

Recruitment System with Placement Prediction

Suraj Gupta¹

Information Technology
MCT Rajiv Gandhi Institute of Technology
Mumbai, India
sg198374@gmail.com

Yuvraj Haryan³

Information Technology
MCT Rajiv Gandhi Institute of Technology
Mumbai, India
yuvrajharyan@gmail.com

Atif Hingwala²

Information Technology
MCT Rajiv Gandhi Institute of Technology
Mumbai, India
atifhingwala786@gmail.com

Swapnil Gharat⁴

Information Technology
MCT Rajiv Gandhi Institute of Technology
Mumbai, India
swapnil.gharat@mctrgit.ac.in

Abstract: 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.

Keywords: Placement Probability, Recruitment System, Random Forest Regressor.

1. INTRODUCTION

The organizations face difficulties in hiring students during the placement drives. The earlier system done manually by training and placements department makes the process slow and leads to inconsistencies and ambiguities. In addition, it is very difficult to maintain coordination between students and companies. Recruitment system is an online application for organization as well as company usage. To manage the student information regarding placements, the college Training and Placement Officer (TPO) can use this application. It can be used as tool to analyze candidate's performance and predict his placement probability prior to the placement sessions. Online training and placement portals are available that allow filtering, resume building ^{[1][2]}. But they don't have any tool for candidate's performance analysis ^{[3][4]}. And do not provide us with any

placement probability. Classification algorithms like K-means and SVM have been implemented earlier but fails to provide a continuous value for placement probability [5][6][7][8]. Also, data mining techniques implement classification generating a discrete value and hence fail to provide accurate probability [9][10][11][12][13]. Fuzzy logic and KNN algorithm have been used for prediction but these provide answers which are frequent and closest to the query [14][15].

The proposed system provides the facility of maintaining the details of the students and gets the requested list of candidates for the company who would like to recruit the students based on given skill. The aim of our project is to reduce maximum chances of errors in the manual work and save time as well as to make the students aware of their strengths and weaknesses for better placement opportunities.

Hence, this Recruitment system finds scope in increasing the efficiency of prediction.

2. BACKGROUND

Random Forest Algorithm:

Random Forest is a supervised learning technique in Machine learning. Problems related to Classification and Regression in ML can be addressed using it. **“Random Forest is a model which divides the dataset into various subsets and builds decision trees on their basis. Then uses the average of the subsets to improve the prediction accuracy.”** The accuracy of the model is directly proportional to the number of trees. Large number of trees also avoid the problem of overfitting.

Random Forest pseudocode:

1. Select “**k**” features randomly from total “**m**” features (Where **k** << **m**)
2. Calculate the node “**d**”, among the “**k**” features using the best split point.

3. Using the **best split**, split the node into **sub-nodes**.
4. Until “**l**” number of nodes has been reached, repeat **1 to 3** steps.
5. To create “**n**” number of trees and build the Forest, repeat steps **1 to 4** for ‘**n**’ times.

Flask API

Flask is a web framework which provides you with libraries, tools and technologies that help you in web application building. This web application may include a blog, a wiki, just some web pages or even a big commercial website. Micro-framework categories have Flask as their sub-part. Micro-frameworks are normal framework that are independent or just a little dependent on external libraries. Extensions are supported in Flask which allow application features to be added and implemented as if they were in Flask itself.

3. PROPOSED SYSTEM

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 –

- Real-time placement prediction system(dynamic)

- Probable candidate list generator(static).

Phase 1: Real-time placement prediction system(dynamic):

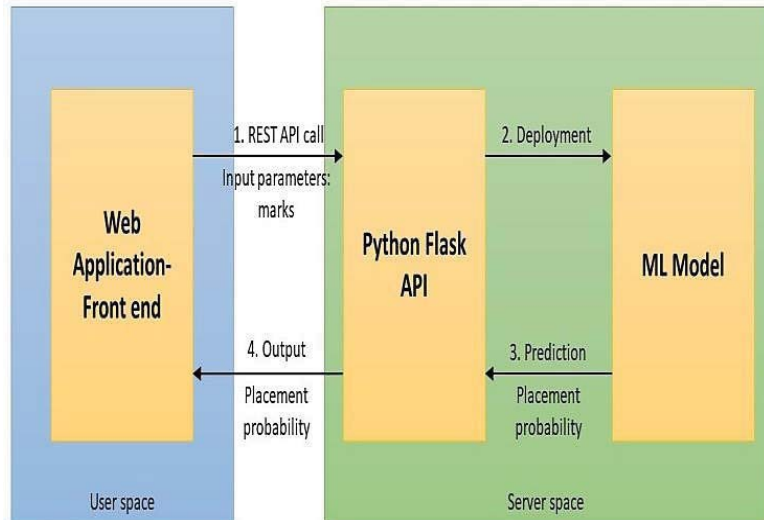


Figure 1: Dynamic predictor

This phase is for walk-in candidates where no previous record of him is available in the system.

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.

Phase 2: Probable candidate list generator(static):

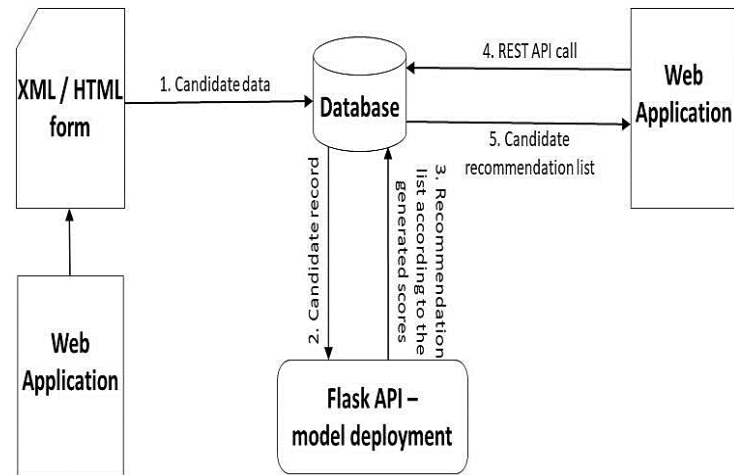


Figure 2: Static predictor

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.

4. RESULTS

Real-time placement prediction:

Figure 3: Dynamic Prediction

In Figure3, the parameters required for dynamic Prediction is the Name, SSC percentage, HSC, Percentage, CGPA, gender (1- Male and 0- Female) and algorithm used i.e. Random Forest Algorithm. Based on these the placement probability is generated. Mostly companies recruit maintaining the 1:1 ratio of males and females hence gender would be useful for company recruiters.

Probable Candidate list generator(static):

Figure 4: Static Prediction Parameters

In figure 4, the parameters, department and skills are used to sort the candidate data in database. Now the algorithm is fitted on selected candidates and accordingly the recommendation list is generated as shown in figure 5.

Navbar Home Register Dynamic Predictor Static Predictor									
	NAME	TENTH	TWELTH	CGPA	GENDER	DEPARTMENT	SKILLS	PLACEMENT_PROBABILITY	
0	ANUSHKA SHETTY	88.00	75.00	8.10	0	IT	C	1.00	
1	SANJAN PARAB	78.00	77.00	8.00	0	IT	C , Java , HTML	1.00	
2	URMILA DOSHI	86.50	64.20	7.20	0	IT	C , C++ , Java	1.00	
3	Yuvraj Haryan	91.00	81.90	9.28	1	IT	Python , Java , C , ML	0.98	
4	JHONY DSOUZA	98.00	68.00	9.50	1	IT	Python, C, HTML, CSS	0.98	
5	PREMA MISHRA	87.00	81.00	9.01	0	IT	C , C++ , Java	0.98	
6	GANPATI RAMAN	76.50	97.70	9.30	1	IT	Python , Java , C , C++	0.97	
7	Atif hingwala	90.00	81.85	9.00	1	IT	Python, C , ML	0.97	
8	PUNARNAVI	69.00	59.00	7.60	0	IT	Java , C	0.89	
9	VAJAKATHALI PADRIGUE	78.00	75.00	5.90	0	IT	UI/UX, C , Java	0.70	
10	SAMRUDDHI PATIL	69.70	47.00	7.20	0	IT	C , Python	0.66	
11	JOBIS JOHNSON	79.33	78.33	8.00	1	IT	C , C++ , HTML , CSS	0.60	
12	APPUN SELVA	66.00	82.00	6.80	1	IT	ML,Python, C	0.37	
13	SURAJ	86.00	77.00	8.30	1	IT	C,Python,MySQL,PHP	0.31	
14	PANDIYAN	78.00	93.20	7.50	1	IT	Rea C US, JavaScript	0.26	
15	SANKET WANI	82.00	61.00	6.80	1	IT	C , Python , AmgilarJS , HTML	0.25	
16	MODI	68.00	75.00	8.30	1	IT	ML,Python, C	0.20	
17	SIDDHESH NAIK	84.20	73.40	8.00	1	IT	C , Java , Python , HTML	0.15	
18	RAJ HEDAU	64.00	80.00	9.20	1	IT	C , Python , ML , Java	0.05	
19	VEERA CHAUDHARY	54.00	76.00	7.98	0	IT	C , Java , JavaScript	0.00	

Figure 5: Static Prediction Results

5. CONCLUSION

In the existing system the recruiters possess only the candidate information mentioned in their resume. It uses many algorithms such as SVM, Fuzzy Logic, KNN and K-means but fails to provide continuous values for student placement probability.

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.

Companies generally look for candidates with certain skill sets such as Python, C etc. The proposed system also provides the static prediction which is used by companies to bifurcate students based on their skills and domain. The recruiter can generate the recommendation list based on the company criteria and searching, sorting can also be performed. Alumni data can also be maintained.

6. REFERENCES

- [1] Jidnyasa Raut, Komal Patil, Payal Gothi, Riddhi Kamat, Prof. Nileema Pathak "CABAL: Training and Placement Departmental Portal" IOSR Journal of Engineering (IOSRJEN), Volume 3, Aug-2019.
- [2] Prof. Anagha Kulkarni, Priyanka Hajare, Priyanka Khandave, ShitalAdhav,SwatiPimpale," Implementation of Online Placement System", IJERMT All Rights Reserved International Journal of Emerging Research in Management & Technology ISSN: 2278 -9359 (Volume-5, Issue-1), January2016.
- [3] Rajnish Tripathi, Raghvendra Singh, Ms. Jaweria Usmani "Campus Recruitment and Placement System" International Conference on Recent Innovations in Science and Engineering, April 2018.
- [4] Swati Choudhary, Monica Landge, Shital Salunke, Swarupata Sutar, Kirti Mhamunkar "Advance Training and placement web portal" International Journal of Technical Research and Application ISSN: 2320- 8163 Volume:4 Issue: 2, 2016.
- [5] P. Anitha Vairamany, Dr. K Subramaniyan "Placement Prediction in Self Employment Using K Means Clustering" (UGC Care Journal, Volume 40, March 2020.
- [6] Shubhangi Patil, Nutan Sonawane, Priyanka Kulkarni, Priyanka Khadse "Training and Placement Portal with Student Performance Analysis" International Research Journal of Engineering and Technology (IRJET), Volume-5, May-2018.
- [7] Dr. Ram Joshi, Mrinal Chaudhari, Pratiksha Gaikwad, Savani Kadam "Training and Placement Portal" International Research Journal of Engineering and Technology (IRJET), Volume-4, Dec-2017.
- [8] Huda Al-Shehri, Amani Al-Qarni, Leena Al-Saati, Arwa Batoaq, Haifa Badukhen, Saleh Alrashed, Jamal Alhiyafi, Sunday O. Olatunji "Student Performance Prediction Using Support Vector Machine and K-Nearest Neighbor" IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), 2017.
- [9] Rashi Bansal, Akansha Mishra "Mining of Educational Data for Analysis Students overall Performance", March 2017.
- [10] Mishra T., Kumar D., Gupta S. "Students' performance and employability prediction through data mining: a survey" Indian J. Science Tech. June 2017.
- [11] Ms.Tismy Devasia ,Ms.Vinushree T., Mr.Vinayak Hegde "Prediction of Students Performance using Educational Data Mining" Amrita Vishwa Vidyapeetham University, Mysuru, Karnataka, India, 2016.
- [12] Dr. Anjali B Raut, Ms. Ankita A Nichat "Students Performance Prediction Using Decision Tree Technique" International Journal of Computational Intelligence Research ISSN 0973-1873 Volume 13, Number 7, 2017.
- [13] K. J Ashika , Jagruthi M. N, Nikhil N, Pooja S "Educational Data Analytics using Association Rule Mining for Student Job Prediction" International Research Journal of Engineering and Technology (IRJET) Volume: 07 Issue: 06, June 2020.
- [14] Mangasuli Sheetal, Prof. Savita Bakare "Prediction of Campus Placement Using Data Mining Algorithms" International Journal of Advanced Research in Computer and Communication Engineering(IJARCCCE), Volume 5, June-2016.
- [15] Animesh Giri, M Vignesh V Bhagavath, Bysani Pruthvi, Naini Dubey "A Placement Prediction System Using K-Nearest Neighbors Classifier" Second International Conference on Cognitive Computing and Information Processing (CCIP), 2016.