is suggested along with an algorithm to make the same prediction. The institution for which forecast will be made has gathered the data by using the appropriate data pre-processing procedures. Utilizing the Support Vector Machine [SVM] technique, this model was created. This algorithm, which is based on the dataset, independently predicts the outcomes. We then compare the algorithm's effectiveness. This strategy will assist the placement cell in concentrating on prospective students and assisting them in strengthening their technical and interpersonal abilities.

1. Introduction

1.1 Why this High-Level Design Documents?

The purpose of this High-Level Design(HLD) Documents is to add necessary details to the current project description to represent a suitable for coding. This document is also intended to help detect contradictions before coding. And can be used as a reference manual for how the modules interact at a high level.

The HLD will be :

- Present all of the design aspects and define them in detail.
- Describe the user interface being implemented.
- Describe the needed Python libraries for the coding.

- Describe the performance requirements.
- Include design features and the architecture of the project.
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application Compatibility
    - Resource Utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture(layers), application flow (Navigation), and technology architecture, The HLD uses non-technical and mildly-technical terms which should be understandable to the administrators of the system

2. General Description

2.1 Product Perspective

The Campus Placement Prediction is an ML-based Web Application that Is able to predict placement status of students based past records. It will tell us whether the students are placed on not placed.

2.2 Problem Statement

The placement prediction model considers only academic performances of the students so that the prediction of the student getting placed or not can be done. We cannot consider the placement of students just by their academic performances because some students may be good at aptitude, technical and communication skills due to their low score in their academic that may tend to be their drawback. For predicting the placement of a Student needs parameters like cgpa, logical and technical skills Academic performances may be important but the model is design to predict the placements based on the parameters of the students..

2.3 Proposed Solution

We have designed a models for checking prediction of student getting placed or not in a campus drive this model will help both student and institutions for preparing well in advance for campus recruitment. The main objective of this model is to know the capability of the student and where he stands by predicting the probability of getting placed. This also help the students and institutions to improve performance of the potential students. This model will consider the academic history of the student such as percentage as well as their domains and specializations which are tested by companies. We used SVM algorithm on the student's data gathered from the institution of previous year. This model take scores of student in secondary education along with academic CGPA of both UG and PG.in the technical education till date and also some parameters which adds weightage to kick start the career.

FURTHER REQUIREMENTS:Machine learning tools help placement team members by tracking a candidate's journey throughout the interview process and helping out to speed up the process of getting feedback. After evaluating, a feedback will be given to candidates in order to improve their competency level or to specify the area of improvemen

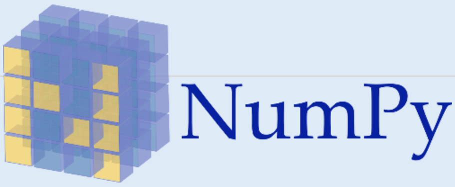
2.4 Data Requirements

File descriptions

- train.csv - the training set
- test.csv - the test set
- SampleSubmission.csv - a sample submission file in the correct format Data fields
- gender - sex of the student
- secondary education percentage-marks obtained in secondary education
- higher secondary percentage-marks obtained in higher secondary education
- degree percentage-marks obtained in degree
- Under-graduation(Degree-type)-Field of degree education
- Work-experience
- Employability-test-package
- specialisation-field ofstudy

2.5 Tool Used

The programming language is Python that is used here, also we will use some other python-based libraries like, for ml, we will use Scikit-Learn library, for data manipulation we will use pandas, for numerical computation Numpy, for custom APIs creation Flask web frameworks. Pycharm is used as python IDE for all modular coding and custom APIs creation. And storing all code files for publically available we will use GitHub.



pandas

ANACONDA

GitHub

2.6 Constraints

The System should be user-friendly, the user should get all proper messages while using the web app. He/she also should get a proper error message if he/she has done something wrong On the web-app page. All the errors and results should be delivered in the easiest possible way and all the buttons are going to insert on the webpage should be labeled properly, so the user did not get confused to use the system.

2.7 Assumptions

The main objective is to implement a system that will produce the placement status of students.

3. Design Details

Data gathering the sample data has been collected from our college placement department which consists of all the records of previous year's students. The dataset collected consist of over 1000 instances of students.

Preprocessing Data preprocessing is a technique that is used to convert raw data into a clean dataset. The data is gathered from different sources is in raw format which is not feasible for the analysis. Pre-processing for this approach takes 4 simple yet effective steps.

Attribute selection some of the attributes in the initial dataset that was not pertinent (relevant) to the experiment goal were ignored. The attributes name, roll no, credits, backlogs, whether placed or not, b.tech %, gender are not used. The main attributes used for this study are credit, back-logs, whether placed or not, b.tech %.

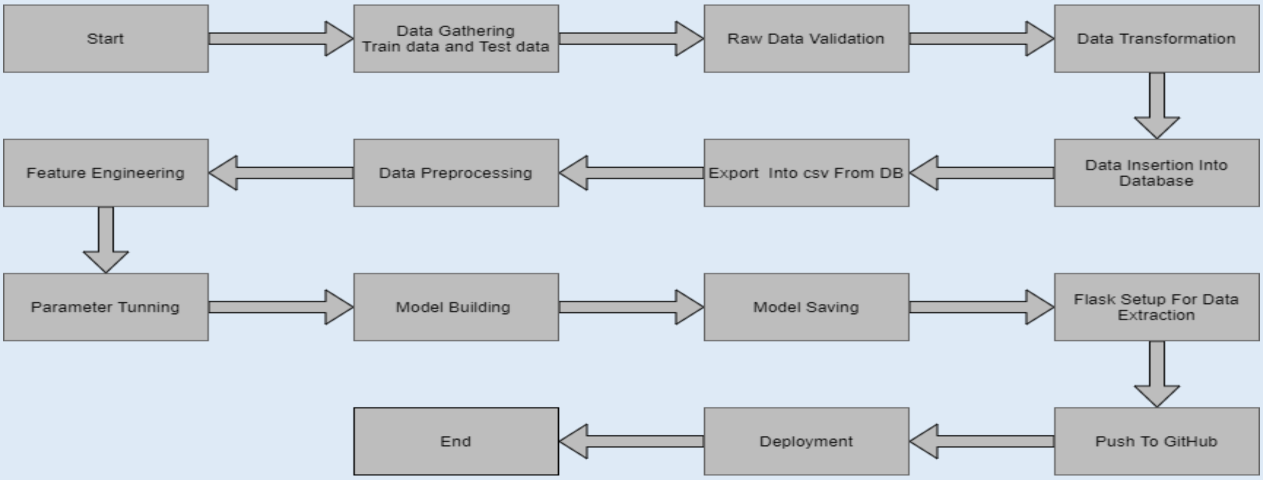
Cleaning missing values in some cases the dataset contain missing values. We need to be Page equipped to handle the problem when we come acrossthem. Obviously you could remove the entire line of data but what if you're inadvertently removing crucial information? After all we might not need to try to do that. One in every of the foremost common plan to handle the matter is to require a mean of all the values of the same column and have it to replace the missing data. The library used for the task is called Scikit Learn preprocessing. It contains a class called Imputer which will help us take care of the missing data.

Training and Test data splitting the Dataset into Training set and Test Set Now the next step is to split our dataset into two. Training set and a Test set. We will train our machine learning models on our training set, i.e. our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to examine how accurately it will predict. A general rule of the thumb is to assign 80% ofthe dataset to training set and therefore the remaining 20% to test set.

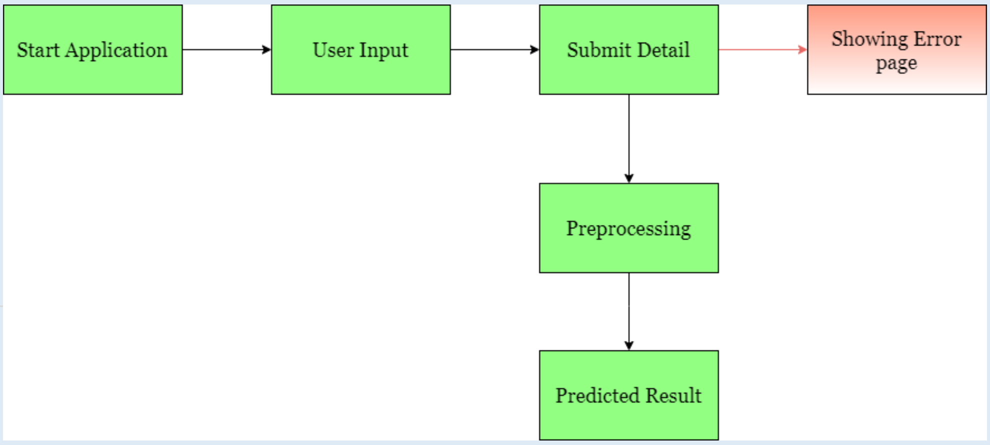
Feature scaling is the final step of data preprocessing is feature scaling. But what is it? It is a method used to standardize the range of independent variables or features of data. But why is it necessary? A lot of machine learning models are based on Euclidean distance. If, for example, the values in one column (x) is much higher than the value in another column (y), (x2-x1) squared will give a far greater value than (y2-y1) squared. So clearly, one square distinction dominates over the other square distinction. In the machine learning equation.

3.1 Process Flow

We will be using following process flow for this project. The process will be based on modular coding i.e. use of oops concepts to build the entire project from start to end.



3.2 Deployment Process



3.3 Error Handling

If any error occurred in the processing way then the error message should be shown to the user in a completely non-technical way that can be understandable by any person. And Meaningful error message should be shown, so the user can spot his mistake and rerun the process with improvement. All the errors that are will occur should be handled properly. And we have to log every error for our application and have to manage the same.

4. Performance

The Campus Placement Prediction is dependent on machine-learning algorithms. We will train various ml algorithms and will find the best fitting algorithm for predicting the target. Our system performance will be based on the data we are going to feed to the algorithms. And the performance will depend on the finalized model. and the web application and the deployment server. With all of these components, our program should run properly.

4.1 Reusability

The code and the module are created during the time of building the project should maintain all coding guidelines and full project code is written in a Modular fashion. Our system should have the flexibility to work properly from any location. And it should handle any improper input value from the user and should give a meaningful error message so the user can correct his/her mistake and enter valid input to get the result. And the system should be reusable in every manner with different types of inputs values that are all are it has been trained.

4.2 Application Compatibility

The different libraries and python programming languages are used to build the system. Every library has its own functionality and it should work properly with our fluctuate system. Flask will be used for making the web APIs and HTML/CSS will be used to make the web application. All the components of the application should work properly and it should produce a result without any interpretation.

4.3 Resource Utilization

Our application should utilize the given resource properly and it should use a minimal amount of internet to work and call the APIs on the Web page. Our system should not use much amount of computational resources hence it will make the application slow. Our application will be deployed cloud platform and it should utilize the resource given on the cloud and work properly.

5. Deployment

For the deployment process, we will using Heroku cloud platforms for hosting our application. The cloud platform will run the system and it will give the flexibility to use our application globally.

HEROKU

6. Conclusion

The Campus Placement Prediction is about to help Educational instituions to predict placement status for there students. It can help them to strengthen their placement department so as to improve their institution on a whole. We have a past record of students realted to the their educational background. We will analyze the past data and will build an ml model that can identify the internal pattern and be able to predict the target value or the placement status of the student in the future.