Campus Placement Prediction using supervised Machine Learning Techniques

Mohammed Raamizuddin[1*], Nagam Vinusha[2], and Arravelli Tejaswi[3]

[1]Department of Computer Science and Engineering, SR University, Warangal, Telangana, India.
[2]Department of Computer Science and Engineering, SR University, Warangal, Telangana, India.
[3]Department of Electronics and Communications Engineering SR University, Warangal, Telangana, India.

*Email: raamiz2510@gmail.com

Abstract: A placement predictor is to be designed to calculate the possibility of a student being placed in a company, subject to the criterion of the company. The placement predictor takes many parameters which can be used to assess the skill level of the student. While some parameters are taken from the university level, others are obtained from tests conducted in the placement management system itself. Combining these data points, the predictor is to accurately predict if the student will or will not be placed in a company. Data from past students are used for training the predictor. . This will always be helpful to both the students, as well as the institution. In this study, the objective is to analyse previous year's student's data and use it to predict the placement chance of the current students. This model is proposed with an algorithm to predict the same. This proposed model is also compared with other traditional classification algorithms such as Decision tree and Random forest with respect to accuracy, precision and recall. From the results obtained it is found that the proposed algorithm performs significantly better in comparison with the other algorithms mentioned.

Keywords: Classifications, Dataset, Machine learning, Placement, Decision tree, Random Forest.

1. INTRODUCTION:

Placements are considered to be very important for each and every college. The basic success of the college is measured by the campus placement of the students. Every student takes admission to the colleges by seeing the percentage of placements in the college. Hence, in this regard the approach is about the prediction and analyses for the placement necessity in the colleges that helps to build the colleges as well as students to improve their placements. In Placement Prediction system predicts the probability of a undergrad students getting placed in a company by applying classification algorithms such as Decision tree and Random forest. The main objective of this model is to predict whether the student he/she gets placed or not in campus recruitment. For this the data consider is the academic history of student like overall percentage, backlogs, credits. The algorithms are applied on the previous years data of the students.

We aim to develop a placement predictor as a part of making a placement management system at college level which predicts the probability of students getting placed and helps in uplifting their skills before the recruitment process starts. We are using machine learning for the placement prediction. Logistic Regression, Decision tree and Random Forest to classify students into appropriate clusters and the result would help them in improving their profile.And accuracy of respected algorithms are noted and With the comparison of various machine learning techniques, this would help both recruiters as well as students during placements and related activities.

In this paper we 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, LogicalReasoning scores, Verbal scores, Programming scores, CGPA, No. of Internships attended, and History of backlogs. The placement prediction is done by machine learning using Logistic Regression, Random Forest, Decision tree for both placement and package prediction.

2. PROBLEM STATEMENT:

The general Placement Prediction System considers only academic performances in order to predict whether a student can be placed or not. Judging the student based only on his academic performances would be unfair for the student, since a student could be having good aptitude, technical and communication skills but unfortunately might not be good in academic performances. It would wrong to judge a student based only on his academic performances, since Predicting the placement of a student needs a lot of parameters to be considered. But in order to get selected in campus interview, the student must be good in technical and aptitude skills. Of course academic performances are important but don't hold the highest importance in the outcome of student placement.

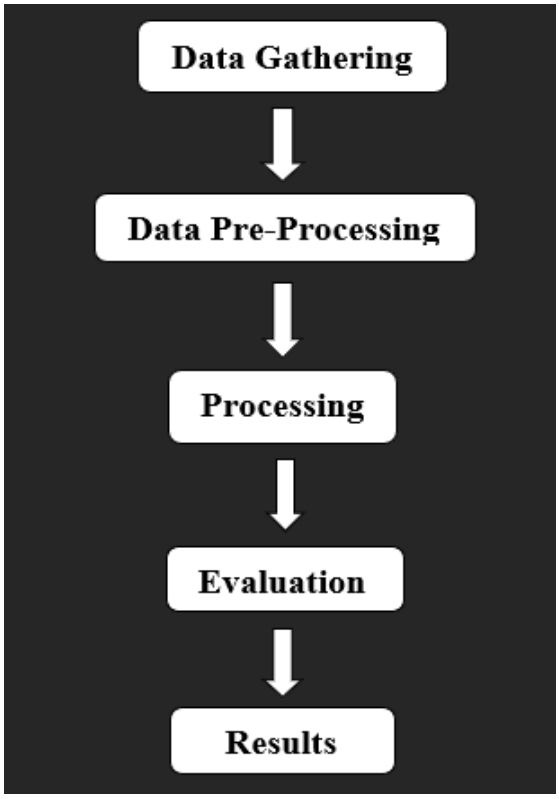


Figure 1 - Processing Steps for Machine Learning for Campus Placement prediction.

3. DATASET AND ATTRIBUTES:

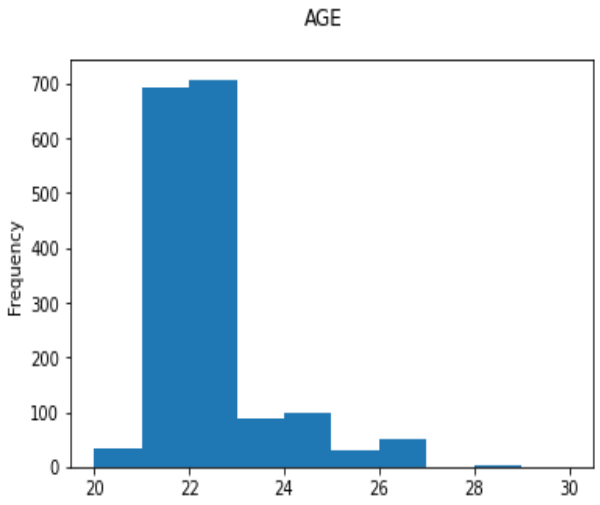
We collected the data from various colleges. We use only 8 attributes to predict whether the student will get placed or not with the package specified.

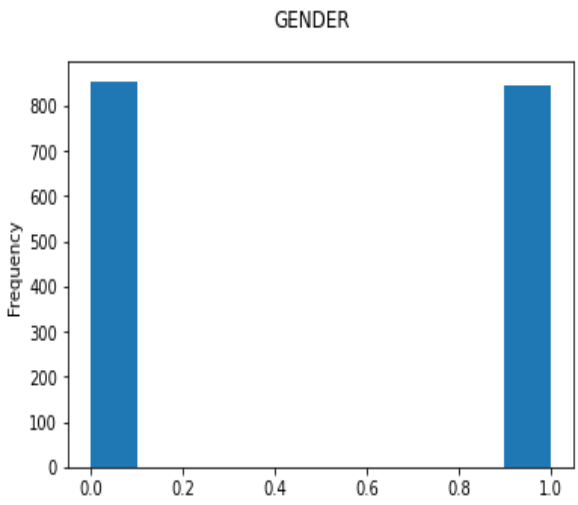
Features :

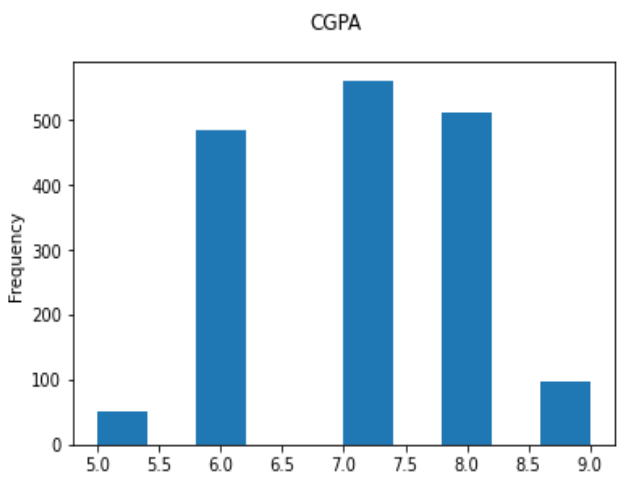
- Age: age in years
- Gender : Gender (1 = male; 0 = female)
- Stream : Branch of Study
    - Value 1: Computer Science Engineering.
    - Value 2: Electronics and Communication Engineering.
    - Value 3: Electrical and Electronics Engineering.
    - Value 4: Mechanical Engineering.
    - Value 5: Civil Engineering.
    - Value 6: Information Technology.
- CGPA: Cumulative Grade Point (Percentage %)
    - Value between 1 – 10.
- Coding Skills: Assessment of coding.
    - Value between 1 – 10.
- Communication Skills: Assessment of Communication.
    - Value between 1 – 10.
- Internships: Number of internships completed by the candidate.
    - Value between 1 – 2.
- History Of Backlogs: Backlog history of candidate.
    - Value 1: Yes.
    - Value 2: No.

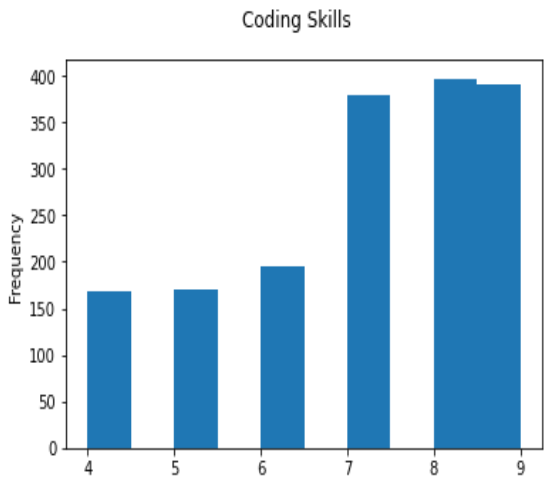
Labels:

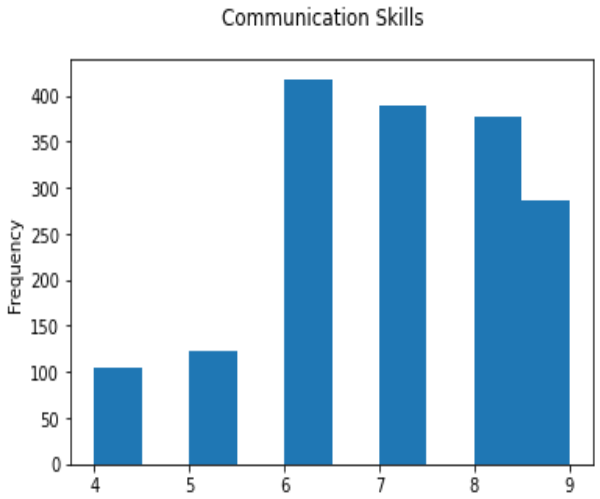
- Placed Or Not : Occurrence of Placement (1 = yes; 0 = no)
- Package Offered (LPA): Package Specification.

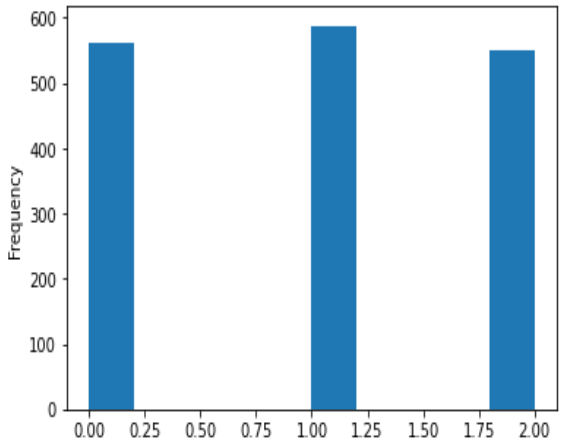

AGE
Frequency
0
100
200
300
400
500
600
700
20
22
24
26
28
30


GENDER
Frequency
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0


CGPA
Frequency
0
100
200
300
400
500
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0


Coding Skills
Frequency
0
50
100
150
200
250
300
350
400
4
5
6
7
8
9


Communication Skills
Frequency
0
50
100
150
200
250
300
350
400
4
5
6
7
8
9


Internships
Frequency
0
100
200
300
400
500
600
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

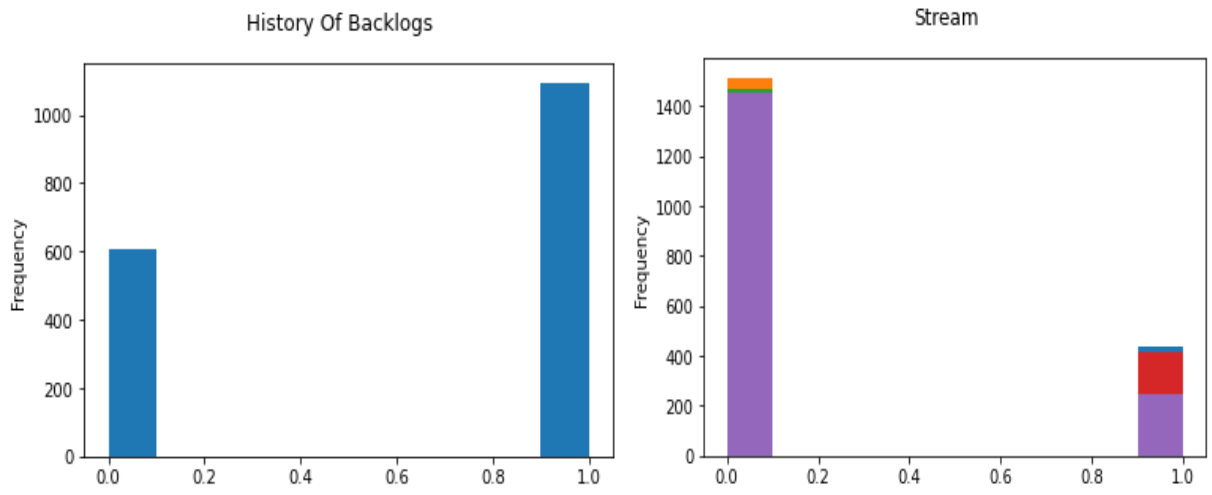


Figure 2 - Visualizing attributes of the dataset.

Serial No	Age	Gender	Stream	CGPA	Coding Skills	Communication Skills	Internships	HistoryOfBacklogs	PlacedOrNot	Package Offered (LPA)
1	22	Female	Electronics And Communication	8	9	6	1	0	0	0
2	21	Female	Computer Science	7	8	9	0	1	1	7
3	22	Male	Information Technology	6	6	8	2	1	1	4
4	21	Female	Information Technology	8	9	9	2	0	1	12
5	22	Male	Mechanical	8	7	7	1	0	1	7
6	22	Female	Electronics And Communication	6	6	5	2	1	1	5
7	21	Male	Computer Science	7	7	8	0	1	0	0
8	21	Female	Information Technology	7	8	9	2	1	1	8
9	21	Male	Computer Science	6	5	6	1	1	0	0
10	21	Male	Computer Science	6	6	5	1	1	1	4

Figure 3 – Dataset Insight.

4. DATA PRE-PROCESSING :

Data pre-processing 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.

5.1 Attribute selection:

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

5.2 Cleaning missing values:

In some cases, the dataset contains 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.

5.3 Training and Test data:

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

5.4 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) are much higher than the value in another column (y), (x2-x1) squared will give a far greater value than (y2-y1) squared. So clearly, one square distinction dominates over the other square distinction. In the machine learning equations, the square difference with the lower value in comparison to the far greater value will almost be treated as if it does not exist. We do not want that to happen. That is why it's necessary to transform all our variables into the same scale. There are several ways of scaling the data. One way is called Standardization which may be used. For every observation of the selected column, our program will apply the formula of standardization and fit it to a scale.

5.5 Evaluating the data:

Correlation Matrix:

A correlation matrix is simply a table that displays the correlation. The measure is best used in variables that demonstrate a linear relationship between each other. The fit of the data can be visually represented in a scatterplot. coefficients for different variables.

How it is calculated?

A correlation matrix is a table showing correlation coefficients between sets of variables. Each random variable (X_i) in the table is correlated with each of the other values in the table (X_j)... The diagonal of the table is always a set of ones because the correlation between a variable and itself is always 1. Let's perform the Correlation matrix to understand the relation between the dependent variable and the independent variable and within the independent variable.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1.000	-0.196	0.204	0.019	-0.156	0.084	-0.640	0.181	-0.426	0.498	-0.164	0.425	-0.152	0.034	-0.533
1	-0.196	1.000	-0.108	0.197	0.013	0.080	0.165	0.320	0.389	0.003	0.232	-0.083	-0.088	0.204	0.279
2	0.204	-0.108	1.000	0.147	0.382	0.485	-0.165	0.047	0.412	0.210	0.487	-0.163	0.104	-0.155	-0.078
3	0.019	0.197	0.147	1.000	-0.082	0.435	-0.311	-0.082	0.243	0.034	0.282	-0.636	0.436	-0.063	-0.312
4	-0.156	0.013	0.382	-0.082	1.000	0.153	0.296	0.330	-0.023	-0.113	-0.060	-0.171	-0.031	0.286	0.087
5	0.084	0.080	0.485	0.435	0.153	1.000	-0.042	0.431	0.178	0.206	0.299	-0.299	0.211	-0.056	0.018
6	-0.640	0.165	-0.165	-0.311	0.296	-0.042	1.000	0.147	0.122	-0.408	0.176	0.096	0.350	0.160	0.409
7	0.181	0.320	0.047	-0.082	0.330	0.431	0.147	1.000	-0.144	0.085	-0.166	0.258	-0.167	-0.005	-0.154
8	-0.426	0.389	0.412	0.243	-0.023	0.178	0.122	-0.144	1.000	-0.035	0.671	-0.446	0.141	-0.036	0.133
9	0.498	0.003	0.210	0.034	-0.113	0.206	-0.408	0.085	-0.035	1.000	-0.167	0.222	-0.030	-0.027	-0.098
10	-0.164	0.232	0.487	0.282	-0.060	0.299	0.176	-0.166	0.671	-0.167	1.000	-0.276	0.417	0.289	-0.029
11	0.425	-0.083	-0.163	-0.636	-0.171	-0.299	0.096	0.258	-0.446	0.222	-0.276	1.000	-0.423	0.015	-0.123
12	-0.152	-0.088	0.104	0.436	-0.031	0.211	0.350	-0.167	0.141	-0.030	0.417	-0.423	1.000	0.079	-0.045
13	0.034	0.204	-0.155	-0.063	0.286	-0.056	0.160	-0.005	-0.036	-0.027	0.289	0.015	0.079	1.000	-0.147
14	-0.533	0.279	-0.078	-0.312	0.087	0.018	0.409	-0.154	0.133	-0.098	-0.029	-0.123	-0.045	-0.147	1.000

Figure 4 - Correlation Matrix.

5. ALGORITHMS:

We use different machine learning model to solve our classification problem:

1. Logistic Regression.
2. Decision tree Classifier.
3. Random Forest Classifier.
4. Linear regression.
5. Decision tree regressor.
6. Random Forest Regressor.

So, let us make our data ready for training and testing our machine learning model.

Logistic Regression:

Logistic regression is a supervised algorithm for study classification. The likelihood of a destination variable was predicted. The nature of the target or dependent variable is dichotomous, meaning that only two possible classes are available.

Steps for Logistic Regression:

Step 1: Data Preprocessing step.

Step 2: Fitting Logistic Regression to the training set.

Step 3: Predicting the test Result.

Step 4: Test accuracy to the result.

Step 5: visualizing the test set result.

Decision Tree Classifier:

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules, and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

Random forest Regressor and Classifier:

Random forest is used for both regression and classification-based applications. This algorithm is flexible and easy to use. Most of the times this algorithm gives accurate results even without hyper tuning the parameters. It builds many decision trees which on merging forms as a forest. While building the decision trees, adds more randomness to the model. This algorithm searches for the best feature in the random subset of features, which results in the formation of a better model.

With the aid of the following steps, we can understand how the Random Forest algorithm works:

Step 1 − First, select random samples from a particular dataset.

Step 2 - Next, for each sample this algorithm will build a decision tree. Then every decision tree will predict the result.

Step 3 − Every predicted result will be voted in this step.

Step 4 − Finally, the final prediction result will be selected as the most voted prediction result.

Linear Regression:

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering, and the number of independent variables being used.

Assumptions for Multiple Linear Regression:

- A linear relationship should exist between the Target and predictor variables.
- The regression residuals must be normally distributed.
- MLR assumes little or no multicollinearity (correlation between the independent variable) in data.

Decision Tree Regressor:

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

6. RESULTS:

PLACEMENT PREDICTION :

LOGISTIC RERESSION ACCURACY				
NUMBER OF ITERATIONS				
50	100	300	500	1000
0.92	0.92	0.92	0.92	0.92

Figure 8 – Logistic Regression Result.

DECISION TREE CLASSIFIER ACCURACIES					
CRITERION	DEPTH OF TREE				
	5	6	8	10	D=16,14
Entropy	0.971764	0.971764	0.971764	0.967058	0.96
Gini	0.971764	0.971764	0.96	0.96	0.950588

Figure 9 – Decision tree Classifier Result**.**

RANDOM FOREST CLASSIFIER ACCURACIES					
CRITERION	ESTIMATORS (TREES)				
	5	10	30	50	100
Entropy	0.974117	0.974117	0.97647	0.976471	0.97647
Gini	0.97647	0.971764	0.97647	0.97647	0.974117

Figure 10 –Random Forest Classifier Result**.**

PACKAGE PREDICTION :

LINEAR REGRESSION ERRORS	
RMSE	1.904476021
MSE	3.627028914
MAE	1.591545194

Figure 11 – Linear Regression Result**.**

DECISION TREE REGRESSOR ERRORS	
RMSE	1.588204108
MSE	2.52239229
MAE	0.93537415

Figure 12 – Decision tree regressor Result**.**

RANDOM FOREST REGRESSOR ERRORS	
RMSE	1.162824152
MSE	1.35216001
MAE	0.809730851

Figure 13 – Random Forest regressor Result**.**

Algorithm	Scores
Logistic Regression	92%
Decision tree Classifier	97%
Random Forest Classifier	97%
Linear Regression	62%
Decision tree regressor	84%
Random Forest regressor	86%

Table 1 – Scores.

7. CONCLUSION:

So, from the above table of scores, we can predict that with Logistic Regression Model and with shows best accuracy result around 92% among all the models respectively. Yet, we attempt to make our other model exactness more precise yet at this stage, we can see that our concern scoring is around 86% which is given by Random Forest regressor Model for package prediction and 97% for placement prediction using random forest classifier.

The campus placement activity is incredibly a lot of vital as institution point of view as well as student point of view. In this regard to improve the student's performance, a work has been analyzed and predicted using the classification algorithms Decision Tree and the Random Forest algorithm to validate the approaches. The algorithms are applied on the data set and attributes used to build the model. The accuracy obtained after analysis for Decision tree is 84% and for the Random Forest is 86%. Hence, from the above said analysis and prediction it's better if the Random Forest algorithm is used to predict the placement results with package.

8. REFERENCES:

[1]. Mangasuli Sheetal B, Prof. Savita Bakare "Prediction of Campus Placement Using Data Mining Algorithm Fuzzy logic and K nearest neighbour" International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 6, June 2016.

[2]. Ajay Shiv Sharma, Swaraj Prince, Shubham Kapoor, Keshav Kumar "PPS-Placement prediction system using logistic regression" IEEE international conference on MOOC,innovation and Technology in Education(MITE), December 2014.

[3]. Jai Ruby, Dr. K. David "Predicting the Performance of Students in Higher Education Using Data Mining Classification Algorithms - A Case Study" International Journal for Research in Applied Science & Engineering Technology (IJRASET) Vol. 2, Issue 11, November 2014.

[4]. Ankita A Nichat, Dr.Anjali B Raut "Predicting and Analysis of Student Performance Using Decision Tree Technique" International Journal of Innovative Research in Computer and Communication Engineering V0l. 5, Issue 4, April 2017.

[5]. Oktariani Nurul Pratiwi "Predicting Student Placement Class using Data Mining" IEEE International Conference 2013.

[6]. Ajay Kumar Pal and Saurabh Pal, "Classification Model of Prediction for Placement of Students", I. J. Modern Education and Computer Science, 2013, 11, 49-56.

[7]. Ravi Tiwari and Awadhesh Kumar Sharma, "A Data Mining Model to Improve Placement", International Journal of Computer Applications (0975 – 8887) Volume 120 – No.12, June 2015.

[8]. Ms.sonal patil, Mr.Mayur Agrawal, Ms.Vijaya R. Baviskar "Efficient Processing of Decision Tree using ID3 & improved C4.5 Algorithm", International Journal of Computer Science and Information Technologies, Vol. 6 (2) , 2015, 1956-1961.