INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 643-646

e-ISSN : 2583-1062

Impact Factor: 5.725

CAMPUS PLACEMENTS PREDICTION & ANALYSIS USING MACHINE LEARNING

Dr. M. V. Vijaya Saradhi[1], Thodeti Sowmya[2], Shaik Sufyaan[3], Gumpula Kavya[4], Ganore Pavan[5]

[1]Head of Department and professor, Computer science and Engineering, ACE Engineering College, India.

[2,3,4,5]Computer Science and Engineering, ACE Engineering College, India.

ABSTRACT

Placement of students is one of the most important objective of an educational institution. Reputation and yearly admissions of an institution invariably depend on the placements it provides it students with. That is why all the institutions, arduously, strive to strengthen their placement department so as to improve their institution on a whole. Any assistance in this particular area will have a positive impact on an institution's ability to place its students. This will always be helpful to both the students, as well as the institution. In this study, the objective is to analyze 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. Data pertaining to the study were collected form the same institution for which the placement prediction is done and also suitable data pre-processing methods were applied. 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: Analysis, investigation, research

1. INTRODUCTION

In recent years, the landscape of campus placements has undergone significant transformation, driven by advancements in technology and evolving industry demands. With the advent of machine learning (ML) and data analytics, the traditional approach to campus placements has evolved into a more data-driven and predictive process. This paradigm shift has empowered educational institutions and recruiters to make informed decisions, optimize resources, and enhance the overall placement experience for both students and employers.

Campus placements prediction and analysis using machine learning entail leveraging historical placement data, student profiles, academic performance metrics, and other relevant factors to develop predictive models. These models are designed to forecast various outcomes, such as the likelihood of a student getting placed, the salary range they might command, the sectors they are likely to be hired in, and other valuable insights.

Data Collection and Preprocessing: The foundation of any predictive analysis is robust data collection. In the context of campus placements, this involves gathering data on students' academic performance, skills, internships, extracurricular activities, and demographic information. Additionally, data related to previous placement records, recruiter preferences, and industry trends are also crucial. Preprocessing techniques are then applied to clean, transform, and normalize the data for further analysis.

Feature Selection and Engineering: Once the data is collected, relevant features are selected or engineered to extract meaningful insights. This step involves identifying the most influential factors that contribute to placement outcomes. Features may include academic scores, performance in technical assessments, communication skills, work experience, and more. Feature engineering techniques such as one-hot encoding, scaling, and dimensionality reduction are applied to prepare the data for model training. Model Development: Various machine learning algorithms such as logistic regression, decision trees, random forests, support vector machines, and neural networks are employed to build predictive models. These models are trained on historical placement data, with the objective of learning patterns and relationships between input features and placement outcomes. Ensemble methods and advanced techniques like gradient boosting and deep learning may be utilized to enhance model performance.Model Evaluation and Validation: The performance of the predictive models is evaluated using metrics such as accuracy, precision, recall, F1 score, and area under the ROC curve (AUC-ROC). Cross-validation techniques are employed to assess model generalization and mitigate overfitting. Furthermore, the models are validated using holdout datasets or through real-time testing on upcoming placement cycles. Deployment and Integration: Once the predictive models demonstrate satisfactory performance, they are deployed into production environments. This involves integrating the models into existing placement portals or developing standalone applications for stakeholders' use. The deployed models continuously monitor and analyze incoming data to provide real-time predictions and recommendations.

2. METHODOLOGY

Data Collection: Gather relevant data pertaining to past campus placements. This data may include student profiles (such as academic performance, skills, and extracurricular activities), company profiles (recruitment history, job roles offered), and placement outcomes (whether students were placed or not).

Data Cleaning: Handle missing values, outliers, and inconsistencies in the dataset.

Feature Engineering: Create new features or transform existing ones to improve the model's performance. This could involve converting categorical variables into numerical representations (one-hot encoding), scaling features, or extracting meaningful information from raw data.

Feature Selection: Identify the most relevant features that contribute to the prediction task. Techniques like correlation analysis, feature importance scores, or domain knowledge can aid in feature selection.

Splitting Data: Divide the dataset into training and testing sets. Typically, a large portion of the data (e.g., 70-80%) is used for training, while the remainder is reserved for testing the model's performance.

Model Selection: Choose appropriate machine learning algorithms suited for the prediction task. Common algorithms for classification tasks like campus placement prediction include Decision Trees, Random Forest, Support Vector Machines (SVM), Logistic Regression, and Gradient Boosting Machines.

Experiment with multiple algorithms to compare their performance and select the best-performing one(s).

Model Training: Train the selected machine learning model(s) on the training dataset. During training, the model learns patterns and relationships between input features and placement outcomes.

Model Evaluation: Evaluate the trained model(s) using the testing dataset to assess their predictive performance. Common evaluation metrics for classification tasks include accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC).

Hyperparameter Tuning: Fine-tune the model's hyperparameters to optimize its performance. Techniques like grid search or random search can be employed to search through a range of hyperparameter values.

Cross-Validation: Perform cross-validation to ensure the model's generalization ability and robustness. Techniques like k-fold cross-validation split the data into multiple folds, training the model on different subsets and evaluating its performance on unseen data.

Model Interpretation: Interpret the model's predictions to gain insights into the factors influencing campus placements. Techniques like feature importance analysis or SHAP (SHapley Additive exPlanations) values can help understand the model's decision-making process.

Deployment: Once satisfied with the model's performance, deploy it for real-world use. This may involve integrating the model into an application or system where it can make predictions on new data.

Monitoring and Maintenance: Continuously monitor the model's performance in production and update it as needed to adapt to changes in data distributions or business requirements.

3. MODELING AND ANALYSIS

Algorithms used :

1. Logistic Regression

2. XGBoost

3.navie bayes classifier

1. Logistic Regression (Binary classification)

Logistic regression is a common statistical method used in predicting binary outcomes, such as whether or not a person will be placed in a job after completing a training program. To use logistic regression for placement prediction, we typically started by collecting data on individuals who have completed the training program, including information such as their education level, work experience, and performance in the training program. To make predictions using the logistic regression model,we simply input the values of the predictors for a new individual and the model would output the estimated probability of that individual being placed in a job. If the probability is above a certain threshold (e.g., 50%), the system would predict that the individual will be placed in a job; otherwise, not

2. XGBoost (eXtreme Gradient Boosting)

XGBoost is a popular machine learning algorithm that can be used for multiclass classification problems, including predicting the package range of students in campus placement based on their performance and other parameters. Here's a high-level overview of problem solving approach using XGBoost:

1) Data Preparation: Collected data on students' academic performance, skills, work experience, and other relevant parameters. Preprocessing the data by handling missing values, handling outlier, encoding categorical variables, and scaling numerical variables.
2) Feature Selection: Identify the most important features that are highly correlated with the target variable (i.e., package range). You can use techniques such as correlation analysis, mutual information, or feature importance ranking provided by XGBoost.
3) Train-Test Split: Split the data into training and testing sets. Typically,80-20 split is used.
4) XGBoost Model Training: Train an XGBoost classifier on the training data. XGBoost is a gradient boosting algorithm that builds an ensemble of decision trees iteratively to minimize the loss function (e.g., softmax for multiclass classification).
5) Hyperparameter Tuning: Optimize the hyperparameters of the XGBoost model using techniques such as grid search or random search. Important hyperparameters include the learning rate, number of trees, maximum depth, and regularization parameters.
6) Model Evaluation: Evaluate the performance of the XGBoost model on the testing data using metrics such as accuracy, precision, recall, F1-score, and confusion matrix. You can also visualize the feature importance and decision boundaries of the model.
7) Deployment: Model is packaged into joblib file and further used in flask API. Here , Comparison of the 3 Algorithms for multiclass classification is done through which the best one is selected.

1. LogisticRegression
2. Random Forest
3. XGBoost

4. NAIVE BAYES CLASSIFIER

Data Preparation: Gather data related to campus placements, including attributes such as academic performance (GPA, scores in specific subjects), skills (programming languages, communication skills), extracurricular activities, internships, and any other relevant information. This data should include both features (attributes) and the target variable (placement outcome, such as 'placed' or 'not placed').

Data Preprocessing: Preprocess the data to handle missing values, encode categorical variables, and scale numerical features if necessary. This step ensures that the data is in a suitable format for training the Naive Bayes classifier.

Training the Naive Bayes Classifier: Split the data into training and testing sets. Use the training set to train the Naive Bayes classifier. Naive Bayes assumes that the features are conditionally independent given the class label, which simplifies the probability calculations. Train the classifier using the training data, where it learns the probability distributions of features given the class labels (placed or not placed).

Model Evaluation: After training the classifier, evaluate its performance on the testing set using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score. These metrics help assess how well the classifier predicts the placement outcomes based on the given features.

Interpretation and Feature Importance: Naive Bayes classifier provides probabilities for each class label given the input features. Analyze these probabilities to understand the importance of different features in predicting placement outcomes. Features with higher probabilities for a particular class label are more influential in determining that label.

Model Optimization (Optional): Depending on the performance of the classifier, you may explore model optimization techniques such as hyperparameter tuning or feature selection to improve its predictive accuracy.

Deployment and Monitoring: Once the Naive Bayes classifier is trained and evaluated satisfactorily, it can be deployed for making predictions on new data. Monitor the model's performance over time and update it as necessary to maintain its accuracy and relevance.

5. RESULTS AND DISCUSSION

Table 1. Comparison of Algorithms

Algorithm	Precision	Recall	f1-score	Accuracy
Logistic Regression	0.522	0.522	0.522	0.522
Random Forest	0.8190	0.819	0.819	0.819
XGBoost	0.831	0.831	0.831	0.831

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 643-646

e-ISSN : 2583-1062

Impact Factor: 5.725

Navie bayes classifier	0.912	0.912	0.912	0.912

6. CONCLUSION

In this way, logistic regression from a machine learning method in predicting the campus placement of students based on his/her performance in academics and other skills and exams. The model is trained using the previous year's data like students marks in all semesters, personal interests, internships, other activities, etc.

The limitations of the Project is that the student should not do any malpractice while appearing for the general aptitude exams which we have asked them to solve, in order to test their knowledge and he/she should fill in their correct academic details .

This seminar report is undertaken to explain machine learning prediction approaches and improve the efficiency of Machine Learning Model to predict the campus placement of students.

7. REFERENCES

[1] Irene Treesa Jose, Daibin Raju, Jeebu Abraham Aniyankunju, Joel James, Mereen Thomas Vadakkel "Placement Prediction using Various Machine Learning Models and their Efficiency Comparison" Volume 5,P 1007,1008 Issue 5, May – 2020

[2] D. Satish Kumar, Zailan Bin Siri, D.S. Rao, S. Anusha "Predicting Student's Campus Placement Probability using Binary Logistic Regression" Volume-8 Issue-9, July, 2019

[3] Manoj K Shukla, Pranay Rambade, Jay Torasakar, Rakesh Prabhu, Prof. Deepali Maste "Students Placement Prediction Model Using Logistic Regression" Volume 5, Issue 01

[4] Ajay Shiv Sharma, Swaraj Prince, Shubham Kapoor, Keshav Kumar "PPS - Placement Prediction System using Logistic Regression" P 338,339 2014 IEEE International Conference on MOOC, Innovation and Technology in Education (MITE)

[5] Animesh Giri, M Vignesh V Bhagavath, Bysani Pruthvi, Naini Dubey "A Placement Prediction System Using K-Nearest Neighbors Classifier" 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP)

[6] Shawni Dutta1 and Samir Kumar Bandyopadhyay "Forecasting of Campus Placement for Students Using Ensemble Voting Classifier" Asian Journal of Research in Computer Science 5(4): 1-12, 2020; Article no.AJRCOS.57125 ISSN: 2581-8260

[7] Vijay N. Kalbandea , Dr. Chandrahas. C. Handa "Predicting the Performance of Engineering Students in Campus Placement for It Sector by Using ANN" Volume III, Issue II, February 2016 IJRSI ISSN 2321 – 2705.