

Campus Placement Assistant

A PROJECT REPORT

submitted by

Arjun R LDMC18CS030

To

The APJ Abdul Kalam Technological University
In partial fulfillment of the requirements for the award of the Degree

Of

Bachelors of Technology

In

Computer Science and Engineering



M Dasan Institute of Technology (MDIT)

Ulliyeri P.O, Kozhikode-673620

Department of Computer Science and Engineering

MAY 2021

DECLARATION

I undersigned declare that the project report “Campus Placement Assistant”, submitted for the partial fulfillment of the requirements for the award of the degree of the Bachelors of Technology under the university of the APJ Abdul Kalam Technological University, Kerala is a bonafide work, done by us under the supervision of the Ms.Nithya V P (Asst Prof). This submission represents our ideas in our own words and where ideas or words of others have been included; I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and the university and can also be evoke penal action from the sorce which has thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other university.

Place :

Date :

ARJUN R

M.DASAN INSTITUTE OF TECHNOLOGY

ULLIYERI

Approved by AICTE, New Delhi & Affiliated to APJ Abdul Kalam
Technological University

(KTU)



CERTIFICATE

This is to certify that the report entitled “CAMPUS PLACEMENT ASSISTANT” by “ARJUN R” (LDMC18CS030) to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelors of Technology in Computer Science Engineering is a bonafide of seminar carried by his under our guidance and supervision. This report in any form not been submitted to any other University or Institute of any purpose.

Project Guide
Ms. NITHYA V P
Asstistant Professor
Dept. of CSE

Project Coordinator
Ms. RAMITHA M A
Asstistant Professor
Dept. of CSE

Head of The Department
Ms. NITHYA V P
Asstistant Professor
Dept. of CSE

ACKNOWLEDGEMENT

I would like to thank THE ALMIGHTYs mercy towards us over the years; it goes by saying that, I would not be where I am today.

I would like to place on record, sincere thanks to all those who have contributed to the successful completion of this project.

I recollect with of gratitude the valuable guidance and the whole hearted support of the project guide Ms. Nithya V P (Asst. Prof. & Head of the department), project Coordinator Ms. Ramitha M A (Asst. Prof.) and CSE Department for their valuable suggestions throughout the project work.

I wish to express the deep sense of gratitude and thanks to our heartley Principal Mr. Mahessan P M and Director Mr. H. Ahinus for the excellent facilities and continual support provided during the course study and project work.

I thank all the staff members of Computer Science and Engineering Department for their valuable suggestions.

I also thank all beloved friends for their encouragement, timely help and valuable suggestions.

ABSTRACT

Each student or job aspirant, dreams to have best career. The campus placement assistant lets students to get a sense of where they are standing and what to do to ensure a decent selection . The placement predictor system helps in predicting whether a student will get placed or not . This system will also be helpful for identifying the areas where the students need to work on for placement. To achieve this, the predictor uses students details like academic marks ,coding skill etc.

Contents

1	INTRODUCTION	1
2	LITERATURE REVIEW	2
2.1	A Survey on Placement prediction system using Machine learning . . .	2
2.1.1	Introduction	2
2.1.2	Working	3
2.1.3	Advantages	5
2.1.4	Disadvantages	5
2.2	Use of ID3 Decision Tree Algorithm for Placement Prediction	6
2.2.1	Introduction	6
2.2.2	Working	7
2.2.3	Advantages	10
2.2.4	Disadvantages	10
2.3	Student Prediction system for Placement training using Fuzzy inference System	11
2.3.1	Introduction	11
2.3.2	Working	12
2.3.3	Advantages	14
2.3.4	Disadvantages	14
2.4	Student Placement Prediction Using Support Vector machine Algorithm	15
2.4.1	Introduction	15
2.4.2	Working	16
2.4.3	Advantages	19
2.4.4	Disadvantages	19
2.5	Recruitment System with Placement Prediction	20
2.5.1	Introduction	20
2.5.2	Working	21
2.5.3	Advantages	24
2.5.4	Disadvantages	24
3	PROBLEM STATEMENT	25

4	PROPOSED SYSTEM	26
4.1	Introduction	26
4.2	Technology used	26
4.2.1	Anvil	26
4.2.2	Python Packages	28
4.2.3	GitHub	30
4.2.4	XGBoost	30
5	FEASIBILITY STUDY	34
5.1	Operational Feasibility	35
5.2	Technical Feasibility	35
5.3	Economical Feasibility	36
5.4	Behavioral Feasibility	36
5.5	Software Feasibility	36
5.6	Hardware Feasibility	36
6	REQUIREMENT SPECIFICATION	37
6.1	Software Requirements	37
6.1.1	User Requirements	37
6.2	Hardware Requirements	37
6.2.1	Development Requirements	37
6.2.2	User Requirements	37
7	SYSTEM DESIGN	38
7.1	Data Flow Diagram	38
7.2	UML DIAGRAMS	40
7.2.1	Sequence Diagram	41
7.2.2	Use Case Diagram	41
7.2.3	Activiy Diagram	43
7.2.4	Class Diagram	45
8	IMPLEMENTATION	46
8.1	Module Description	46
8.2	Working	48

9 TESTING	49
9.1 Testing Methodology	49
10 RESULT	52
11 PERFORMANCE AND EVALUATION	53
12 SCREENSHOT	55
13 CONCLUSION	58
REFERENCES	

List of Figures

1	System Architecture	3
2	Ensemble Learning	4
3	Student Data	7
4	Decision Tree	8
5	Fuzzy Inference System of each Input	12
6	Output of Classified Individual student result	13
7	System Architecture	16
8	Dynamic predictor	21
9	Static predictor	22
10	Bagging Classifier	31
11	Boosting	32
12	0-level DFD	38
13	1-level DFD	39
14	2-level DFD	40
15	Sequence Diagram	41
16	Use Case Diagram	42
17	Activity Diagram	44
18	Class Diagram	45
19	System Architecture	46
20	Student Predicted to be Placed	52
21	Student Predicted to be not Placed	52
22	Confusion Matrix for XGBoost classifier	53
23	AUC for XGBoost classifier	54
24	AP for XGBoost classifier	54
25	Login Page	55
26	Sign Up Page	55
27	Home Page	56
28	Other options	56
29	Placement Details	57
30	Placement Prediction	57

CHAPTER 1**1 INTRODUCTION**

According to statistics 1.5 million engineers graduate in India every year. The demand for skilled and qualified students is rising day by day. Thus the company use a good amount of capital in recruiting students from in-campus and off-campus because number of skilled and qualified students are very low. Colleges and Institution needs to focus on practical knowledge of real world rather than completing their syllabus. Placements are the biggest opportunities in the life of a student and they need to be fully prepared while attempting it. Placement Predictor system helps in predicting whether a student will get placement or not. This system can also be helpful for identifying the areas where student needs to work on for placement. This system uses student's details like academic marks, coding skills, etc. This system uses previous year placement statistics and student dataset for the placement prediction so the placement cell of the organization could set up a placement anticipated rundown for the present students. Along these lines it is important to direct an investigation on different placement prediction frameworks. This report studies on various placement prediction framework models.

CHAPTER 2**2 LITERATURE REVIEW****2.1 A Survey on Placement prediction system using Machine learning****2.1.1 Introduction**

The purpose of placement management system is to modify the present manual system by the assistance of computerized software system fulfilling their needs, so their valuable data/information is stored for a longer time with simple accessing and manipulation of data. Student's academic achievements and their placement in campus selection is a difficult issue in current manual system. Monitoring the student's progress for his or her campus placement helps in monitoring the student's progression within the academic surroundings. the aim of organizations is to supply superior opportunities to their students.

This proposed student prediction system is most important approach which can be used to differentiate the student data/information on the basis of the student performance. Managing placement and coaching records in any larger organization is quite tough because of the large number of students. This system can classify the student knowledge with ease and can be useful to several educational organizations. There are several classification algorithms and mathematics-based techniques which can be taken nearly as good assets for classifying the students' information set in the education field.

In Our system, Naïve Bayes, SVM, KNN algorithm is applied to predict student performance which can facilitate to identify performance of students and also provides suggestion to improve performance for students such as I are going to classify the student's knowledge set for placement and non-placement classes based on that result, education organizations can give superior training to their students. Based on data received by system, student's performance is analysed in numerous views to check the achievements of the students through their activities and suggests improvement for better placement.

2.1.2 Working

A system is deployed in which the students will register/Login into the system and enter their biodata and skillsets, according to students' academic details the system will identify whether the student is eligible for the placement and recommend the courses to the students. Admin creates the courses and registers students to the respective courses.

Admin can view the courses and students along with their attributes. Admin predicts placement status of current students. If the student is eligible for placement mail will be sent to the student from admin and students names will be displayed on the dashboard in their colleges.

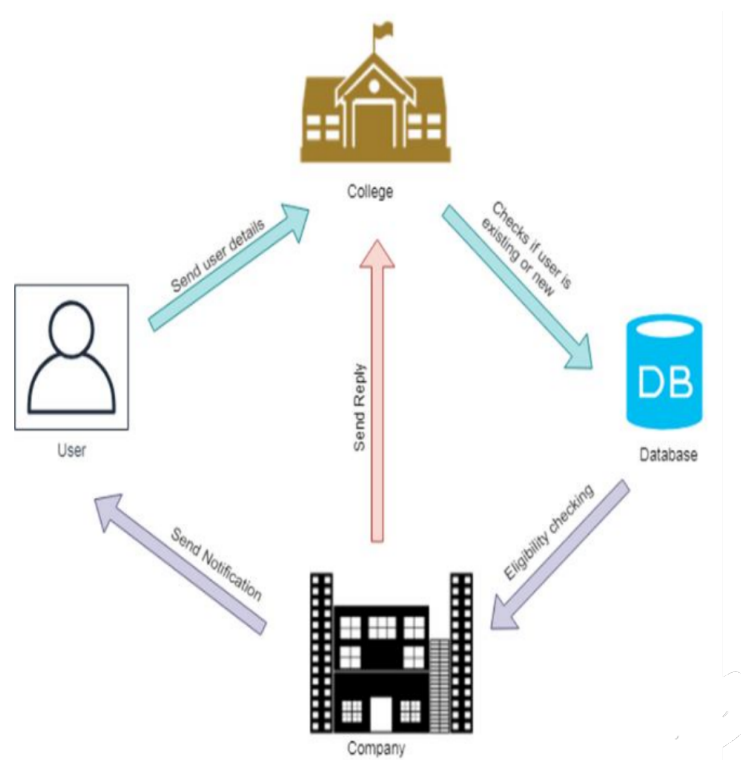


Figure 1: System Architecture

Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model. To better understand this definition let's take a step back into goal of machine learning and model building.

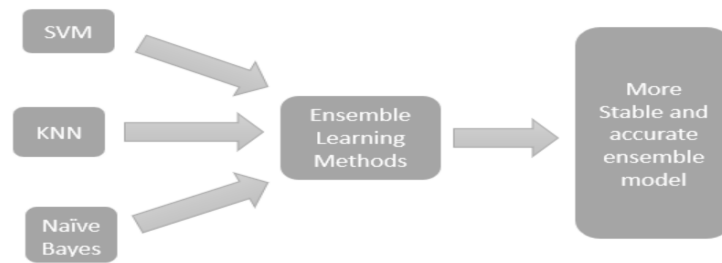


Figure 2: Ensemble Learning

The goal of any machine learning problem is to find a single model that will best predict our wanted outcome. Rather than making one model and hoping this model is the best/most accurate predictor I can make, ensemble methods take a myriad of models into account, and average those models to produce one final model.

This system is helpful for institutions to predict student's campus placement. This system would help reduce tedious job of manual placement system. The placement officer can work on identifying the weaknesses of each students and can suggest improvements so that the students can overcome the weakness and perform to the best of their abilities.

2.1.3 Advantages

- Ensemble methods have higher predictive accuracy, compared to the individual models.
- Ensemble of models is always less noisy and is more stable.
- It reduces the tedious job of manual placement system.

2.1.4 Disadvantages

- The art of ensembling is hard to learn and any wrong selection can lead to lower predictive accuracy than an individual model.
- Ensembling is less interpretable, the output of the ensemble model is hard to predict and explain.
- Ensembling is expensive in terms of both time and space.
- It takes more time to compute.

2.2 Use of ID3 Decision Tree Algorithm for Placement Prediction

2.2.1 Introduction

Every year corporate companies come to colleges in order to recruit students. Recruitment is one of the most essential processes for any organization as they look for skilled and qualified professionals to fill up the positions in their organization. Many companies hire students through campus recruitment process.

Campus recruitment is an efficient way to get the right resources at the right time with minimal cost and within minimum time frame. While the industry hires candidates from different institutes, students too get a chance to start their career with some of the best companies in the corporate sector. The main aim of this paper is to identify relevant attributes based on quantitative and qualitative aspects of a student's profile such as CGPA, academic performance, technical and communication skills and design a model which can predict the placement of a student.

For this purpose ID3 classification technique based on decision tree has been used. The result of this analysis will assist the academic planners to design a strategy to improve the performance of students that will help them in getting placed at the earliest.

2.2.2 Working

Campus placement is a process where companies come to colleges and identify students who are talented and qualified, before they finish their graduation. The proposed system determines the likelihood of placement based on various attributes of a student's profile. Depending on the parameters, manual classification is done whether the student is placed or not placed. A decision tree is then implemented to determine the probable outcome of a student being placed.

	A	B	C	D	E	F	G	H	I	J
1	Dept	CGPA	'Programming Skills'	'Future Studies'	SSC	HSC	Communication skills	Backlog	Internships	Placement
2										
3	CS	8.33	Good	Yes	92	88	Good	0	1	'Not Placed'
4	EXTC	6.78	Good	No	58	56	Average	1	0	'Not Placed'
5	EXTC	6	Good	No	73	64	Good	2	0	'Not Placed'
6	IT	9.3	Poor	Yes	90	86	Average	0	0	'Not Placed'
7	IT	9.67	Good	No	85	82	Good	0	0	Placed
8	CS	6.87	Poor	No	65	67	Average	0	0	'Not Placed'
9	CS	9.22	Poor	Yes	93	86	Average	0	0	'Not Placed'
10	EXTC	7.9	Good	No	82	78	Good	0	1	Placed
11	IT	7.99	Good	No	80	77	Good	0	0	Placed
12	CS	9.9	Average	No	98	92	Good	0	1	Placed
13	CS	8.6	Average	No	93	87	Good	0	1	Placed
14	CS	8.7	Poor	No	76	63	Good	0	0	'Not Placed'
15	CS	6.12	Good	No	89	84	Average	0	0	'Not Placed'
16	IT	7	Good	Yes	94	89	Good	0	1	'Not Placed'

Figure 3: Student Data

The combination of various attributes determines whether the student is placed or not. The quantitative aspects like undergraduate CGPA, Marks obtained in X and XII form the major aspect of a student's academic endeavours. The qualitative aspects like communication and programming skills form a backbone for a student to get placed as each recruiting company desires to hire students that have a sound technical knowledge and ability to communicate effectively. The other factors like internships, backlogs, future studies add value only when the prior requirements are met.

Based on the training set, information gain and entropy is calculated to determine the splitting attribute for constructing the decision tree. The algorithm gives a pruned decision tree with leaves as the decision that is placed or not placed. The primary node consists of programming skills which can accommodate three possible values viz.

Good, Average and Poor. If the programming skills are Poor, the student is not placed. Furthermore, if the programming skills are good, the student may be placed based on the academic credential which is CGPA. If the CGPA of the student is above 7, student will be placed otherwise the student will not be placed. Also, if the student has average programming skills, he may still be placed based on other attributes like internships, future studies, communication skills, etc. The final decision tree with four leaf nodes is obtained as shown below. The leaf nodes hold the value whether the student is placed or not placed.

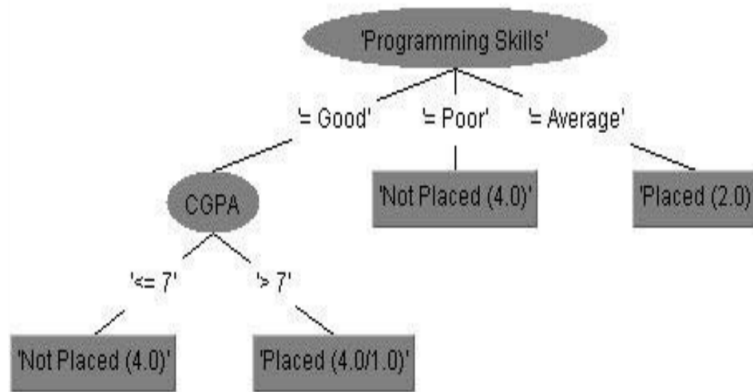


Figure 4: Decision Tree

The root node chosen here is Programming Skills. Further classification is done by calculating information gain and entropy for each attribute. Consider the attribute future studies; it has two possible classes viz. Yes and No. There are four students who wish to pursue future studies and remaining ten out of fourteen who do not have any plans to opt for higher studies. According to the training set, all the students who wish to pursue future studies are not placed. This indicates that all the information is contained in a single class. Hence, entropy becomes zero. Higher value of entropy indicates higher degree of distribution of information among classes. The lowest value of information gain is obtained for programming skills. Thus it is chosen as the root node. Further, the next lowest value (CGPA) is taken as the split node for next level. The subsequent nodes of decision tree at each level are determined by the value obtained in information gain.

In this paper ID3 classification algorithm is used to generate decision rule. The

generated decision rule can be used to predict a student's campus placement. The result of this algorithm can be used by the placement-in-charge to identify those set of students that are likely to face problems in campus placement. The classification model can play an important role in increasing the placement statistics. It can be concluded that classification algorithms can be used successfully in order to predict student placement. Further the implementation can be done in development and application of novel computational techniques for the analysis of large datasets.

2.2.3 Advantages

- The best feature of using trees for analytics is that they are easy to interpret and explain. Decision trees are very intuitive and easy to explain.
- Nonlinear relationships between parameters do not affect tree performance.
- Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.
- Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.

2.2.4 Disadvantages

- A small change in the data can cause a large change in the structure of the decision tree causing instability.
- For a Decision tree sometimes calculation can go far more complex compared to other algorithms.
- Decision tree often involves higher time to train the model.
- Decision tree training is relatively expensive as the complexity and time has taken are more.

2.3 Student Prediction system for Placement training using Fuzzy inference System

2.3.1 Introduction

Proposed student prediction system is most vital approach which may be used to differentiate the student data/information on the basis of the student performance. Managing placement and training records in any larger organization is quite difficult as the student number are high; in such condition differentiation and classification on different categories becomes tedious.

Proposed fuzzy inference system will classify the student data with ease and will be helpful to many educational organizations. There are lots of classification algorithms and statistical base technique which may be taken as good assets for classify the student data set in the education field.

In this paper, Fuzzy Inference system has been applied to predict student performance which will help to identify performance of the students and also provides an opportunity to improve to performance. For instance, here I will classify the student's data set for placement and non-placement classes.

2.3.2 Working

The main objective of this project is to classify large set of student data set using fuzzy logic and predicting student for placement training whether the student is eligible or not for placement training. As I discussed above I am taking two classes for final year student, classes are placement training and non-placement training. For the placement training those students will be eligible those will get good marks or good CGPA in exam and remaining student will go for extra classes to improve their performance. From this approach I can predict all students in few times, if I want to classify all students on excel sheet so it will take lot of time, so these approaches are much better compare to other and it will help you.

This proposal may also be able to judge the performance of the student continuously. The main advantage of this system is to improve accuracy and speed of the student by conducting an exam and decide some criteria to pass that exam if student fails in exam than some important classes or training will provide to the student it will help to student to improve itself. The advantage of the system is the accuracy of the prediction and speed of the result provided. So the fuzzy system provides the easy way to classify several numbers of student data set. By making a fuzzy inference system and create some rules which will predict the student for pre-defined classes such as placement training and non-placement training.

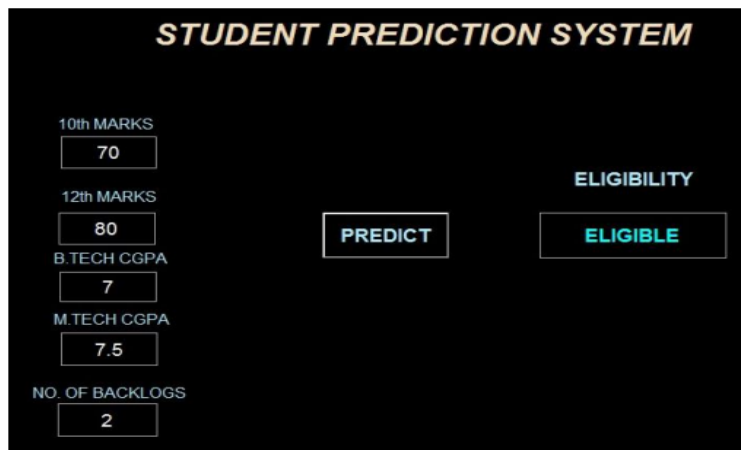


Figure 5: Fuzzy Inference System of each Input

For this system I collected 31 sample data set of M.Tech computer science student from our organization. In this data set put student registration number, name, 10th

marks, 12th marks, B.Tech cgpa, M.Tech cgpa and number of backlogs. For classifying data set I skip some part of data set like registration number as well as name also here I take only academic marks and current backlog only and created more than 500 rules for predicting student performance for placement training on the basis of certain parameters.

For example, if (10th marks is poor) and (12th marks is poor) and (B.Tech cgpa is poor) and (M.Tech cgpa is poor) and (backlog is less than 2) then (output is NO). From this fuzzy system can classify thousands number of student's dataset. The main step of fuzzy based student prediction system is creating membership function for each input (10th marks, 12th marks, B.Tech CGPA, M.Tech CGPA, and number of backlogs) and creates output in terms of eligibility corresponding inputs.



STUDENT PREDICTION SYSTEM

10th MARKS
70

12th MARKS
80

B.TECH CGPA
7

M.TECH CGPA
7.5

NO. OF BACKLOGS
2

PREDICT

ELIGIBILITY
ELIGIBLE

Figure 6: Output of Classified Individual student result

This system easily predicts and analyzes lot of student data set for predefined classes by using fuzzy logic which will definitely be a good asset to organization to analyze the information of huge number of student information and data sets. Future studies are required to investigate new hybrid models of fuzzy classification algorithms to improve the performance of prediction system.

2.3.3 Advantages

- It is a robust system where no precise inputs are required.
- The Fuzzy Logic algorithms can be coded using less data, so they do not occupy a huge memory space.
- As it resembles human reasoning, these systems are able to solve complex problems where ambiguous inputs are available and take decisions accordingly.
- These systems are flexible and the rules can be modified.

2.3.4 Disadvantages

- There is no single systematic approach to solve a problem using Fuzzy Logic. As a result, many solutions arise for a particular problem, leading to confusion.
- A major drawback of Fuzzy Logic control systems is that they are completely dependent on human knowledge and expertise.
- You have to regularly update the rules of a Fuzzy Logic control system.

2.4 Student Placement Prediction Using Support Vector machine Algorithm

2.4.1 Introduction

Campus placement plays a vital role in every educational institution in helping students to achieve their goals. All students dream to obtain a job offer in their hands before they leave their college. In this paper, a predictive model is designed which can predict whether a student get placed or not.

The main objective of this project is to analyze the student's academics data, aptitude data and predict the placement possibilities of students to have an idea about where they stand and what to be done to obtain a good placement. Which also aids to increase the placement percentage of the institutions?.

The data has been collected by the institution for which prediction is going to be done and by applying suitable data pre-processing techniques. The model is built by both training and test set which gives accuracy in prediction. Here I use a single supervised machine learning algorithm named support vector machine algorithm. This algorithm independently predicts the results and I then compare the efficiency of the algorithm, which is based on the dataset. This model will help the placement cell to focus on the potential students and help them to improve their technical and other skills.

2.4.2 Working

In this paper I use machine learning techniques to predict the placement status of students based on a dataset. The parameters in the dataset which are considered for the prediction are Quantitative scores, Logical Reasoning scores, Verbal scores, Programming scores, CGPA, internal marks, external marks, list of students placed in a company. The placement prediction is done by machine learning Algorithm using SVM.



Figure 7: System Architecture

1. **Data Collection** Sample data has been collected from college placement department. As an input for model prediction, which consist of all the required dataset.
2. **Data Preparation & Pre-processing** Data preparation is a step in a data analysis process in which data from one or more sources is cleaned, transformed and enriched to improve the quality of data prior to its use. The collected data were then pre-processed to fill the missing data and made compatible for further processing.

3. **Data Splitting** 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. I 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 I will test the models on our test set to examine how accurately it will predict. A general rule of the thumb is to assign 80to test set.
4. **Algorithm Building** SVM algorithm is appied on the dataset. SVM stands for Support Vector Machine. It is also a supervised machine learning algorithm that can be used for both classification and regression problems. However, it is mostly used for classification problems. A point in the n-dimensional space is a data item where the value of each feature is the value of a particular coordinate. Here, n is the number of features you have. After plotting the data item, I perform classification by finding the hyper-plane that differentiates the two classes very well. Now the problem lies in finding which hyper-plane to be chosen such that it is the right one. The Support Vector Machine (SVM) helps in identifying the hyperplane for classifying the data samples. In the case of multiple hyperplanes, the one which has maximum distance from the data points was chosen for better classification.
5. **Evaluation and Testing:** The performance measurement of the model was evaluated with the help of various metrics like accuracy, sensitivity, F1-score and precision. The performance visualization of the multi-class classification problem was analyzed using a graphical plot AUC (Area under the Curve) ROC (Receiver Operating Characteristics) curve that reveals the analytical ability of a binary classifier system as its discrimination threshold. The ROC curve is generated by plotting the true positive rate against false-positive rates at various threshold rates. The best algorithm based on the performance parameters was selected to predict the placement category of students. Based on the details provided by the students, the placement category could be predicted and the result would be displayed along with the suggestions for further improvement.

From the study it is clear that the student dataset containing academic and placement details are a potential source for predicting the future placement chances and It is clear that SVM gives an accuracy of 100. This prediction can enlighten students to

identify their capabilities and improve accordingly. This system also helps in the academic planning of an institution to prepare proper strategies and improve the placement statistics for the future years.

2.4.3 Advantages

- SVM works relatively well when there is a clear margin of separation between classes.
- SVM is more effective in high dimensional spaces & SVM is relatively memory efficient.
- SVM is effective in cases where the number of dimensions is greater than the number of samples.

2.4.4 Disadvantages

- SVM algorithm is not suitable for large data sets.
- SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.
- In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.

2.5 Recruitment System with Placement Prediction

2.5.1 Introduction

The availability of information and the facility for the user to take action on the information collected have been revolutionized by the use of the Internet and the World Wide Web. The placement process can be managed using the internet which arises a need to develop a web-based placement management system specifically by the recruiters and the software engineers that can be used as a Recruitment system (Online TnP portal).

This system can be used as an application for both candidates and recruiters. Advanced features for recruiters are available as they can shortlist candidates for further rounds according to their requirements on the basis of the probability obtained. The current recruitment system recruiters do not possess candidate information apart from his/her CV.

This proposed system aims to analyze the candidate performance and recommend candidates fittest for the job using Random Forest Regressor algorithm that will help to maximize the placement probability of candidates easing the recruiter's task. Random Forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. This system will provide ease and efficiency in recruitment process.

2.5.2 Working

The proposed system is a Web application meant to be used for recruitment process. It can be used by organizations as a tool for effective recruitment by analyzing the candidate's fitness for the job. And colleges can use the system to get an idea about the probability of the student to be placed prior to the placement drive with the help of Placement prediction feature. This system recommends the candidates on the basis of their likelihood to get placed. It considers parameters like candidate's SSC marks, HSC marks, CGPA, gender to predict the placement probability. Machine learning technique is used to implement Random Forest Regressor algorithm. The model is first trained on a dataset of any previous placement drives and then used to predict the probability of the candidate to get placed. The system consists of two phases –

1. Real-time placement prediction system (dynamic).
2. Probable candidate list generator(static).

Real-time placement prediction system (dynamic):

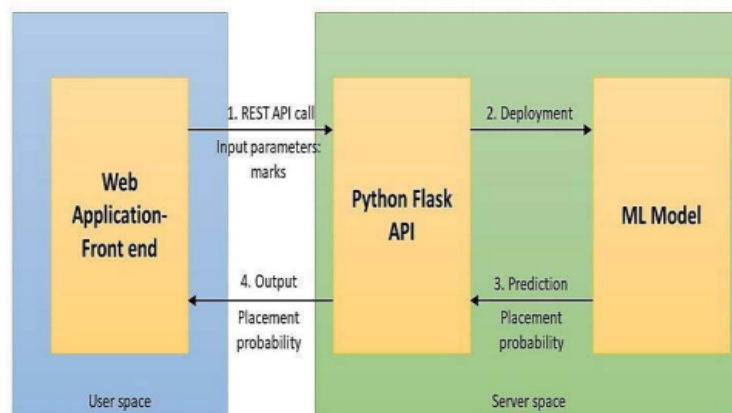


Figure 8: Dynamic predictor

This phase is for walk-in candidates where no previous record of him is available in the system. The following steps are carried out:

Step1: Placement candidate enters required credentials.

Step2: These input parameters will be passed to the Flask API using REST API call.

Step3: The data forwarded by API will be given to trained ML model for calculation of placement probability.

Step4: This placement probability will be displayed on the web page back through Flask API and the candidate record will be saved in a csv file.

Probable candidate list generator(static):

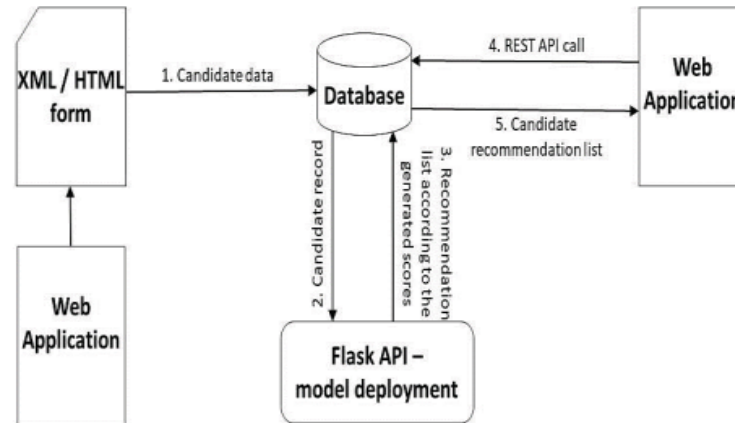


Figure 9: Static predictor

This phase is for candidates who enter their data through a form or whose records are already present in the database of the system.

Step1: HTML/XML form is filled by candidate to store data in database using browser.

Step2: Flask API is used to implement ML algorithms on database data.

Step3: A trained ML model is implemented on the candidate records and scores are generated.

Step4: A candidate recommendation list is generated based on the scores.

Step5: Whenever a placement client requests the list, it will be retrieved from database and displayed on the web page.

Prediction Process in both phases: A pickle file is created, in which scikit-learn is imported which is a Machine Learning module. Using scikit-learn, Random Forest Regressor algorithm is trained on a dataset which contains historic data of placement drive. Now this pickle file is read by the Flask API file and the trained model is fitted on the current candidate records.

The proposed system consisting of dynamic prediction uses Machine learning to

predict the placement probability of candidates dynamically using the parameters such as CGPA, HSC marks, SSC marks. It overcomes the limitations of current recruitment system which displays discrete values and gives an idea about placement to the candidates. The Scikit learn module provides us with a Random Forest Regressor algorithm which helps in generating probabilities with accuracy for large datasets and hence is comfortably suited for this purpose.

2.5.3 Advantages

- It reduces overfitting in decision trees and helps to improve the accuracy.
- It is flexible to both classification and regression problems.
- Normalising of data is not required as it uses a rule-based approach.
- It works well with both categorical and continuous values & it automates missing values present in the data.

2.5.4 Disadvantages

- It requires much computational power as well as resources as it builds numerous trees to combine their outputs.
- It also requires much time for training as it combines a lot of decision trees to determine the class.
- Due to the ensemble of decision trees, it also suffers interpretability and fails to determine the significance of each variable.

CHAPTER 3

3 PROBLEM STATEMENT

Previous Literatures Survey's have used machine learning algorithms like :

1. Ensemble learning (with SVM, KNN and Naive Bayes) for recruiters and colleges to predict the placement.
2. ID3 algorithm for academic planner to predict the placement.
3. Fuzzy logic for analyzing student performance to predict the placement.
4. SVM for placement cell to predict the placement.
5. Random forest regressor for recruiters to predict the placement.

Eventhough the algorithms has classified the placement probability as can be Placed or not, the expected accuracy was not attained with any of the above mentioned algorithm as the used data set was small, algorithm was complex, algorithm was not efficient etc... . So in this paper I have decided to use huge dataset and select the best algorithms out of the following:

1. Logistic Regression
2. KNN Algorithm
3. SVM Algorithm
4. XGBoost Classifier
5. Decision Tree Algorithm

CHAPTER 4

4 PROPOSED SYSTEM

4.1 Introduction

The importance of campus placement assistant system is to modify the present manual system where the data is in an unorganized manner. Due to unorganized data the stakeholders namely Placement Officers and Recruiters face difficulty in selecting the best student from the campus. The students should also need to have a clear picture of the placements going in the campus, so an effective computerized system will help the stakeholders to select the best student and the student to understand their level that is whether they can be placed or not.

The proposed system helps a student to get the placement details and can check for placement prediction, mock test and interview question this helps the student to have an environment where the student can get placement details. This also helps the admin or placement officer to get the details of student after prediction and the placement details from the company or recruiters can be directly passed to the students.

4.2 Technology used

4.2.1 Anvil

Anvil is a new way to build web apps, with nothing but Python. This documentation will tell you all about how it works. You do not need to know anything about HTML, Javascript or website development to use Anvil. All you need is some familiarity with the Python programming language. Anvil fills in these gaps by allowing you to build a full-stack web app using only Python. You can build a user interface with a simple drag and drop UI (or build it with code if you insist), plot with your favorite Python plotting library (Plotly, Matplotlib, etc.), and then deploy to the web in one click. No servers or containers to deal with.

Structure of an Anvil app

An Anvil app is made up of:

1. A User Interface, which you design with a drag-and-drop designer.
2. Client-side Python code, which runs in the web browser.

3. Server-side Python code, which runs on Anvil's servers.
4. A built-in database (Data Tables), which stores your data.
5. Some Python code running on your computer, which can also interact with your app.

What else you can do. Anvil also has built-in support for:

1. One-click hosting and deployment
2. Versioning your app with Git version control
3. Sending and receiving emails
4. Exposing and using HTTP APIs
5. Integrating with services from: Google, Microsoft, Facebook and anything else with a Python SDK!

Deep Notebook

Deepnote is a seriously slick Python notebook, hosted in the cloud, with incredible real-time collaboration. It's great for working with other data scientists. But what happens when you want to share your project with non-programmers? They need an easy-to-use interface – so you need to deploy what you've built. Deepnote has been used to integrate the machine learning code to Anvil.

4.2.2 Python Packages

NumPy NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

At the core of the NumPy package, is the ndarray object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:

- NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an ndarray will create a new array and delete the original.
- The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.
- NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.

pandas

pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way towards this goal. Pandas makes it simple to do many of the time consuming, repetitive tasks associated with working with data, including:

- Data cleansing

- Data fill
- Data normalization
- Merges and joins
- Data visualization
- Statistical analysis
- Data inspection
- Loading and saving data etc..

Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc. **Seaborn**

Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data. One has to be familiar with Numpy and Matplotlib and Pandas to learn about Seaborn.

Seaborn offers the following functionalities:

- Dataset oriented API to determine the relationship between variables.
- Automatic estimation and plotting of linear regression plots.
- It supports high-level abstractions for multi-plot grids.
- Visualizing univariate and bivariate distribution.

4.2.3 GitHub

GitHub is a Git repository hosting service that provides a web-based graphical interface (GUI). It helps every team member work together on a project from anywhere, making it easy to collaborate.

GitHub is one place where project managers and developers coordinate, track, and update their work, so projects stay transparent and on schedule. The packages can be published privately, within the team, or publicly for the open-source community. Downloading packages from GitHub enables them to be used and reused. GitHub helps all team members stay on the same page and stay organized. Moderation tools, like issue and pull request locking, helps the team focus on the code.

GitHub Desktop GitHub Desktop is an application that enables GUI-based interaction with GitHub. Git provides a wide range of commands for Git activities like creating repositories, commits, pull requests, and so on. However, GitHub desktop provides GUI-based those activities using best practices with Git and GitHub. I can use GitHub desktop to do most of the Git commands with UI and clicks which makes collaboration and working with Git more flexible. You can connect to your account, create git repositories, add projects, do the changes, and commit easily with the interface. I will be doing hands-on with the GitHub desk in this article.

4.2.4 XGBoost

XgBoost stands for Extreme Gradient Boosting, which was proposed by the researchers at the University of Washington. It is a library written in C++ which optimizes the training for Gradient Boosting.

Before understanding the XGBoost, I first need to understand the trees especially the decision tree: **Decision Tree:** A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

A tree can be “learned” by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions.

Bagging: A Bagging classifier is an ensemble meta-estimator that fits base classi-

fiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it. Each base classifier is trained in parallel with a training set which is generated by randomly drawing, with replacement, N examples (or data) from the original training dataset, where N is the size of the original training set. The training set for each of the base classifiers is independent of each other. Many of the original data may be repeated in the resulting training set while others may be left out.

Bagging reduces overfitting (variance) by averaging or voting, however, this leads to an increase in bias, which is compensated by the reduction in variance though.

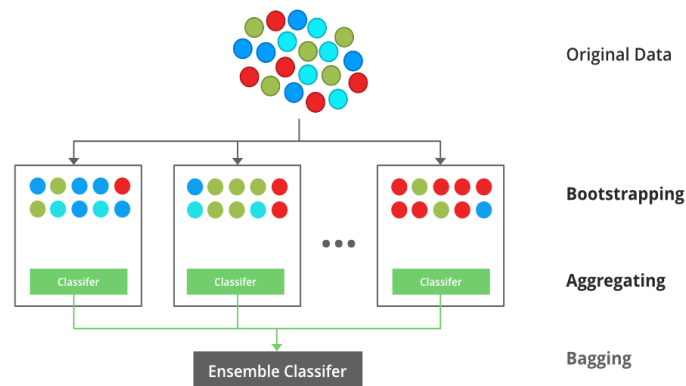


Figure 10: Bagging Classifier

Random Forest: Every decision tree has high variance, but when I combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data and hence the output doesn't depend on one decision tree but multiple decision trees. In the case of a classification problem, the final output is taken by using the majority voting classifier. In the case of a regression problem, the final output is the mean of all the outputs. This part is Aggregation.

The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. Random Forest has

multiple decision trees as base learning models. I randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

Boosting: Boosting is an ensemble modelling, technique that attempts to build a strong classifier from the number of weak classifiers. It is done by building a model by using weak models in series. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models are added.



Figure 11: Boosting

Gradient Boosting Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error. In contrast to Adaboost, the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of predecessor as labels.

There is a technique called the Gradient Boosted Trees whose base learner is CART (Classification and Regression Trees).

XGBoost XGBoost is an implementation of Gradient Boosted decision trees. XGBoost models majorly dominate in many Kaggle Competitions. In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual

classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.

Features of XGBoost

1. **Regularization:** Since the ensembling of decisions, trees can sometimes lead to very complex. XGBoost uses both Lasso and Ridge Regression regularization to penalize the highly complex model.
2. **Parallelization** and Cache block: In, XGboost, I cannot train multiple trees parallel, but it can generate the different nodes of tree parallel. For that, data needs to be sorted in order. In order to reduce the cost of sorting, it stores the data in blocks. It stored the data in the compressed column format, with each column sorted by the corresponding feature value. This switch improves algorithmic performance by offsetting any parallelization overheads in computation.
3. **Tree Pruning:** XGBoost uses max_depth parameter as specified the stopping criteria for the splitting of the branch, and starts pruning trees backward. This depth-first approach improves computational performance significantly.
4. **Cache-Awareness and Out-of-score computation:** This algorithm has been designed to make use of hardware resources efficiently. This is accomplished by cache awareness by allocating internal buffers in each thread to store gradient statistics. Further enhancements such as ‘out-of-core computing optimize available disk space while handling big data-frames that do not fit into memory. In out-of-core computation, Xgboost tries to minimize the dataset by compressing it.
5. **Sparsity Awareness:** XGBoost can handle sparse data that may be generated from preprocessing steps or missing values. It uses a special split finding algorithm that is incorporated into it that can handle different types of sparsity patterns.
6. **Weighted Quantile Sketch:** XGBoost has in-built the distributed weighted quantile sketch algorithm that makes it easier to effectively find the optimal split points among weighted datasets.
7. **Cross-validation:** XGboost implementation comes with a built-in cross-validation method. This helps the algorithm prevents overfitting when the dataset is not that big.

CHAPTER 5**5 FEASIBILITY STUDY**

All projects are feasible when given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. An estimate is made of whether the identified user needs may be satisfied using current software and hardware technologies. The study will decide if the proposed system will be cost effective from the business point of view and if it can be developed in the given existing budgetary constraints. The feasibility study should be relatively cheap and quick. The result should inform the decision of whether to go ahead with a more detailed analysis. Feasibility analysis is the procedure for identifying the candidate system, evaluating and electing the most feasible system. This is done by investigating the existing system in the area under investigation or generally ideas about a new system. It is a test of a system proposal according to its work ability, impact on the organization, ability to meet user needs, and effective use of resources. The objective of feasibility study is not to solve the problem but to acquire a sense of its scope. Feasibility analysis involves following steps

- Form a project and appoint a project leader
- Prepare system flowcharts
- Create a web site
- Weigh system performance and cost data
- Prepare and report final project directive to management The study is done in these phases
- Operational feasibility
- Technical feasibility
- Economical feasibility
- Behavioral feasibility
- Software feasibility

- Hardware feasibility

5.1 Operational Feasibility

Proposed projects are beneficial only if they can be turned into information system that will meet the organization's operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to implementation? Here are questions that will help test the operational feasibility of a project: Is there sufficient support for the projects from management? Are current business methods acceptable to the users? Have the users been involved in the planning and development of the project? Will the proposed system cause harm? The purpose of the operational feasibility study is to determine whether the new system will be used if it is developed and implemented. And whether there will be resistance from users that will undermine the possible application benefits

5.2 Technical Feasibility

A study of function performance and constraints may improve the ability to create an acceptable system. Technical feasibility is frequently the most difficult area to achieve at the stage of product engineering process. Considering that are normally associated with the technical feasibility include

- Development risk
- Resource availability
- Technology Technical feasibility study deals with the hardware as well as software requirements. The scope was whether the work for the project is done with the current equipments and the existing software technology has to be examined in the feasibility study. The outcome was found to be positive. In proposed system, data can be easily stored and managed using database management system software. The reports and results for queries can be generated easily. Thus, system is technically feasible.

5.3 Economical Feasibility

A cost evaluation is weighed against the ultimate income or benefit derived from the developed system or product. When compared to the advantage obtained from implementing the system its cost is affordable. Also the system is designed to meet the modifications required in the future. So, most of the required modifications can be done without much re-work. Proposed system was developed with the available resources. Since cost input for the software is almost nil the output of the software is always a profit. Hence software is economically feasible. In the existing system, manpower is required. In the proposed system, number of employee

5.4 Behavioral Feasibility

People are inherently resistant to changes and computer is known for facilitating the changes. An estimate should be made of how strongly the user staff reacts towards the developments of the computerized system. In the existing system more manpower is required and time factor is more. In proposed system, both man power and time factors are reduced and also unnecessary burden is reduced. Thus the remaining people are made to engage in some other important work. Therefore, the system is behaviorally feasible.

5.5 Software Feasibility

Even though software is developed in a very high software environment, it will be supported by many other platform and environments with minimum changes.

5.6 Hardware Feasibility

The software can be developed with resource already existing. Here the consideration is that the existing hardware resources support the technologies that are to be used by the new system. No hardware was newly bought for the project and hence. Software is to achieve hardware feasibility.

CHAPTER 6**6 REQUIREMENT SPECIFICATION****6.1 Software Requirements**

- Python (Version 3.0+)
- Anvil
- GitHub
- DeepNote
- ipynb file

6.1.1 User Requirements

- Any Web Browser

6.2 Hardware Requirements**6.2.1 Development Requirements**

- Intel 7th gen i3 / AMD Ryzen 3
- 4GB RAM
- Hard Disk 8GB or more

6.2.2 User Requirements

- Intel Pentium / AMD A6
- 512 MB RAM
- 112 GB Storage space

CHAPTER 7

7 SYSTEM DESIGN

7.1 Data Flow Diagram

DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modeling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

In Software engineering DFD(data flow diagram) can be drawn to represent the system of different levels of abstraction. Higher-level DFDs are partitioned into low levels-hacking more information and functional elements. Levels in DFD are numbered 0, 1, 2 or beyond. Here, I will see mainly 3 levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.

- **0-level DFD:** It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

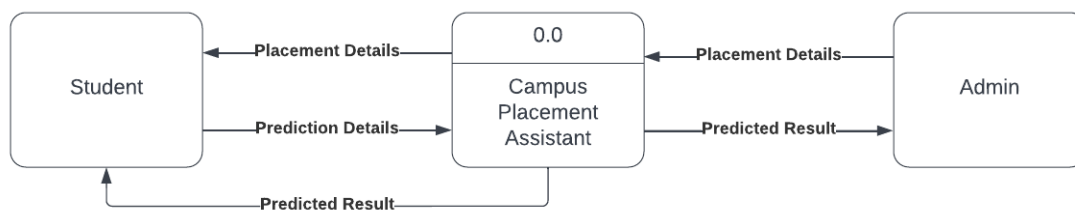


Figure 12: 0-level DFD

- **1-level DFD:** In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, I highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses.

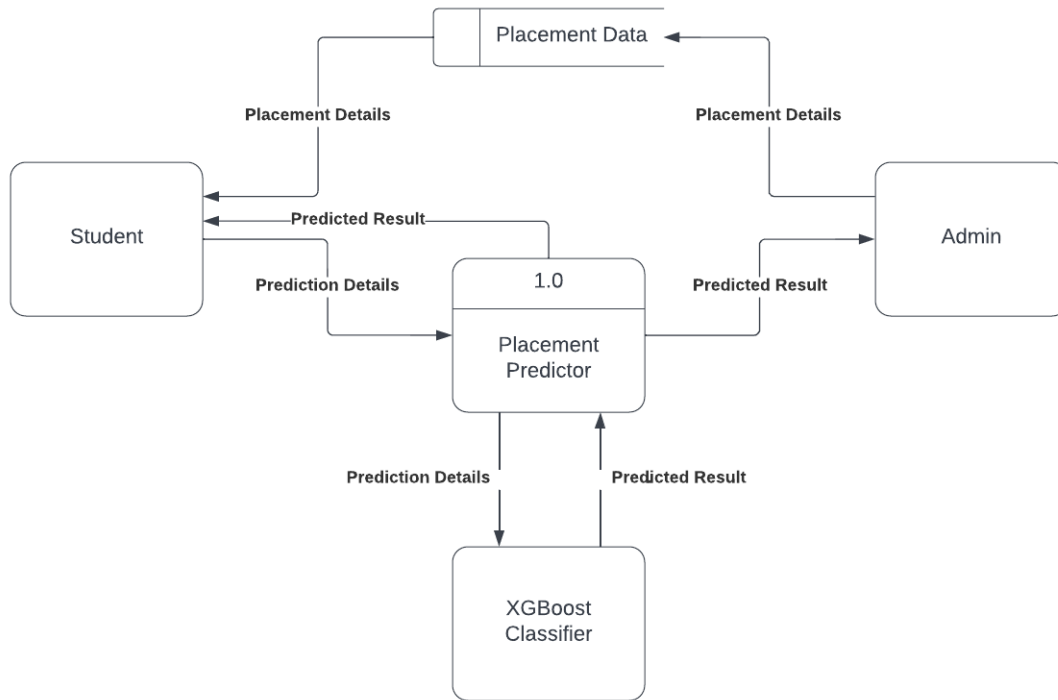


Figure 13: 1-level DFD

- **2-level DFD:** 2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning.

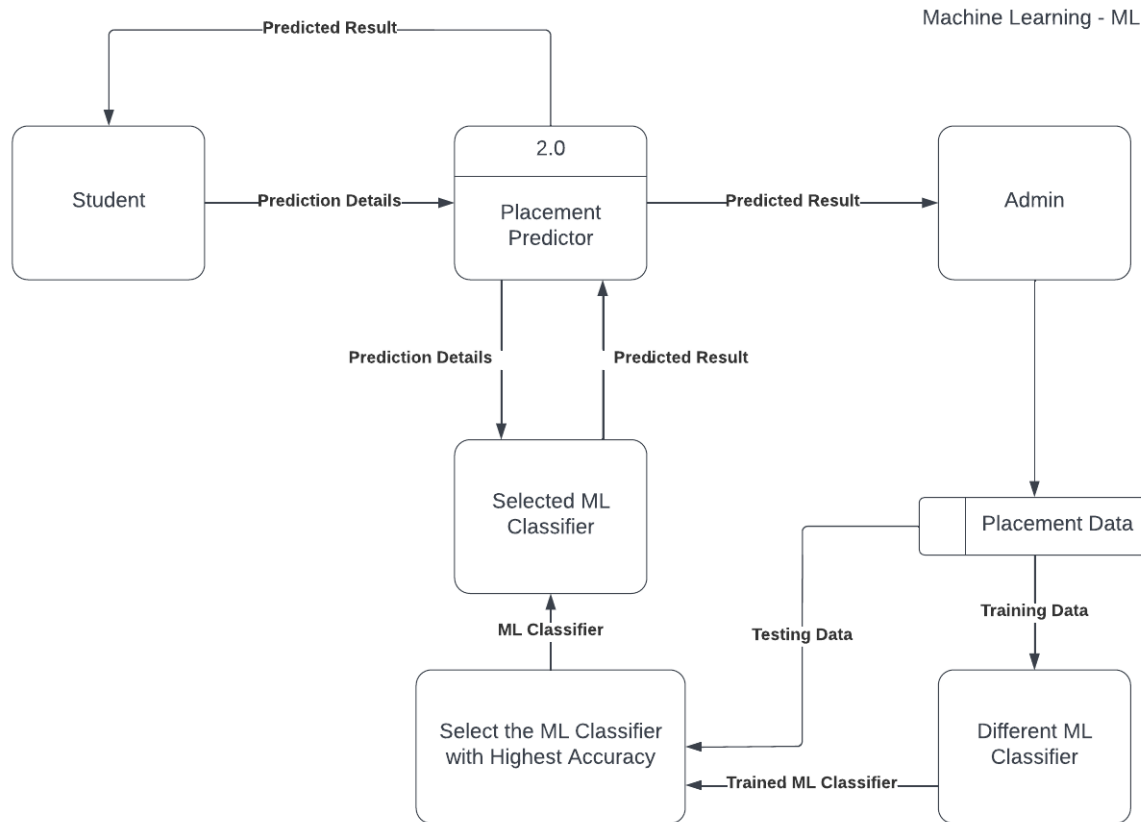


Figure 14: 2-level DFD

7.2 UML DIAGRAMS

It is the general-purpose modeling language used to visualize the system. It is a graphical language that is standard to the software industry for specifying, visualizing, constructing, and documenting the artifacts of the software systems, as well as for business modeling. Benefits of UML:

- Simplifies complex software design, can also implement OOPs like a concept that is widely used.
- It reduces thousands of words of explanation in a few graphical diagrams that may reduce time consumption to understand.
- It makes communication more clear and more real.

- It helps to acquire the entire system in a view.
- It becomes very much easy for the software programmer to implement the actual demand once they have a clear picture of the problem.

7.2.1 Sequence Diagram

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

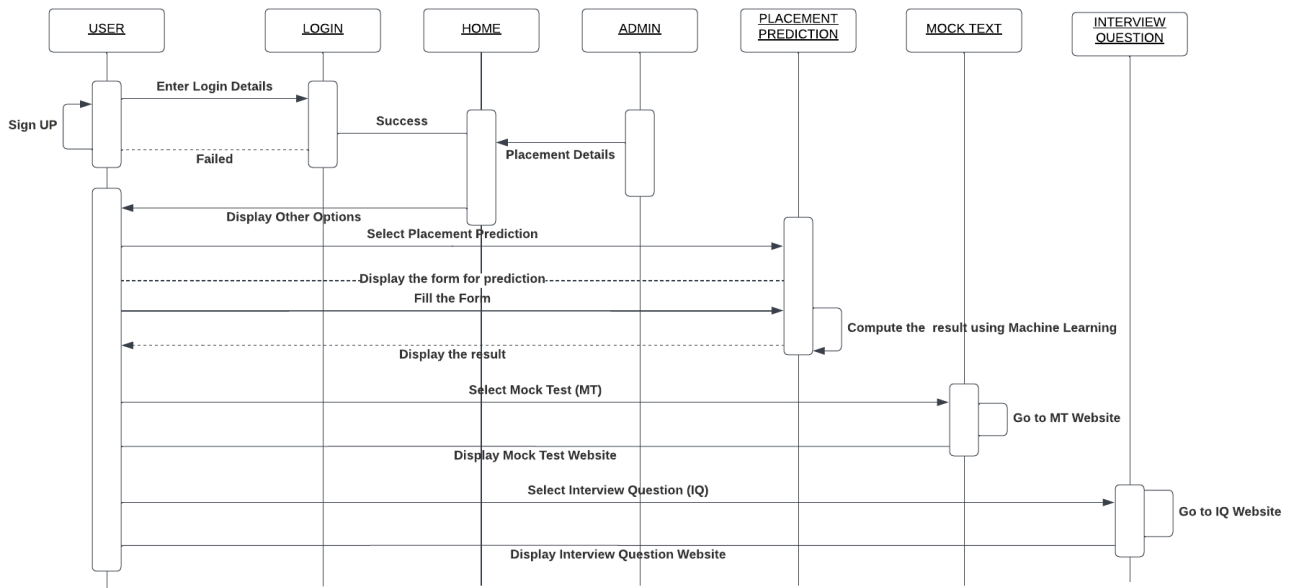


Figure 15: Sequence Diagram

7.2.2 Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of

an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

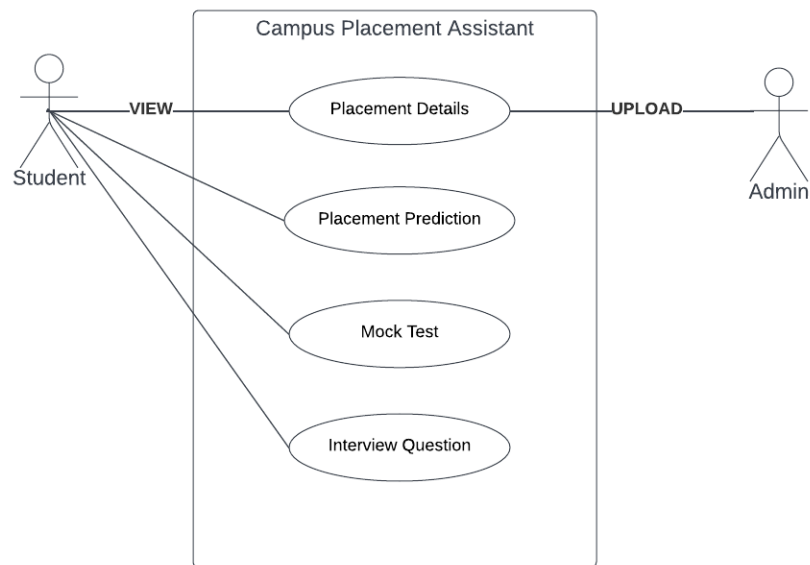


Figure 16: Use Case Diagram

7.2.3 Activiy Diagram

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. I can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system. An activity diagram is very similar to a flowchart.



Figure 17: Activity Diagram

7.2.4 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

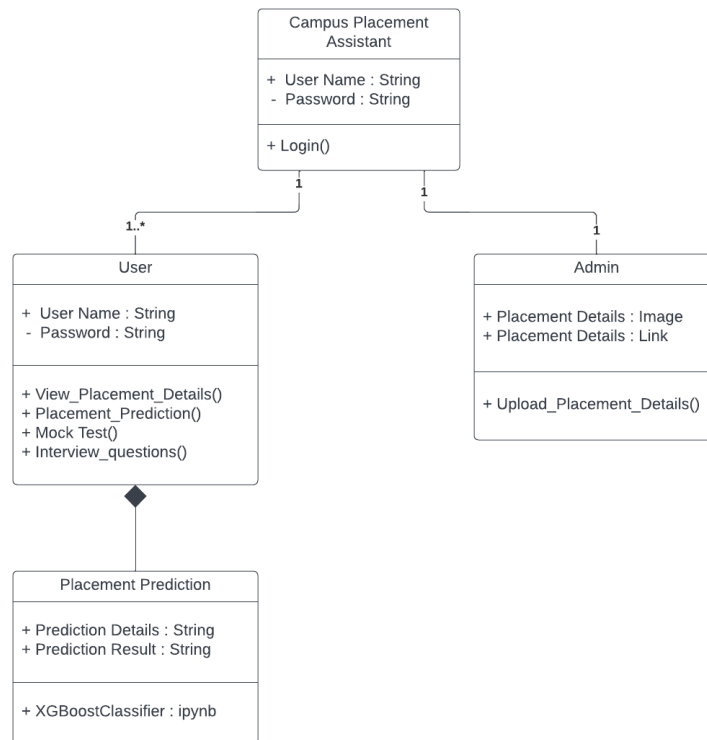


Figure 18: Class Diagram

CHAPTER 8

8 IMPLEMENTATION

8.1 Module Description

The system architecture depicts the overall working of our system .

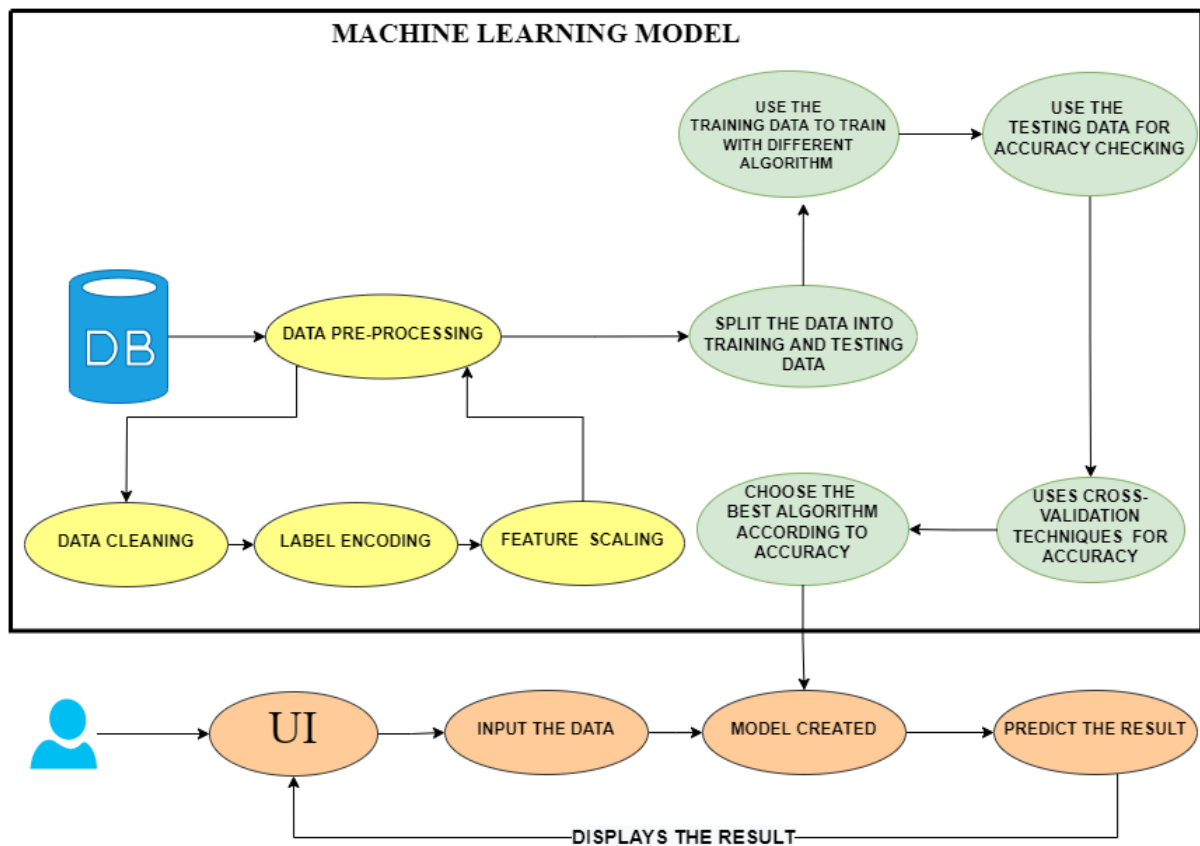


Figure 19: System Architecture

The modules in the system architecture are:

- **Database:** The database contains the dataset of the student placement information from the previous year data. This data is used to train the machine learning

model.

- **Data Pre-Processing:** Today's real world databases are highly susceptible to noisy, missing and inconsistent data due to their huge size and their lightly origin from multiple heterogeneous sources. Low quality data will lead to low quality Prediction results. Data prerocessing helps to improve the quality of the data and consequently of the Prediction results. Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format. There are several pre-processing techniques such as :

1. **Data Cleaning:** It can be applied to remove noise and correct inconsistency in the data. It cleans the data by filling the missing values, smoothing noisy data, identify or removing outlayers and removing inconsistencies.
 2. **Label Encoding:** Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.
 3. **Feature Scaling:** Feature Scaling or Standardization: It is a step of Data Pre Processing that is applied to independent variables or features of data. It basically helps to normalize the data within a particular range. Sometimes, it also helps in speeding up the calculations in an algorithm.
- **Splitting the Data:** It splits the data into training and testing data where it uses the training data to train the machine learning algorithms with 80% of the data and remianing 20% data for testing the accuracy of the machine learning algorithm.
 - **Training:** The training data is trained with the different machine learning algorithm such as Logistic regression, KNN Algorithm, SVM Algorithm, Decision Tree and XGBoost Classifier.
 - **Testing:** All the trained machine learning algorithms are being tested with test- ing data and accuracy of all the machine learning algorithms are compared to get the best machine learning algorithm.

- **Prediction:** The best machine learning algorithm is used to predict the result of the individual prediction of the input taken from the user interface.

8.2 Working

The working of the system is described as follows :

- Step 1: **Login:** Login to the campus placement assistant website by signing up with email or using google for sign up. After sign up with google or email it will show the respective email id used for sign up and the user can sign in to the Login using the password or google with security enabled.
- Step 2: **Home Page:** The Login directs to the home page where I can go to options such as Placement Details, Placement Prediction, Mock Test and Interview Question.
- Step 3: **Placement Details:** The placement details shows the current placement updates that are uploaded by the admin. Students can view all those details and can apply for those jobs and this allows a dedicated environment for the students to go for placement details updated by the admin.
- Step 4: **Placement Prediction:** The student can check the probability that they can get placed using placement predictor. The user details are transferred to deepnote after pressing the predict button, where the user details are used to predict the result by XGBoost classifier and the result is passed back to the placement prediction module as the person can get placed or not. If the person can't get placed the corresponding course details are displayed.
- Step 5: **Mock Test and Interview Question:** To get better placement student can go through these modules where they can practice with mock test and familiarize with interview questions.
- Step 6: **Logout:** The user can logout and the page will be redirected to the sign in page.

CHAPTER 9

9 TESTING

This section describes the testing procedure, starting with the testing methodology and then continuing with the tests performed on the system.

- Testing is a process of exceeding with the internet of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.
- A successful test is one that uncovers undiscovered errors.

9.1 Testing Methodology

The most natural and customary way of verifying any piece of work is just to operate it in some representative situation and verify whether its behavior is as expected. In general it is impossible to test it under all possible operating conditions. Thus it is necessary to find suitable test cases that provide enough evidence that the desired behavior will be exhibited in all remaining case. Testing is a critical activity in software engineering and should be performed as systematically as possible by stating clearly what result one expects from it and how one obtains these results. On the contrary, often in practice, testing is performed in an unstructured way without applying any criterion. There are two approaches to testing:

- White box testing

This is testing software using information about the internal structure of the software. It tests what the program does. The test is being carried out to check the internal structure of the software. The test is carried out successfully and the internal structure of the software meets the required criteria.

- Black box testing

Testing a piece of software as a black box (also called functional testing) means operating the software without relying on any knowledge of the way it has been designed and coded. It tests what it is supposed to do. The Programs run successfully against task assigned.

- Unit Testing

This is the first level of testing. In this different modules are tested against the specification produces during the design of the modules. Unit testing is done during the coding phase and to test the internal logic of the modules. It refers to the modules. It refers to the verification of single program module in an isolated environment. Unit testing first focuses on the modules independently of one of another to locate errors. After coding each dialogue is tested and run individually. All necessary coding were removed and it was ensured that all modules are worked, as the programmer would expect. The logical errors found were corrected so, by working all the modules independently and verifying the outputs of each module in the presence of staff, I observed that the program was functioning as expected. In unit testing – Module is tested to ensure that information properly flows into and out of the program under test – Local data structures are examined to ensure that data stored temporarily maintains its integrity during all steps in algorithm execution. Boundary condition is tested to ensure that module operates properly at boundaries established to limit or restrict processing. – All independent paths through the control structures are executed to ensure that all statements in the module have been executed at least once. – Error handling paths are also tested.

- Integration testing

Data can be lost across an interface: one module can be adversely affected on another; sub functions when combine may not produce the desired major functions. Integration testing is a systematic testing for constructing the program structure. Conducting the tests is to uncover errors associated within the interface. The objective is to take unit tested to modules and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the vast expenses of the entire program complicate the isolation of causes. Thus in the integration testing step, all the errors uncovered are corrected for the next testing steps.

- Low – Level modules are combined to form clusters .
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure.

- Alpha Testing

A series of acceptance tests were conducted to enable the users to validate requirements. The suggestions, along with the additional requirements of the end user were included in the project.

- Beta Testing

It is to be conducted by the end – user without the presence of the developer. It can be conducted over a period of weeks or month. Since it is a long time consuming activity, its result is out scope of this project report. But its result will help to enhance the product at a later time.

- Validation Testing

This provides final assurance that the software meets all the functional, behavioral and performance requirement. The software is completely assembled as a package. Validation succeeds when the software functions in a manner in which user wishes. Validation refers to the process of using software in live environment in order to find errors. During the course of validation the system failure may occur and sometimes the coding has to be hanged according to the requirement. Thus the feedback from the validation phase generally produces changes in software. Once the application was made of all logical and interface errors, inputting dummy data ensure that the software developed satisfied all the requirements of the user. This dummy data is known as test case.

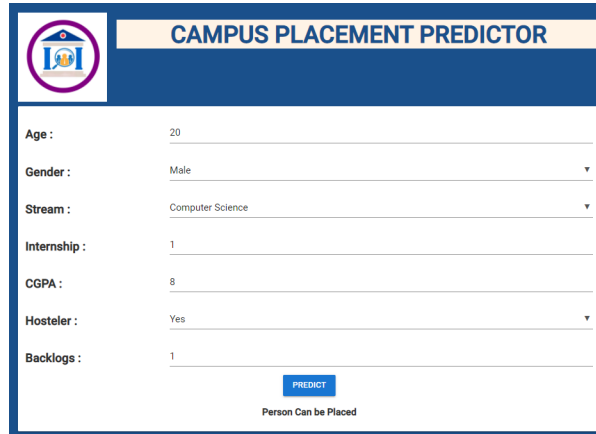
- Output Testing

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in a specific format. Asking the users about the format required by them, tests the output generated are considered into two ways. One is on screen and another is printed format. The output format on the screen found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, the output comes out as the specified requirement by the user. Hence output testing does not result in any correction in the system.

CHAPTER 10

10 RESULT

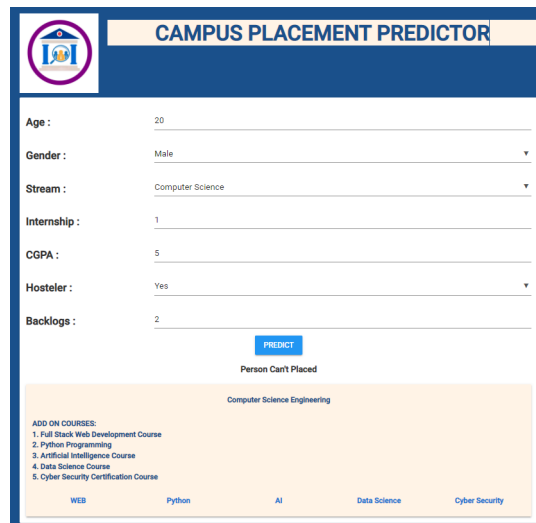
The placement predictor predicts that a student can be placed or not . **Placed**



The screenshot shows the 'CAMPUS PLACEMENT PREDICTOR' interface. The input fields are: Age: 20, Gender: Male, Stream: Computer Science, Internship: 1, CGPA: 8, Hosteler: Yes, and Backlogs: 1. A blue 'PREDICT' button is visible. Below the button, the text 'Person Can be Placed' is displayed.

Figure 20: Student Predicted to be Placed

Not Placed



The screenshot shows the 'CAMPUS PLACEMENT PREDICTOR' interface with the same input fields as Figure 20, but with Backlogs set to 2. The 'PREDICT' button is present. Below it, the text 'Person Can't Placed' is displayed. Further down, a list of 'ADD ON COURSES' is shown: 1. Full Stack Web Development Course, 2. Python Programming, 3. Artificial Intelligence Course, 4. Data Science Course, and 5. Cyber Security Certification Course. At the bottom, there are five buttons labeled WEB, Python, AI, Data Science, and Cyber Security.

Figure 21: Student Predicted to be not Placed

CHAPTER 11

11 PERFORMANCE AND EVALUATION

The campus placement assistant uses the placement predictor to predict whether a student can be placed or not and for prediction I have derived to XGBoost Classifier and to check the accuracy of the model confusion matrix has been used, the confusion matrix conveys the visual representation of the Actual VS Predicted values. It measures the performance of our Machine Learning classification model and looks like a table-like structure.

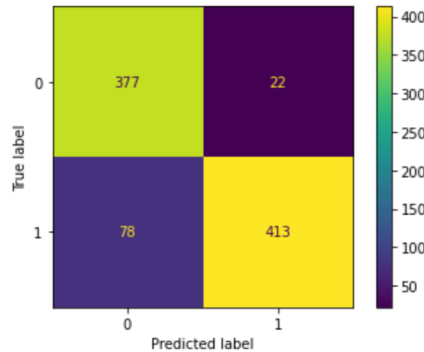


Figure 22: Confusion Matrix for XGBoost classifier

The different values of the Confusion matrix would be as follows:

- **True Positive:** (TP) = 377; meaning 377 positive class data points were correctly classified by the model
- **True Negative:** (TN) = 413; meaning 413 negative class data points were correctly classified by the model
- **False Positive:** (FP) = 22; meaning 22 negative class data points were incorrectly classified as belonging to the positive class by the model
- **False Negative:** (FN) = 78; meaning 78 positive class data points were incorrectly classified as belonging to the negative class by the model

The graphs such as Area Under Curve AUC and Average Precision AP has been found :
AUC: The AUC indicates whether your model can correctly rank examples. The AUC is the probability that a randomly selected positive example has a higher predicted probability of being positive than a randomly selected negative example. The AUC

is calculated as the area underneath a curve that measures the trade off between true positive rate (TPR) and false positive rate (FPR) at different decision thresholds d .

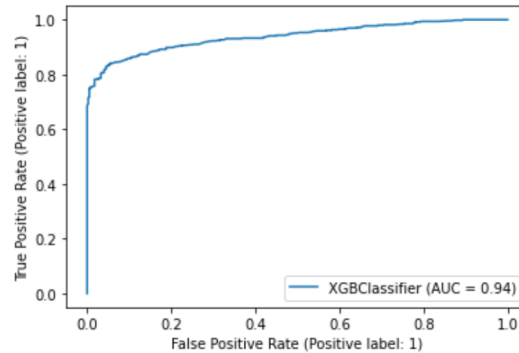


Figure 23: AUC for XGBoost classifier

AP: Average precision indicates whether your model can correctly identify all the positive examples without accidentally marking too many negative examples as positive. Thus, average precision is high when your model can correctly handle positives. Average precision is calculated as the area under a curve that measures the trade off between precision and recall at different decision thresholds.

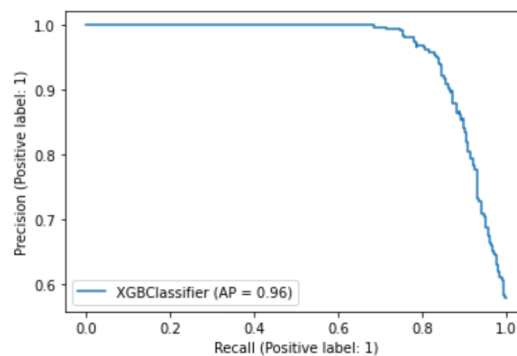


Figure 24: AP for XGBoost classifier

The result that I obtained using XGBoost classifier has got about 89% of accuracy.

CHAPTER 12

12 SCREENSHOT

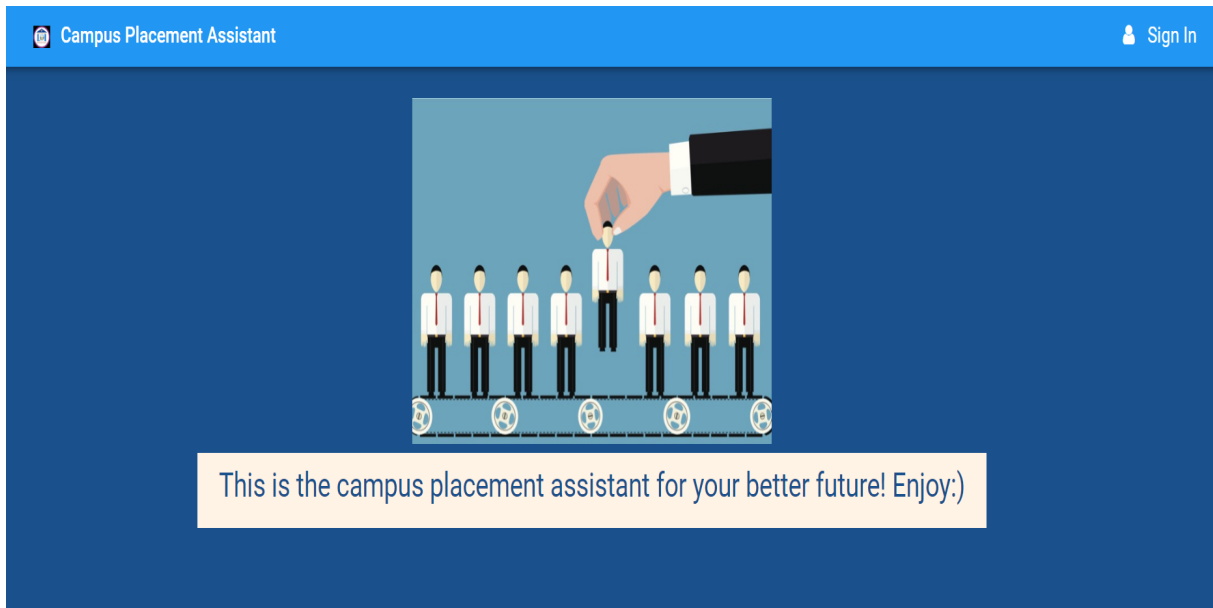


Figure 25: Login Page

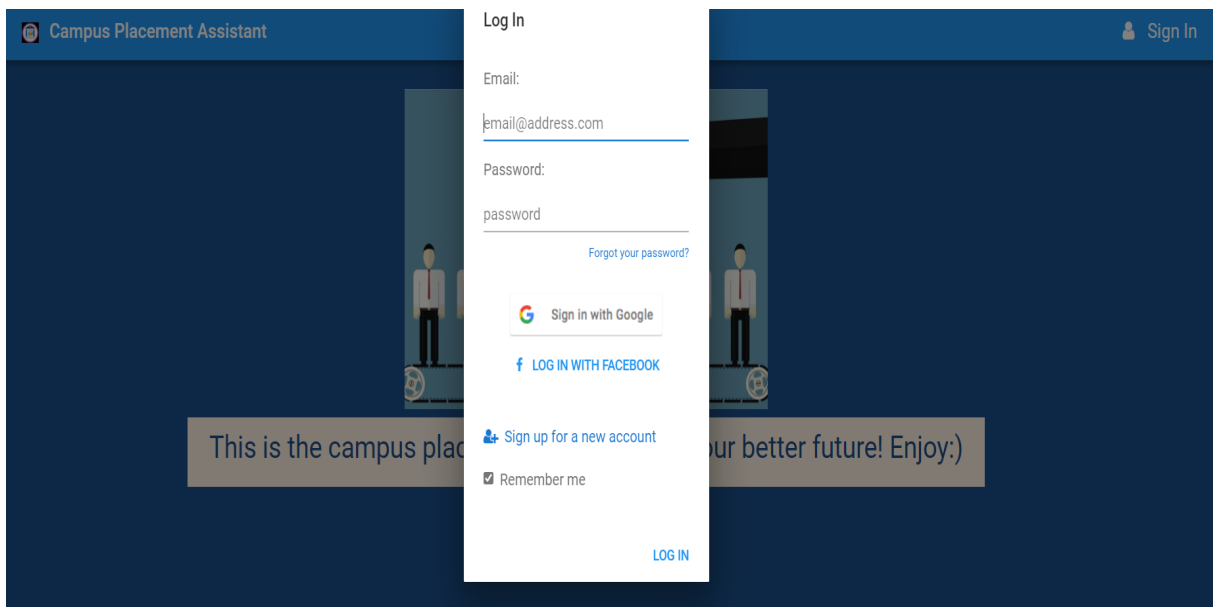


Figure 26: Sign Up Page

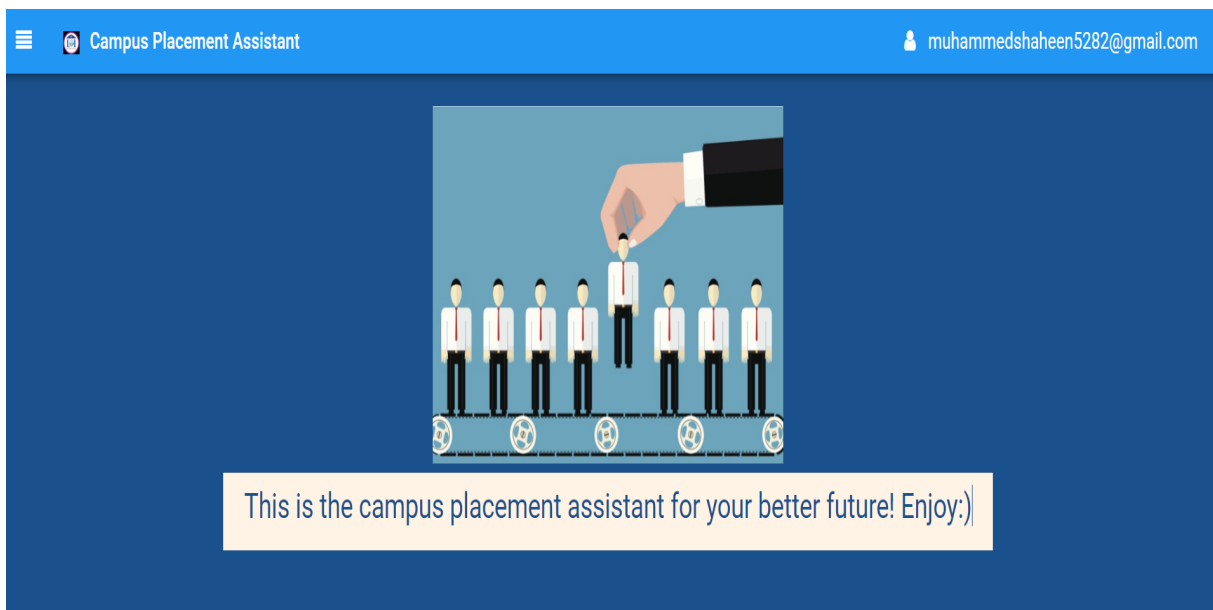


Figure 27: Home Page

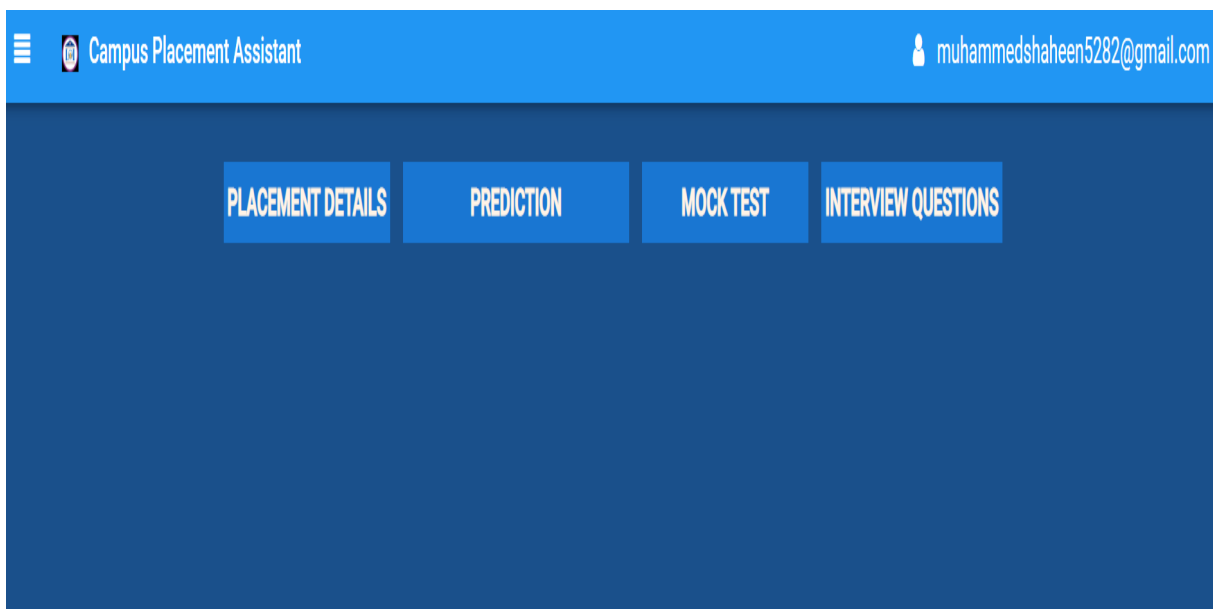


Figure 28: Other options



Figure 29: Placement Details

The screenshot displays the "CAMPUS PLACEMENT PREDICTOR" form. It includes a logo on the left and a title bar. The form fields are: "Age" (text input), "Gender" (dropdown), "Stream" (dropdown), "Internship" (text input), "CGPA" (text input), "Hosteler" (dropdown), and "Backlogs" (text input). A blue "PREDICT" button is located at the bottom right of the form area.

Figure 30: Placement Prediction

CHAPTER 13**13 CONCLUSION**

The campus placement assistant will help the student to get all the placement support in a nutshell that include placement details from the admin, placement prediction using the best machine learning algorithm, mock test and interview question. In the case of Placement Prediction a point-by - point analysis was performed based on specific machine learning algorithms for predicting the placement probability and XGBoost classifier has the best accuracy with 89%. The student dataset including academic and selection subtleties is clearly a possible hotspot from the exam for forecasting future selection possibilities. Such forecast will empower students to consider their skills and develop according to their needs. This methodology also helps to prepare appropriate procedures and strengthen the recruitment statistics for future years in the scholastic arrangement of an establishment. The placement officer of a campus can also benefit from this system to have an organised platform for placement.

References

- [1] Performance Comparison of Machine Learning Algorithms for Prediction of Students' Social Engagement <https://doi.org/10.1109/ICCMC51019.2021.9418260>
- [2] An EDM Approach to the Analysis of Students' Engagement in Online Courses from Constructs of the Transactional Distance <https://doi.org/10.1109/ICALT.2016.17>
- [3] Using Ranking and Multiple Linear Regression to Explore the Impact of Social Media Engagement on Student Performance <https://doi.org/10.1109/ICALT.2016.140>
- [4] Student Academic Performance and Social Behavior Predictor using Data Mining Techniques <https://doi.org/10.1109/CCAA.2017.8229794>
- [5] Student Engagement Predictions in an e-Learning System and Their Impact on Student Course Assessment Scores <https://doi.org/10.1155/2018/6347186>
- [6] Cluster and Logistic Regression Distribution of Students' Performance by Classification

gr