Campus Placement Prediction

Jahnvi Shah[1], Shivangi Kochrekar[2], Neha Kale[3], Sakshi Patil[4], and Anand Godbole[5]

Computer Engineering Department, Sardar Patel Institute of Technology, Mumbai, India
[1]jahnvi.shah@spit.ac.in,[2]shivangi.kochrekar@spit.ac.in,[3]
neha.kale@spit.ac.in,[4]sakshi.patil@spit.ac.in,[5]anand_godbole@spit.ac.in

Abstract. The high rates of unemployment in India can be battled by increasing the employability of people. The 20-24 age group is one of the largest groups of unemployed people, of which college graduates constitute a big portion. Colleges can drastically reduce the number of unemployed graduates by introducing courses and changing the curriculum to help develop the skills that employers look for in graduates. We built a system that helps analyze the difference in the skill sets of placed and not placed students. It predicts whether a student with a given skill set would be able to secure a job or not. It uses not only technical skills but also takes into consideration other soft skills which are essential to land a job. The accuracy obtained is 87% and 90% for the SVM model and XGBoost model, respectively. We found that the technical skills, projects, certified courses are taken, and the internships of the student are the most important parameters. The results are promising and sure to improve placement rates in colleges.

Keywords: Machine Learning, Placement, Prediction, SVM, XGBoost

1 Introduction

As the unemployment rate rose in urban and rural India, over 1.5 million people lost their jobs in August 2021 [1]. According to the Centre for Monitoring Indian Economy (CMIE), the unemployment rate rose from 6.96% in July 2021 to 8.32% in August 2021. Furthermore, the unemployment rate in urban areas is more than two percent greater than in rural areas. Youngsters in the 20-24 age group reported an unemployment rate of 37 percent, whereas the graduates amongst them reported a much alarming and higher unemployment rate of over 60 percent [2]. These high rates of unemployment amongst graduates are primarily attributed to the unemployability of the youth. Employability is a prerequisite for employment. There is a gap between the skills of graduates and the skills needed by industries. The education system mainly focuses on knowledge and written examinations rather than on practical skills [3]. Colleges which help students develop skills required by the industry observe a rise in their placements.

We propose a system that will predict whether a student gets placed or not with his/her current skill set. This system will help with analysis and show how students should be groomed to increase their employability. We collected data from students regarding their skills and placements, cleaned the data and performed data visualization. Then we created SVM and XGBoost models and trained them. We then used the testing dataset to test the model. Next, we performed hyper-parameter tuning. We used C and gamma parameters to fine-tune the SVM model and used Stratified-K-fold Cross-Validation for the XGBoost model. We found that the technical skills possessed by the student, projects done by the student and certified courses taken by the student were most important to enhance the chances of getting hired. We obtained an accuracy of 87% for the SVM Model and an accuracy of 90% for the XGBoost Model. We believe that this model will help students and colleges immensely.

2 Literature Survey

The system proposed by Manvitha and Swaroopa [4] predict the placements of the current students using parameters like backlogs, credits, and overall GPA of previous year students. The accuracy of the random forest algorithm was 86% and was found to be better as compared to the Decision tree. Nagamani et al. [5] applied random forest and support vector machine on the previous and current record of students for placement prediction, obtaining an accuracy of 82.85% and 85.14% for the respective algorithm. Accuracy and execution time can further be improved by the removal of unwanted attributes. In [6] Patel and Tamrakar discussed various clustering algorithms, their experimental results, and execution times. The algorithms used were Simple K means, Farthest First, Filtered, Hierarchical, and Make Density-Based Cluster. The results were tested on placement data of 150 final-year students. A study by Premalatha and Sujatha [7] shows that for placement prediction, the most preferred methods are ANN and Decision tree. However, they observe that the most accurate results are shown by Linear Regression Model and Naïve Bayes Classifier.

A machine learning cluster scheduler named Harmony is used by Bao et al. [8] for maximizing the performance of a model. The reward prediction model is built which uses the historical data for training and unseen data for predicting the outcome of the model. It has been observed that Harmony has outperformed the representative schedulers by 25% based on the evaluation of the Kubernetes cluster. Linsey [9] correctly predicted the employment of 107 students and unemployment of 78 students, with an accuracy of 87%. It was observed that internship was the most important factor for assessing the employability of students followed by co-curricular activities. Shreyas et al. [10] not only predicts the placement of students but also the company where they will be placed based on the data of previous year students. Naive Bayes classifier and KNN algorithm were used. Parameters considered are technical, aptitude skills, GPA, or diploma results. Apart from predicting whether a student gets placed or not, the model by Thangavel et al. [11] also predicts the type of company a student might get into i.e., Dream, Core, or mass recruiter. They used Classification via regression, Logistic regression, Decision Tree, Metabagging classifiers, and Naïve Bayes. The decision tree had the highest match percentage of 84.42. Their system also had a Not Interested category which we considered while picking the data. Chinmay et al. [12] presents the comparison of various data mining algorithms for the prediction of student's placement along with the creation of a website for the same. Parameters such as CGPA and technical skills are taken into account. Even phrases such as group discussion, interviews, and aptitude tests can be implemented using the proposed system. The model proposed by Irene et al. [13] considers parameters like CGPA, number of hackathons, number of certifications,current backlogs and quantitative, logical reasoning, verbal, programming scores. Samtha et al. [14] used the ANN algorithm on a dataset generated by taking an exam on their portal to understand the relative expertise of students in the skills. The model is also tested with their college's placement cell database, which helped us identify an essential data source. Ishizue et al. [15] created a classification model of their own to assess the effect of the explanatory variables like Psychological Scales, Programming Tasks, and Student-answered Questionnaires. The analysis carried out in [16] shows that the gender and residential status of the student weighs a little higher than the marks of the students while predicting their placement. The use of logistic regression along with gradient descent algorithm gave them an accuracy of 83.33%. Muthusenthil et al. [17] present a model for predicting the final CGPA and placement of students which passed out in 2018 and 2019. 20 attributes are taken into consideration for improving the accuracy of the model. K-nearest neighbors algorithm (KNN), Decision Tree, and Linear regression are used, Linear regression has outperformed the other algorithms with an accuracy of 94%.

After conducting extensive research, we found some gaps, gathered data sources and gained significant insight to proceed further. We found that some parameters are more important than others like internships and have used them accordingly. Some unexpected parameters like gender were also revealed through our thorough research. Out of the various algorithms used, the SVM model and the XGBoost model gave the best results. Hence we used these models in our project. We found various data sources and some authors even created their own datasets. Similarly, we created our own dataset based on the data of students of top engineering colleges in India.

3 Proposed Methodology

In this section, we have described our proposed methodology, which predicts the placement of students using parameters that consider a student's holistic abilities. We found that numerous research papers used data that incorporated only academics for predicting the outcome, not the other factors. Our research identified essential factors using which we created two classification models and compared them. The steps to create the model are shown diagrammatically in Fig. 1 and are described as follows:

3.1 Assembling Data

We collected the data of students i.e passed out and currently pursuing degrees in their respective colleges for further research. The parameters taken into consideration were an academic performance that is their SSC, HSC, and current year CGPA along with backlogs, number of projects undertaken, number of internships and total duration of internships, number of research papers, communication skills, technical skills along with extracurricular activities which includes several competitions, case-studies and hackathons won.

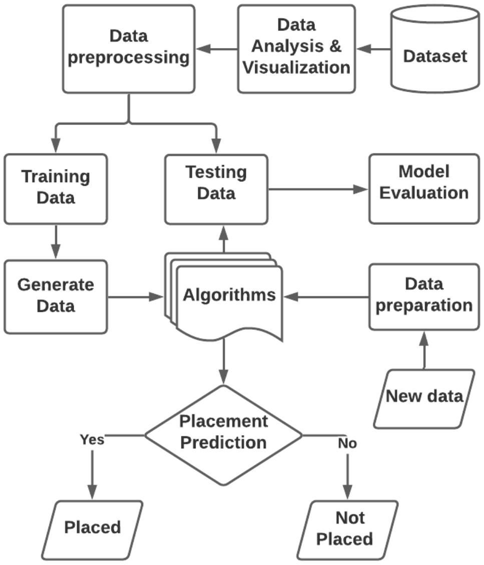

Fig. 1. System Diagram

3.2 Data Analysis and Visualization

To understand the data further and analyze the collected data, we visualized it. The highly correlated parameter and eliminated the parameters with high correlation using heatmap.

3.3 Data Preprocessing, Labelling and Splitting

The data was preprocessed by dropping columns of the parameters that were not relevant to our system. The missing values in the HSC percentage and diploma column were handled by replacing them with 0 and they were combined in a single column. The data were then normalized using a min-max scaler so that the features are on the same scale and compatible. After preprocessing the data, we labeled the categorical data of the target variable was encoded using label encoding. Then we split the data set into training and testing data sets. The splitting ratio was kept at 80-20, that is 80% data was used for training and the remaining 20% was used for testing.

3.4 Data Generation

A widely used technique to generate synthetic data. It is done after the data is split into training and test sets. Generation is done only on the training set otherwise the performance measures could get skewed. The training data was augmented using a GAN generator which automatically discovers patterns and regularities in the input data. A discriminator is used to differentiate between the real or fake that is generated data. The output probabilities of the discriminator help the generator to produce better examples.

3.5 Built and Trained the Models

We created a Support Vector Machine and an XGBoost model. Then we trained these models on the training data set.

SVM Model: Support Vector Machine algorithm is applied to the data. It is a supervised machine learning algorithm that is majorly used for Classification. A decision boundary is created by the algorithm that segregates n-dimensional space into classes so that a new data point appearing in the future can be easily classified in the correct category. The best decision boundary is called a hyperplane. A standard SVM tries to segregate the two different classes such that it does not allow any point to be misclassified. However, in this case, accuracy on the test dataset might be lower because the decision boundary is too sensitive to noise and to small changes in the independent variables. This might result in an overfit model. Conversely, if the decision boundary is placed far off from the classes, the model may be underfitted. So to take care of these points and improve the performance of the model a soft margin SVM is used which allows some points to be misclassified but gives a more generic model. This will work better on new previously unseen examples. To implement a soft margin SVM, some parameters are adjusted before we train the model which are called hyperparameters. They help us find the balance between bias and variance and thus, prevent the model from overfitting or underfitting. Hence we performed Hyperparameter Tuning for SVM to get better results.

The parameters considered in the SVM algorithm are 'c' and 'gamma'. C parameter adds a penalty for each misclassified data point. If c is large, SVM tries to minimize the number of misclassified examples due to high penalty which results in a decision boundary with a smaller margin and conversely is true for a small value of c. Gamma parameter controls the distance of influence of a single training point. Low values of gamma indicate a large similarity radius which results in more points being grouped together. The c and gamma parameters are to be optimized in this case keeping (1) and (2) in check.

$$0.0001 < gamma < 10 \tag{1}$$

$$0.1 < c < 1000 \tag{2}$$

In our model, we used GridSearchCV to select the best parameters using the constraints mentioned in (1) and (2). Choosing the kernel as 'rbf', the 'c' value yielded by the model was 100 and the 'gamma' value yielded was 0.1. The c parameter is close to the upper bound meaning the penalty was kept high for misclassifying the examples whereas the gamma parameter is kept not too high or low.

XGBoost Model: XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. It is used in supervised learning for problems of Regression and Classification. It has a variety of regularizations which helps in reducing overfitting. It also supports auto tree pruning hence the decision tree will not grow further after certain limits internally. Since we had fewer amounts of data in our dataset using normal train test split it was possible that all values of a particular class would be in the train set and the model would become biased towards that value. Hence Stratified-K Fold Cross Validation was used.

4 Implementation & Results

After finding the best parameters for the model and implementing our methodology, we obtained an accuracy of 87% for the SVM Model and an accuracy of 90% for the XGBoost Model.

Apart from the accuracy, we used other evaluation criteria like the precision, recall, and F1 score to check the model's performance. We plotted a graph to display the same. The score of the evaluation parameters: precision, recall, and F1 scores are plotted on the Y-axis and the Target Variable is on the X-axis as shown in Fig. 2. The value '0' on the X-axis refers to the prediction of students who are 'Not Placed' and '1' refers to prediction of students being 'Placed'.

In Fig. 2 the values of precision, recall and f1-score for XGBoost Model are the same for both the classes, since the algorithm classified equal amounts of False Positives and False Negatives. False Positive samples (FP) are samples that were classified positive but should have been classified negative whereas True Positive samples (TP) are samples that were classified positive and are really positive. Analogously, False Negative samples (FN) were classified negative but should be positive.

$$P = \frac{TP}{TP + FP} \tag{3}$$

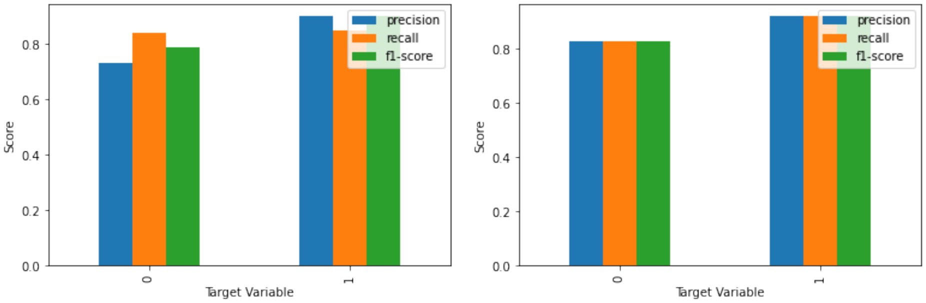

Fig. 2. Left: Results of SVM Model, Right: Results of XGBoost Model

Precision as given in (3) is the classifier's ability to only predict really positive samples as positive. Basically, the less false positives a classifier gives, the higher is its precision.

$$R = \frac{TP}{TP + FN} \tag{4}$$

Recall as given in (4) is the amount of positive test samples which were actually classified as positive. If the false negatives a classifier gives are lesser, its recall becomes higher. Now, if the number of False Positives is equal to the number of False Negatives then (3) and (4) would give the same results.

$$F1 = \frac{2PR}{P + R} \tag{5}$$

F1 Score is the weighted average between recall and precision. The higher the recall and precision are, the higher is the F1 score. Hence, we can see that if P=R in (5), then F1=P=R. For the SVM model in Fig. 2, the recall for the 'Not Placed' class and the precision for the 'Placed' class are high. This means that a high number of actually 'Not Placed' students were classified as 'Not Placed' and the precision of 'Placed' tells that a high number of predicted 'Placed' students were actually 'Placed'.

We also calculated the relative importance of the features. it was found out that Technical Skills, number of Projects, number of certified courses, and number of Internships undertaken by the student are of higher importance compared to other features.

5 Conclusion

After implementing two different algorithms we determined that the SVM model gives an accuracy of 87% and the XGBoost model gives an accuracy of 90%. There are a few limitations to this project. The data we worked on was from students of top engineering colleges. The dataset is small and more parameters can be added. The parameters will vary according to the job profile. We found

that the technical skills, projects, certified courses taken and the internships of the student matter the most for predicting if they will get placed or not.
Our model will help colleges better their poor placement track record. It can be used to improve the curriculum and add relevant courses which help the graduates to secure a job.

References

1. P. K. Nanda, "1.5 mn indians lost jobs in aug as unemployment rate soars," Sept 2021.
2. M. Vyas, "The real unemployment challenge," Jan 2021.
3. P. Deka, "Unemployment in india," Jun 2021.
4. P. Manvitha and N. Swaroopa, "Campus placement prediction using supervised machine learning techniques," *International Journal of Applied Engineering Research*, vol. 14, no. 9, 2019.
5. S. Nagamani, K. M. Reddy, UmaBhargavi, and S. RaviKumar, "Student placement analysis and prediction for improving the education standards by using supervised machine learning algorithms," *Journal of Critical Reviews*, vol. 7, no. 14, pp. 854–864, 2020.
6. T. Patel and A. Tamrakar, "A data mining techniques for campus placement prediction in higher education," *Indian J.Sci.Res.*, vol. 14, 2017.
7. N. Premalatha and D. S. Sujatha, "A comparative study on students placement performance using data mining algorithms," *International Journal Of Scientific Technology Research*, vol. 8, 2019.
8. Y. Bao, Y. Peng, and C. Wu, "Deep learning-based job placement in distributed machine learning clusters," pp. 505–513, 2019.
9. L. S. H., "Predicting employment through machine learning," May 2019.
10. H. Shreyas, P. Aksha, H. S. Suma, A. Suraksha, and M. Tojo, "Student placement prediction using machine learning," *International Research Journal of Engineering and Technology*, vol. 6, no. 4, 2019.
11. S. K. Thangavel, P. D. Bkaratki, and A. Sankar, "Student placement analyzer: A recommendation system using machine learning," pp. 1–5, 2017.
12. C. Chinmay, T. Kunal, S. Siddhant, and R. T. Neha, "Placement prediction by mining student's information," *Proceedings of the International Conference on Recent Advances in Computational Techniques (IC-RACT)*, 2020.
13. T. J. Irene, R. Daibin, A. A. Jeebu, J. Joel, and T. V. Mereen, "Placement prediction using various machine learning models and their efficiency comparison," *International Journal of Innovative Science and Research Technology*, vol. 5, no. 5, 2020.
14. J. Samtha, D. Manjusha, B. Pooja, and A. Usha, "Student placement chance prediction," *Journal of Emerging Technologies and Innovative Research*, vol. 7, 2020.
15. R. Ishizue and H. Sakamoto, K.and Washizaki, "Student placement and skill ranking predictors for programming classes using class attitude, psychological scales, and code metrics.," *Research and Practice in Technology Enhanced Learning volume*, vol. 7, 2018.
16. M. Shahil, "Campus placement analysis and prediction," May 2020.
17. D. Muthusenthil, V. Mugesh S, D. Thansh, and R. Subaash, "Predictive analysis tool for predicting student performance and placement performance using ml algorithms," *International Journal Of Advance Research And Innovative Ideas In Education*, vol. 6, 2020.