Campus Placement Prediction Using Brain.js

Ekansh Khandekar, Vishal Jadhav, Sanyami Naik, Shubham Solanki

Department of Computer Engineering, Zeal College of Engineering and Research, Pune, Maharashtra, India

ABSTRACT

Finding a job is difficult, and even when you do, you cannot be certain that the organisation is the greatest fit for you. Several job recommendation sites provide a plethora of possibilities, but how many of those options are actually useful is a well-known reality among everybody. We developed a Machine Learning system that uses Brain.js, a javascript library for neural networks, to forecast an individual's chances of being hired in a specific firm based on numerous characteristics such as their percentage, specialisation, and work experience. Our research aims to assist students, colleges' Training and Placement Offices (TPO), and companies' Human Resources departments in finding the best fit companies.

I. INTRODUCTION

to Placement of students is one of the extremely important activities in educational organizations Every student dreams to be successful in life. But every student cannot be placed in their dream company, hence this project plans to anticipate the placement opportunity of the students who have a better chance of placement in certain companies and thus find the best fit for the applicant, it will not only preserve time but will also assist in increasing the trust of the students.

We normally browse for jobs and apply for them on LinkedIn or Naukri.com, but these platforms provide us a wide range of options. We've noticed that we get quite a few of these. We are either disqualified or have outrageously priced options. To solve this problem, we intend to develop a technique for forecasting the possibility of being chosen in order to save time and allow a person to focus more on the task at hand. This will help the learner achieve his or her objectives. Recruiters or Human Resources, as well as the firm with the best fit, discover the ideal candidate for the job. As a result, the assignment will be completed to their satisfaction.

II. METHODS AND MATERIAL

A. Neural Network

A Neural network is similar to the human brain in that it has multiple interconnected nodes arranged into layers. Input, hidden, and output layers are the three types of layers. In neural networks, data flows from the input to the output in a specific direction.

This technique enables us to load a large amount of labelled data into a neural network and train it to produce a desired output given some input. Neural networks are considerably superior to the human brain in terms of processing speed and accuracy.

An Artificial Neural Network (ANN) is a data processing paradigm based on how biological nerve systems process data, such as the brain. This paradigm relies heavily on the particular structure of the information processing system. It's made up of a lot of highly interconnected processing units (neurons) that work together to solve problems. ANNs, like humans, learn by doing. An artificial neural network (ANN) is trained for a specific task, such as pattern recognition or data classification. An ANN is tailored for a specific goal, such as pattern recognition or data categorization, through a learning process. Changes in synaptic connections between neurons are involved in learning in biological systems. They can't be programmed to perform a certain task, otherwise else valuable time will be lost, and the network may potentially malfunction. The negative is that the network's operation might be unexpected because it figures out how to solve the problem on its own. Neural networks are currently just a simple grouping of primitive artificial neurons.

This clustering is accomplished by forming layers that are then linked together. The "craft" of building networks to solve real-world problems also includes figuring out how these levels connect.
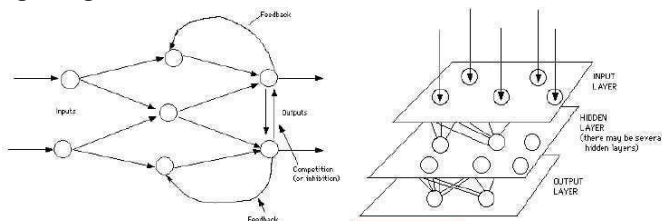

Figure 1: A sample Neural Network Design

As seen in Figure 1, all artificial neural networks have a similar structure or topology. Some of the neurons in that structure interface with the real environment to accept inputs. The network's outputs are sent to the outer world by other neurons. This output could be the character that the network thinks it scanned or the image that it thinks it's seeing. The remaining of the body's neurons are hidden from view.

In comparison to typical computers, neural networks approach problem solving in a unique way. In order to solve a problem, traditional computers employ an algorithmic approach, in which the computer follows a series of instructions. The computer will not be able to solve the problem until the particular procedures that the computer must take are known. As a result, traditional computers' problem-solving capabilities are limited to situations that we already understand and know how to solve. However, computers would be far more beneficial if they could perform tasks that we are unfamiliar with. When it comes to processing information, neural networks are analogous to the human brain. The network is made up of a large number of densely connected processing components (neurons) that operate in tandem to solve an issue. Neural networks learn by observing others.

B. Brain.js

Building a neural network with Brain.js is a phenomenal approach to do it. It learns the patterns and relationships between the inputs and outputs so that it may make an educated guess when dealing with similar problems.

Brain.js is a JavaScript-based GPU-accelerated neural network toolkit for browsers and Node.js. It's simple, quick, and easy to use. It has a variety of neural network implementations since different neural nets can be trained to do different things.

The web-application uses the JavaScript library, Brain.js to train the machine learning model. It trains the students data, collected during the initial registration by the admin, efficiently. Resulting in an outspoken accuracy score of approximate 97 percent.

C. Proposed System

The Campus Placement web application built using various frontend and backend technologies consists of three important domains. The following are the domains which have their own specific role:

a) Student:

The student is the prime user of the Campus Placement Prediction System, online web based

application. The web-application built using Node js will prominently be used for the sole purpose of on campus placements of a particular college. The student will foremost need to register himself on the web portal by entering in some basic information such as Personal information, Admission details, SSC as well as HSC score, current course details along with marks up until the previous semester. He will also be required to state his achievements and skills along with his experience as an intern in a firm. Most of the data collected during registration will in turn be used as input to the machine learning model for the predicative analysis in order to analyze whether the student is placed or not.

After the initial registration the student can directly apply for the companies which are hiring on his campus. He can get access to the top jobs available and can also get direct access to the company's profile. He can have a look at the Department wise statistics of the previous year's placements of that particular college. Collectively, he can update his profile and new skills he has grasped time to time. By accessing the Campus Prediction button on the dashboard, the student will get to know whether he may get placed for a particular company he wishes to apply for, and if not, he will be suggested various skills he will need to acquire so as to get fit for the job. If the student needs help regarding any activity, he can contact the coordinator, who will help him resolve the issue. The students will also be notified on daily basis about the placement activities.

b) Admin:

The Training and Placement Officer of the college is the admin of the web-application. He acts as the ultimate administrator and has control over most of the functions offered by web app. The major work of the admin is to retrain the machine learning model with data of the students for Campus Placement Prediction. He gets to supervise the data being generated from time to time, dump it whenever

required and suggest new skills to the students in order to enhance their placement chances.

The admin has the ultimate right to select the coordinator from among the students. The coordinator in turn acts as the Campus Ambassador for the placements. The companies who wish to recruit freshers from the campus, first and foremost need to approach the admin. After the admin deems the company fit for the on-campus placement, only then the company can hold the recruitment drive. If the college starts offering a new course to the students, the admin himself can add the particular course or department to the web portal.

Thus, the admin is the ultimate bearer of the Campus Placement Prediction System web-application.

c) Company:

Once a certain company, which wishes to recruit the students from the campus, is approved by the admin, it will get its own login details. Thereon, the company can post a job on the portal which will include the job description, number of vacancies, skills and qualification required, salary structure and the recruitment process. The company can manually schedule an interview for the job posted, by approaching the admin.

They will have access to the profile of the students who have applied for the job and can select the particular candidate in their own proximity.

III. RESULTS AND DISCUSSION

After we implement the neural network and train the model we will get the following output, which contains the accuracy score. The Figure 1 also shows the total number of records along with the number of entries correctly predicted and the ones predicted incorrectly.
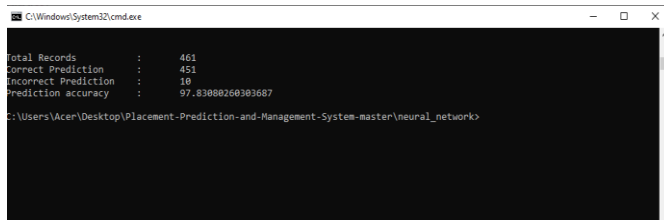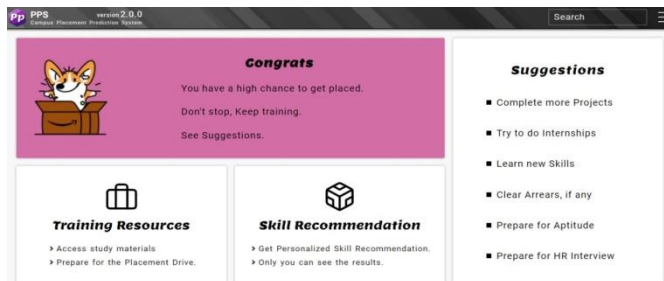
Figure 2: Prediction Analysis

Figure 3: Prediction Result (Placed)

Figure 3 depicts the scenario where the student gets the prediction result, where he has high chances of getting placed for the recruiting companies. He gets a congrats message as the result. The result is based on the various attributes given as input to the model.
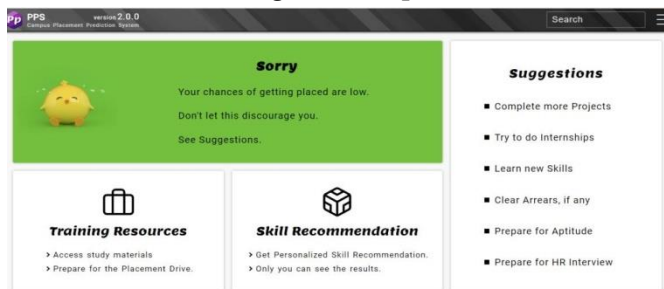

Figure 4: Prediction Result (Unplaced)

Figure 4 depicts the scenario where the student gets the result as sorry and he has low chances of getting placed, which in turn is based on the attributes given as input to the machine learning model. Not just that, the respective student will also get the training resources and suggestions from the admin, along with the recommendation of the skills he should grasp, inorder to crack the placements.

IV. CONCLUSION

After the implementation of the above-stated algorithm, we have got good accuracy of 97.83%

which will help our system to predict the chances of the candidate getting selected in a particular company with greater precision. With the help of a Deep Neural Network and the javascript library for neural networks, brain,js, this accuracy was achieved because of the several hidden layers with the activation functions of logistic sigmoid functions, which helped us get better accuracy than any of the normal Machine Learning algorithms. With the addition of more parameters such as the number of certification courses, prizes won, hours of study, accuracy can be improved and prediction will be more accurate.

V. REFERENCES

[1]. Bharat Udawat , Advait Kale , Divit Sinha , Hardik Sharma, "Placement Prediction System", August 2021, Springer, (AISC,volume 1394). DOI: https://doi.org/10.1007/978- 981-16-3071-2_3.

[2]. Nikhil Kumar, Ajay Shanker Singh, Thirunavukkarasu K., E. Rajesh, "Campus Placement Predictive Analysis using Machine Learning", 2020, IEEE, 2nd ICACCCN, DOI: 10.1109/ICACCCN51052.2020.9362836.

[3]. Tasneem Abed, Ritesh Ajoodha, Ashwini Jadhav, "A Prediction Model to Improve Student Placement at a South African Higher Education Institution", March 2020, IEEE, DOI:https://doi.org/10.1109/SAUPEC/RobMech /PRASA4 8453.2020.9041147.

[4]. Aparna V.Mote, Utakarsha Musmade, "GROSS DOMESTIC PRODUCT (GDP) PREDICTION: A REVIEW", JETIR May 2019, Volume 6, Issue 5, DOI: http://www.jetir.org/papers/JETIRBO06061.pdf

[5]. https://brain.js.org/#/

[6]. https://blog.logrocket.com/an-introduction-to-deep- learning-with-brain-js/