

ASSIGNMENT:8

Low-Level Design

Campus Placement Prediction

Introduction

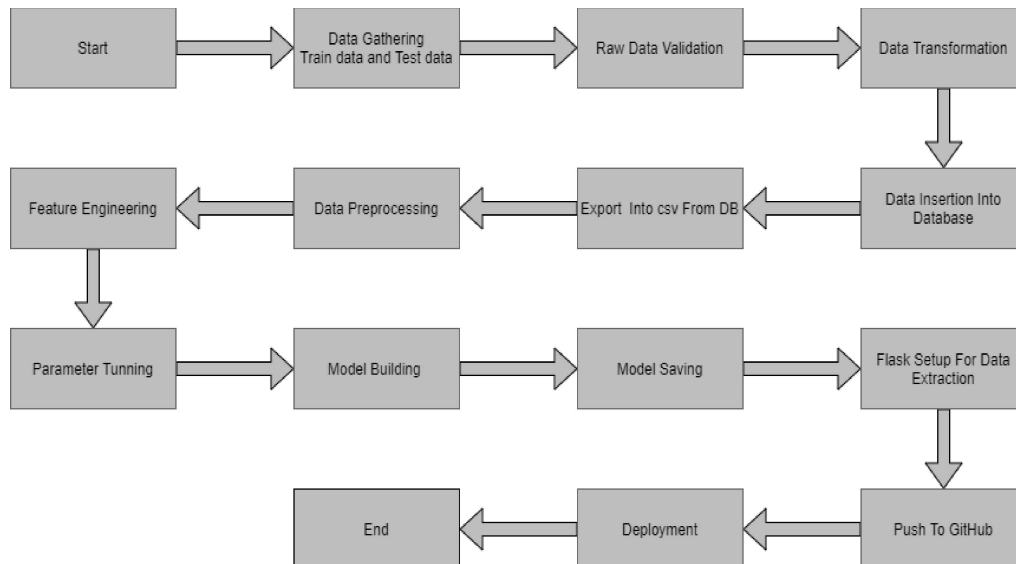
1.1 What is Low-Level Design Document.

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for '**Campus Placement Prediction**'. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2 Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

Architecture :



2. Architecture Description

2.1 Data Description

Given is the variable name, variable type, the measurement unit, and a brief description. The concrete compressive strength is the regression problem. The order of this listing corresponds to the order of numerals along the rows of the database.

Name	Data Type	Measurement
Gender	Integer	Gender of student
ssc_p	Float	Ssc percentage
ssc_b	Object	Ssc board
hsc_p	Float	Hsc percentage
Hsc_b	Object	Hsc board
Hsc_s	Object	Hsc stream
Degree_p	Float	Degree percentage
Degree_t	Object	Graduation stream
Workex	Object	Are they having any work experience
Etest_p	Float	Online test percentage
Specialisation	Object	Specialization the choosed in Mba
Mba_p	Float	Mba percentage
Status	Object	Are they placed or not placed. This the outcome column.

2.2 Data Gathering

Data source: <https://www.kaggle.com/competitions/ml-with-python-course-project/data>

Train and Test data are stored in .csv format.

2.3 Raw Data Validation

After data is loaded, various types of validation are required before we proceed further with any operation. Validations like checking for zero standard deviation for all the columns, checking for complete missing values in any columns, etc. These are required because The attributes which contain these are of no use. It will not play role in predicting the status of placement.

Like if any attribute is having zero standard deviation, it means that's all the values are the same, its mean is zero. This indicates that either the student placed or not that attribute will remain the same. Similarly, if any attribute is having full missing values, then there is no use

in taking that attribute into an account for operation. It's unnecessary increasing the chances of dimensionality curse.

2.4 Data Transformation

Before sending the data into the database, data transformation is required so that data are converted into such form with which it can easily insert into the database. Here, the columns which have any missing values are filled in both the train set as well as the test set with supported appropriate data types.

2.5 Data Preprocessing

In data preprocessing all the processes required before sending the data for model building are performed. Here, I removed all those columns whose standard deviation is zero because they are not contributing anything and those columns also which are of no use in predicting the status of Placement. After that I separate label and features columns from the train and test data.

2.6 Feature Engineering

After preprocessing it was we saw that there are few categorical columns present in the data. For converting the categorical column, I have created the column transformer object and inside it I performed one hot encoding on categorical columns. Afterwards I save the transformer object so that it can be used for transforming the test and prediction data in same way as train data. I have converted the target column categorical values into numerical using label encoder.

2.7 Parameter Tuning

I have used sklearn pipeline library with grid search cv to perform hyper parameter tuning. I have used different algorithms in pipeline and set different parameters for their important attributes.

2.8 Model Building

After doing all kinds of preprocessing operations mention above and performing scaling and hyperparameter tuning, the data set is passed into Logistic Regression, Decision Tree Classifier, Random Forest Classifier, SVC. It was found that SVC performs best with the highest accuracy score equals 0.88. So 'Random forest classifier' performed well in this problem

2.9 Model Saving

Model is saved using pickle library.

2.10 GitHub

The whole project directory will be pushed into the GitHub repository.

2.11 Deployment

The cloud environment was set up and the project was deployed from GitHub into the Streamlit cloud platform.

App link- <https://mihirkudale-campus-placement-prediction-app-s4vle8.streamlit.app/>

3. Unit Test Cases.

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1.Application URL is accessible 2.Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether a user is able to see input fields while opening the application	1.Application is accessible 2.The user is able to see the input fields	Users should be able to see input fields on logging in
Verify whether a user is able to enter the input values.	1.Application is accessible 2. The user is able to see the input fields	The user should be able to fill the input field
Verify whether a user gets predict button to submit the inputs	1. Application is accessible 2.The user is able to see the input fields	Users should get Submit button to submit the inputs
Verify whether a user is presented with recommended results on clicking submit	1. Application is accessible 2. The user is able to see the input fields. 3. The user is able to see the submit button	Users should be presented with recommended results on clicking submit
Verify whether a result is in accordance with the input that the user has entered	1. Application is accessible 2. The user is able to see the input fields. 3. The user is able to see the submit button	The result should be in accordance with the input that the user has entered

High-Level Design

Campus Placement Prediction

Abstract

The project is about building a system that can predict placement status of students studying in various institutions. By analysing past student records according to placement status. Educational institutions kept records of their students to forecast future placement status. Yearly admission varies proportionately with the placement that it provide to it's students. That's why every institution is looking forward to strengthen its placement cell. This will always be helpful for both the students, as well as the institution.

1. Introduction

1.1 Why these 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

1.3 Definition

TERM	Description
DB	A database is information that is set up for easy access, management and updating . Computer databases typically store aggregations of data records or files that contain information, such as sales transactions, customer data, financials and product information.
ML	Machine Learning
API or APIs	APIs are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols.

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 on past records. It will tell us whether the students are placed or not placed.

2.2 Problem Statement

To build a system that will be able to take information about a student and can predict if he will be placed or not placed. We have to build an application and that will be able to produce results.

2.3 Proposed Solution

We will perform EDA to find the important relation between different attributes and will use a machine-learning algorithm to predict the placement status. The client will be filled with the required feature as input and will get results through the web application.

2.5 Data Requirements

The data required for building the project is already available on the dashboard. The Campus Placement Prediction data is recorded with many students' information along with educational career description.

This data set consists of Placement data of students in our campus. It includes secondary and higher secondary school percentage and specialization. It also includes degree specialization, type and Work experience and salary offers to the placed students.

Source: <https://www.kaggle.com/benroshan/factors-affecting-campus-placement> or <https://www.kaggle.com/c/ml-with-python-course-project/data>

Attribute Information: sl_no Serial Number

gender Gender- Male='M', Female='F' (male=0, female=1)

ssc_p Secondary Education percentage- 10th Grade

ssc_b Board of Education- Central/ Others
hsc_p Higher Secondary Education percentage- 12th Grade
hsc_b Board of Education- Central/ Others
hsc_s Specialization in Higher Secondary Education
degree_p Degree Percentage
degree_t Under Graduation(Degree type)- Field of degree education
workex Work Experience
etest_p Employability test percentage (conducted by college)
specialisation Post Graduation(MBA)- Specialization
mba_p MBA percentage
status Status of placement- Placed/Not placed
salary Salary offered by corporate to candidates

2.6 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 we use frameworks.

2.7 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.

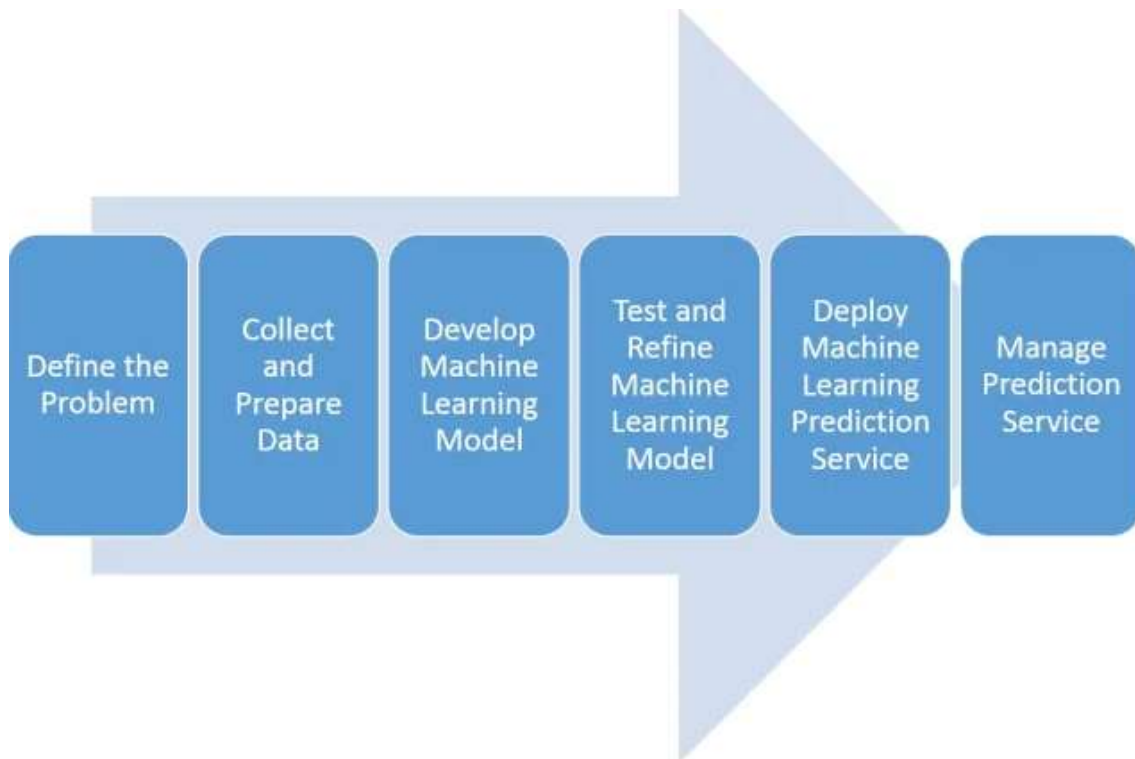
2.8 Assumptions

The main objective is to implement a system that will produce the placement status of students and help them to choose the better college for them.

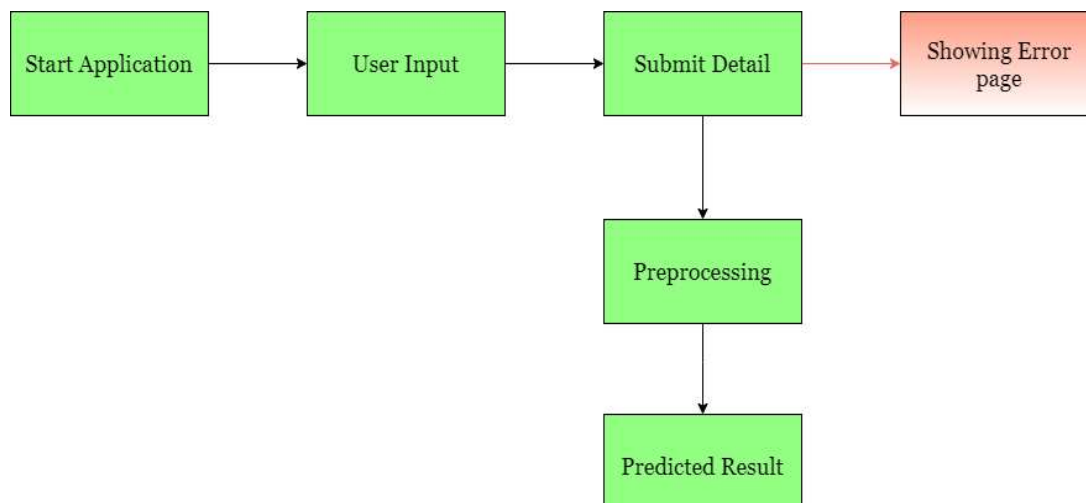
3. Design Details

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.

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. model training is also very important to improve the performance.

4.1 Reusability

The **code** written **and** the components **used** should have the **ability** to be **reused with no problems**.

4.2 Application Compatibility

The different **components for this project** will be **using** Python as an interface between **them**. Each **component will** have **its** own **task** to perform, and it **is** the job **of the** Python to ensure proper transfer of information.

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.

5. Deployment

The cloud environment was set up and the project was deployed from GitHub into the Streamlit cloud platform.

6. Conclusion

The Campus Placement Prediction is about to help Educational institutions to predict placement status for their students. It can help them to strengthen their placement department so as to improve their institution on a whole.