Low Level Design

Campus Placement Prediction

Written By	Akarsh Rastogi
Document Version	0.3
Last Revised Date	18-Mar-2024

Document Control

Change Record:

Version	Date	Author	Comments
0.1	10-March -2024	Akarsh Rastogi	Introduction & Architecture defined
0.2	15-March -2024	Akarsh Rastogi	Architecture & Architecture Description appended and updated
0.3	18-March-2024	Akarsh Rastogi	Unit Test Cases defined and appended

Reviews:

Version	Date	Reviewer	Comments

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

1. Introduction

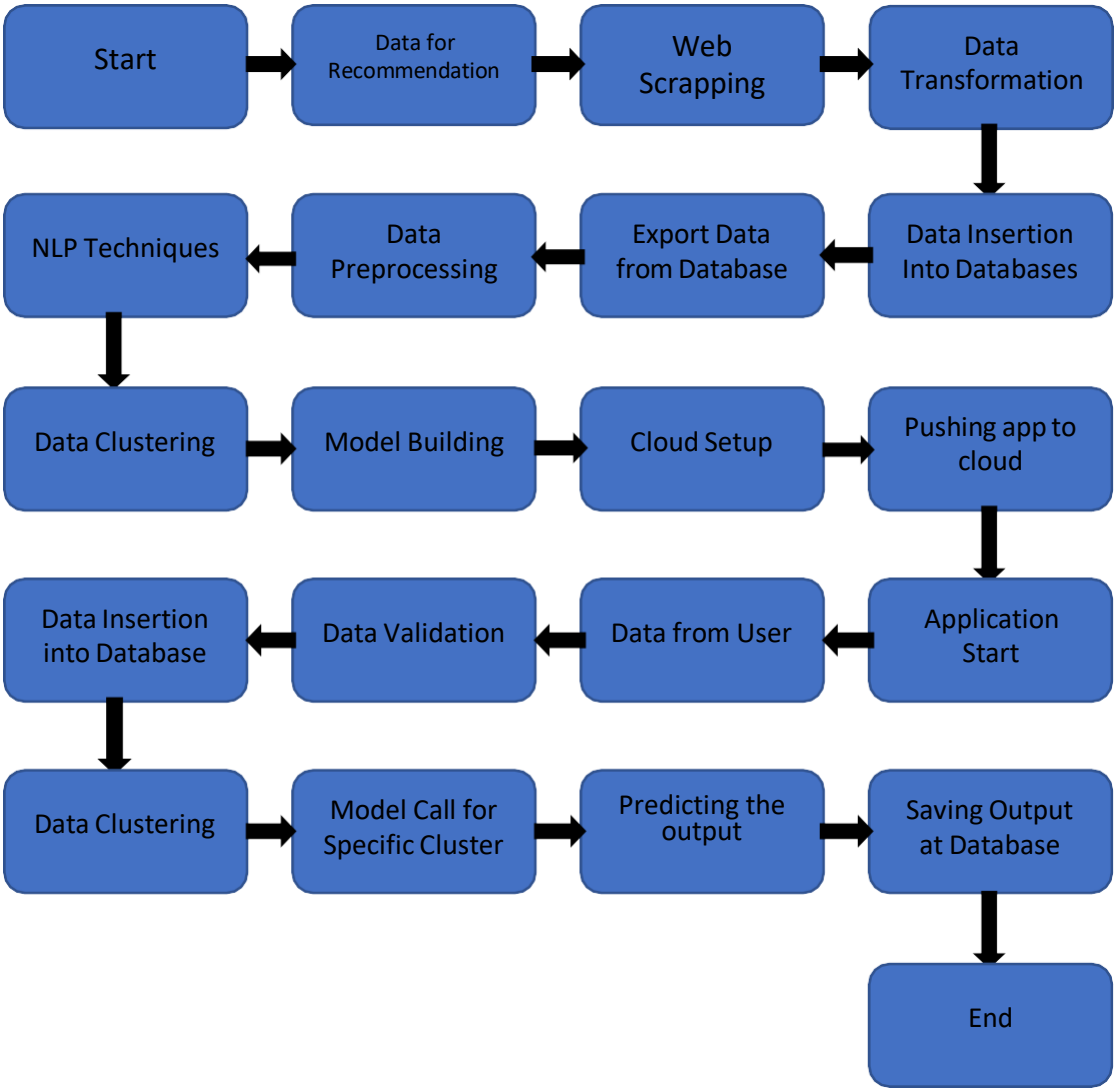
What is Low-Level design document?

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.

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

2. Architecture



3. Architecture Description

Data Description

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 of study

Data Transformation

In our dataset a lot of categorical values are present, we transform those attributes into numerical values using one hot encoding. A one hot encoding is a representation of categorical variables as binary vectors.

This first requires that the categorical values be mapped to integer values. Then, each integer value is represented as a binary vector that is all zero values except the index of the integer, which is marked with a 1.

Data Pre-processing

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 sl_no , ssc_b , hsc_b , salary are not used. The main attributes used for this study are -

hsc_s degree_t workex specialisation status gender ssc_p hsc_p degree_p etest_p mba_p

Cleaning missing values in some cases the dataset contain missing values. We need to be equipped to handle the problem when we come across them. 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.
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% of the dataset to
Training and Test data splitting the Dataset into Training set and Test Set Now the next step is training set and therefore the remaining 20% to test set.

Feature scaling 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.

Data Clustering

The campus placement activity is incredibly vital from institution point of view as well as student point of view. In this regard to improve the student's performance, a dataset has been analyzed and predicted using the classification algorithms like Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, SVC to validate the approaches.

Models used:

1. Logistic Regression: A statistical method used for predicting the probability of a binary outcome based on one or more predictor variables.

2. Decision Tree: A predictive model that maps features (or variables) to possible outcomes by splitting the data into branches based on conditions.

3. Random Forest: An ensemble learning technique that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees.

4. Gradient Boosting: An ensemble learning method that builds a series of weak learners (typically decision trees) sequentially, where each new learner corrects errors made by the previous one.

5. Support Vector Machine (SVC): A supervised learning algorithm that analyzes data and recognizes patterns, used for classification and regression analysis. It finds the hyperplane that best separates classes in high-dimensional space.

Model Building

After clusters are created, we will find the best model for each cluster. For each cluster, algorithms will be passed with the best parameters. The algorithms are applied on the data set and attributes used to build the model. The accuracy obtained after analysis for Logistic Regression algorithm is 88%, Random Forest is 76%, Gradient Boosting is 76%, Decision tree is 72% and for the SVC is 79%. Hence, from the above said analysis and prediction it's better if the LR algorithm is used topredict the placement results.

Data from User

- 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 of study is collected.

Data Validation

Here Data Validation will be done, given by the user

Data Clustering

The model created during training will be loaded, and clusters for the user data will be predicted.

Model Call for Specific Cluster

Based on the cluster number, the respective model will be loaded and will be used to predict/Recommend the data for that cluster.

Deployment

We will be deploying the model on HEROKU.

4 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 URLis accessed	1. Application URLis accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether the User is able to sign up in the application	1. Application is accessible	The User should be able to sign up in the application
Verify whether user is able to successfully login to the application	1. Application is accessible 2. User is signed up to the application	User should be able to successfully login to the application
Verify whether user is able to see input fields on logging in	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to see input fields on logging in
Verify whether user is able to edit all input fields	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should get Submit button to submit the inputs
Verify whether user is presented with recommended results on clicking Submit	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be presented with recommended results on clicking submit