FutureFit: ML-Based Predictive Modeling for Campus Placement Success

Name: Habibun Nabi Hemel
Department of Computer Science
BRAC University, Dhaka, Bangladesh
Email: firstname.middlename.lastname@g.bracu.ac.bd

Name: Ishrak Hamim Mahi
Department of Computer Science
BRAC University, Dhaka, Bangladesh
Email: firstname.middlename.lastname@g.bracu.ac.bd

***Abstract*—FutureFit the placement predictor utilizes machine learning to forecast job opportunities and potential salaries for BSc students, empowering them to make informed decisions about their future careers. It employs various algorithms such as Decision Tree, Random Forest, SVC, Logistic Regression, Naive Bayes, and KNN to analyze a dataset consisting of student data and predict placement outcomes. Through meticulous data preprocessing, model training, and evaluation, the project aims to guide students towards achieving successful career trajectories. The results and comparison analysis demonstrate the efficacy of the models, with Random Forest emerging as the most accurate predictor among the evaluated algorithms.To make it user friendly FutureFit is live on internet where anyone can see their outcomes by putting their inputs. Overall, FutureFit serves as a valuable tool for students, offering insights into future job prospects and motivating them to strive for excellence.**

***Index Terms*—Machine learning, Data Analysis, Ter classification, K- nearest neighbors (KNN),Naive Bayes, SVM, regression, Random Forest and Decision tree, LSTM.**

I. Introduction

FutureFit,employs machine learning to forecast job opportunities and potential salaries for BSc students, serving as a roadmap for students to monitor their career aspirations and visualize their future prospects. If a student finds the projected job package unsatisfactory, they can use it as motivation to enhance their efforts. By leveraging insights from past graduates' data, our model illuminates how specific actions can impact future earnings. We utilize previous alumni datasets to train machine learning models, enabling us to provide personalized predictions tailored to individual students. As third-year students ourselves, we understand the uncertainties surrounding career paths and sought to create a tool that offers a reality check on our progress, fostering a drive for continuous improvement. Thus, FutureFit acts as a guiding beacon for students, empowering them to make informed decisions and strive towards a successful career journey.

A. Architecture Diagram

In our workflow, we start by carefully curating and preparing our dataset, a foundational step that sets the tone for accurate predictions. Through meticulous data preprocessing, we clean, transform, and organize the information, ensuring its quality and relevance for our analysis. Subsequently, we split the dataset into two distinct parts: a training dataset comprising 70 of the data and a test dataset containing the remaining 30.

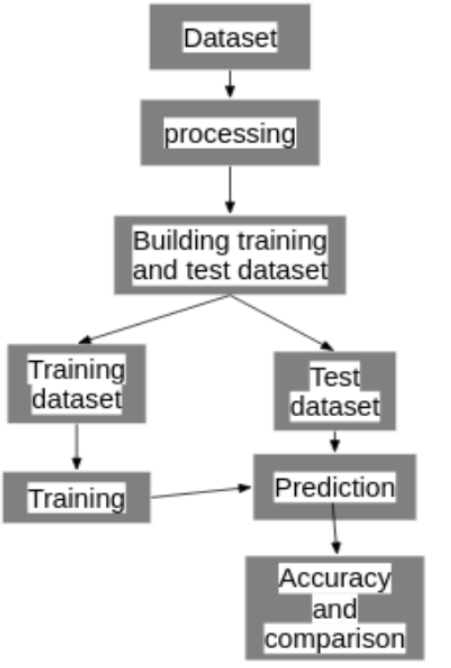

Fig. 1. Work Flow

This division facilitates robust model training on a diverse range of examples while allowing us to assess the model's performance on unseen data. With the datasets prepared, we embark on training various models, employing algorithms tailored to our predictive task. Once the models are trained, we utilize them to make predictions on new data, evaluating their accuracy and effectiveness through comprehensive analysis. This iterative process of dataset handling, model training, prediction, and accuracy assessment forms the backbone of our workflow, ensuring the reliability and efficiency of our predictive modeling efforts.

B. Sample Dataset

We used a synthetic dataset because we couldn't find a large enough real-life dataset to train our models. This made our data distribution balanced, with about half "yes" and half "no" values. Real datasets we found were very small, with only around 215 rows and 13 to 14 features, which wasn't ideal for

Name of Student	Roll No.	No. of DSA	CGPA	Knows ML	Knows DSA	Knows Python	Knows JavaScript	Knows HTML	Knows CSS	Knows Cricket	Knows Dance	Participated i	Was in Coding Club	No. of backlogs	Interview Room Tempe	Age of Candidate	Branch of Engineering	Placement Package
Todd Pope	30678	151	8.52	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes	2	24.2	24	Computer Science	20.01
Sandra Brown	49191	24	1.23	Yes	No	No	Yes	No	No	Yes	No	No	Yes	1	20.5	18	Computer Science	10.97
Mrs. Amanda Singleton	83519	333	9.85	No	Yes	Yes	No	Yes	No	No	No	No	No	1	21.6	25	Mechanical Engineering	7.51
Matthew Alvarado	56203	132	1.96	No	No	Yes	No	Yes	No	No	No	Yes	No	4	21.2	20	Computer Science	4.96
Christine Smith	82173	198	9.73	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	1	20.7	21	Electrical Engineering	46.3
Dustin Hernandez	62701	32	0.56	No	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	0	20.7	23	Mechanical Engineering	4.54
Ruben Thompson	75231	130	6.01	Yes	No	No	No	No	Yes	Yes	No	Yes	No	2	18.1	24	Mechanical Engineering	36.11
Jennifer Kane	40599	209	4.04	No	No	No	No	Yes	Yes	Yes	Yes	Yes	No	3	24.9	23	Mechanical Engineering	3.86
Elizabeth Watkins	93213	100	1.7	No	Yes	No	No	No	No	Yes	Yes	No	No	5	22.9	24	Civil Engineering	6.61
Bryce Price	73969	369	1.8	No	No	Yes	No	Yes	No	No	Yes	No	No	3	21.3	24	Mechanical Engineering	2.78
Hector Wagner	11902	205	4.35	No	No	No	No	Yes	No	No	Yes	No	Yes	0	18.7	19	Electrical Engineering	3.17
Joyce Fisher	66347	179	5.52	Yes	Yes	No	No	No	No	Yes	No	Yes	No	2	26.4	18	Mechanical Engineering	24.72
Caroline Burgess	35096	30	6.43	Yes	Yes	Yes	No	No	No	No	Yes	No	Yes	0	21.7	21	Civil Engineering	23.32
John Nelson	96296	219	3.16	No	No	Yes	Yes	Yes	No	Yes	No	No	Yes	2	20.9	20	Computer Science	3.59
Anthony Swanson	72502	292	2.48	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	1	29.3	19	Computer Science	10.67
Patrick Hayes	36168	121	5.69	Yes	Yes	Yes	No	No	No	Yes	Yes	No	Yes	4	27.5	23	Electrical Engineering	39.62
Kathleen Downs	23207	387	5.54	No	Yes	Yes	No	Yes	Yes	No	Yes	No	No	2	21.4	24	Civil Engineering	45.13
Kim Stein	58002	128	1.16	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	No	0	23.8	20	Mechanical Engineering	11.33
Anthony Adams	38778	204	0.53	No	No	Yes	No	No	Yes	Yes	Yes	Yes	Yes	1	20.6	18	Civil Engineering	3.63
Luis Thomas	63706	314	5.45	Yes	No	Yes	No	Yes	Yes	Yes	No	No	Yes	1	18.2	23	Electrical Engineering	31.24
Jonathan Russell	60257	203	7.56	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	2	20.5	20	Electrical Engineering	37.4
James Ortiz	19318	479	1.46	No	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes	0	29.1	24	Computer Science	3.25
Rebecca House	57328	255	5.63	No	Yes	Yes	No	Yes	No	No	No	Yes	Yes	2	25.3	18	Mechanical Engineering	34.89
Thomas Jones	62362	239	2.03	No	Yes	Yes	No	Yes	No	Yes	No	Yes	Yes	2	26.8	22	Electrical Engineering	6.7
Dr. Anthony Pierce Jr.	57480	207	8.12	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes	2	27.6	25	Electrical Engineering	52.87
Mark Williams	75817	262	9.23	No	Yes	No	No	No	No	Yes	Yes	No	Yes	3	20.7	25	Electrical Engineering	29.05
Douglas Martin	95864	448	8.87	Yes	No	No	Yes	Yes	Yes	Yes	No	No	Yes	4	21.9	19	Computer Science	39.52
Amy Coleman	69397	155	5.23	No	No	No	Yes	Yes	No	No	No	No	No	2	18.3	24	Electrical Engineering	14.16
Catherine Adams	34690	255	3.69	Yes	No	Yes	Yes	No	No	No	Yes	No	Yes	5	19.9	25	Mechanical Engineering	9.14
Joseph Vargas	90607	400	3.35	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	4	23.7	21	Civil Engineering	3.84
Mr. Keith Weber DDS	88403	1	1.42	No	Yes	No	Yes	No	Yes	Yes	No	No	No	4	27.5	24	Civil Engineering	5.27
Wanda Myers	57928	132	9.03	No	Yes	Yes	No	Yes	Yes	Yes	No	No	No	2	23.3	24	Computer Science	48.66
Katherine Wolf	32441	488	9.47	No	Yes	No	Yes	Yes	No	No	No	No	Yes	4	18.6	23	Mechanical Engineering	35.51
Margaret Ramos	23414	221	7.32	Yes	No	Yes	Yes	No	Yes	No	Yes	No	Yes	5	20.8	24	Computer Science	25.38
Jessica Huang	45265	164	3.94	No	No	Yes	No	Yes	Yes	Yes	No	No	Yes	5	29.7	24	Civil Engineering	2.58
Carrie Thompson	62057	111	0.0		No	No	Yes	No	No	Yes	No	Yes	No	0	18.8	20	Computer Science	4.02
Dustin Myers MD	98002	176	4.77	Yes	Yes	Yes	No	No	Yes	Yes	No	Yes	Yes	1	19.4	19	Computer Science	11.24
Brianna Murray	86898	381	6.79	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	5	18.6	25	Mechanical Engineering	50.27
Megan Oneill	84255	77	6.96	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No	4	29.7	22	Electrical Engineering	22.29

Fig. 2. Dataset used for Prediction and Analysis

research. However, we were able to find a larger dataset with 220,000 rows and 19 columns, which gave us plenty of data to work with. This dataset includes various features like Roll No., No. of DSA questions, CGPA, and information about the candidate's skills and activities. This larger dataset provided a better opportunity for data processing and analysis, so we decided to use it for our research.

II. LITERATURE REVIEW

This study [1] develops a placement predictor using machine learning algorithms to forecast student placement in companies. Four algorithms - KNN, SVM, Logistic Regression, and Random Forest - are evaluated for accuracy using parameters like scores and CGPA. SVM and Logistic Regression show high accuracy.

The paper [2] explores the use of machine learning algorithms to predict student placement in companies based on academic data. By employing classification algorithms like Support Vector Machine and Random Forest, the study aims to enhance placement opportunities for students and aid educational institutions in strengthening their placement cells. Results show Random Forest outperforming other algorithms with an accuracy of 85.14, suggesting its efficacy in predicting placement outcomes. Further enhancements could focus on optimizing parameters for specific company requirements and reducing processing time.

In the paper [3]Naïve Bayes, Logistic Regression, support vector machine (SVM). Random Forest and Decision tree performed well, but Logistic Regression outperformed others with 93 accuracy. The work proposes to build an efficient model to predict personnel or applicant employment status among graduate students of the tertiary institution with five classifiers such as logistic Regression, I Bayes, Decision Tree and Support Vector Machine and Random Forest.

The authors of the paper [4] introduce MAYA framework to address student employment status diversity. Components include embedding academic performance, using GAN for class imbalance, LSTM for sequential info, and bias-based regularization for job market biases. The study analyzes biases across majors and predicts early job landing challenges.

The study [5] employs Decision Tree, Naïve Bayes, and Random Forest algorithms to forecast student placement based on MBA student data from Jain University, Bangalore (2020). Random Forest outperforms with an accuracy of 86, showing promise for improving placement classification methods.

This study [6] presents a comprehensive approach to predicting student placements using machine learning models such as KNN, SVM, RF, and Logistic Regression. By analyzing various factors including academic performance and demographic information, the models demonstrate high accuracy in forecasting placement outcomes, with SVM showing the highest precision followed closely by Logistic Regression. These findings underscore the potential of machine learning techniques in enhancing placement prediction systems for educational institutions.

The literature review summarizes studies on predicting student placements using machine learning algorithms. Various approaches, including KNN, SVM, Logistic Regression, Ran-

dom Forest, and Naïve Bayes, are evaluated for accuracy in forecasting placement outcomes based on academic and demographic data. Studies highlight the efficacy of Random Forest and SVM in achieving high accuracy, suggesting their potential for improving placement prediction systems in educational institutions. The MAYA framework addresses student employment diversity by embedding academic performance, overcoming class imbalances, capturing sequential information, and identifying job market biases, offering insights into early job landing challenges.

III. Methodology

The research objective is to develop a machine learning model capable of accurately classifying job placements in college as "Great," "Decent," or "Poor," aiding in assessing the effectiveness of career services and improving overall employment outcomes.Basically our Dataset contain Continous data which can give us regression types result but our goal is to train classification model so that reason we classify the dataset in three parts such as, below 10 LPA = Poor, 10 LPA to 20 LPA = Decent, more than 20 LPA = Great.

A. Data Collection

We used a synthetic dataset because we couldn't find a large enough real-life dataset to train our models. This made our data distribution balanced, with about half "yes" and half "no" values. Real datasets we found were very small, with only around 215 rows and 13 to 14 features, which wasn't ideal for research. However, we were able to find a larger dataset with 220,000 rows and 19 columns, which gave us plenty of data to work with. This dataset includes various features like Roll No., No. of DSA questions, CGPA, and information about the candidate's skills and activities. This larger dataset provided a better opportunity for data processing and analysis, so we decided to use it for our research.

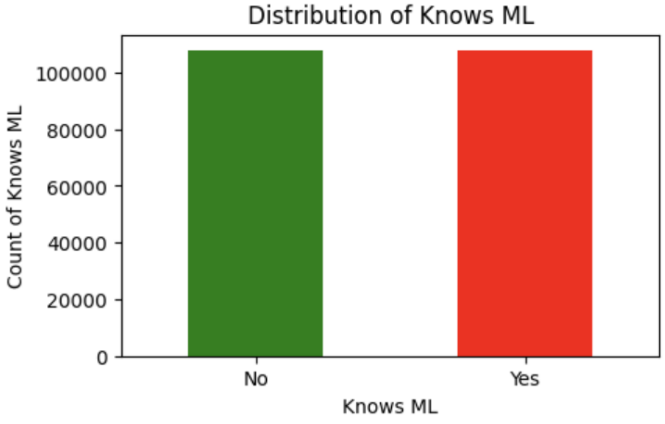

Fig. 3. Distribution of Data

B. Imblanced Dataset

Our goal is to classify job placements into three categories: "Great," "Decent," or "Poor," aiming to assess the effectiveness of career services and improve overall employment results. However, we notice an imbalance in the number of instances across these classes. Specifically, the "Great" category has a significantly larger number of instances compared to the other two categories, while the "Decent" category has fewer than 50,000 instances.

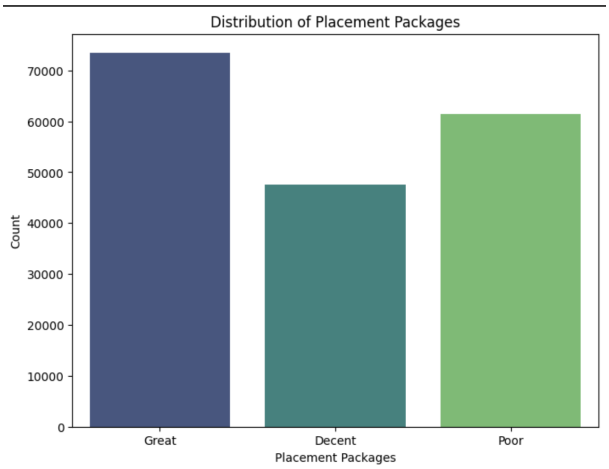

Fig. 4. Classes of Output

C. data cleaning and handling outliers

In the data processing stage, we get rid of unimportant details like student names, roll numbers, and irrelevant factors such as cricket or dance knowledge. Then, we delete any duplicate entries and remove any blank spots because they won't help in training our model. After doing this, we're left with 203426 rows of data and 13 useful features. Once we also remove the blank spots, we end up with a dataset that has 182522 rows and 13 columns.

D. Correlation Matrix

The correlation matrix unveils associations among variables, assisting in feature selection and identifying multicollinearity, while heatmaps offer straightforward data visualizations. The flatness of our heatmap results from symmetrical values across the matrix.It also helps us to decide the most usefull features to train our models.Like the most blue part shows that CGPA has the most empact to the job placement package.

E. Label encoding

To convert categorical features into numerical ones for training our ML model, we've chosen label encoding as it suits our dataset. However, alternatives like one-hot encoding (creating binary columns), ordinal encoding (assigning unique integers based on order), and target encoding (replacing categories with mean target values) are also available.

F. Model Training

We chose decision tree, random forest, support vector machine (SVM), logistic regression, naive Bayes, and k-nearest neighbors (KNN) algorithms for our analysis. We imported

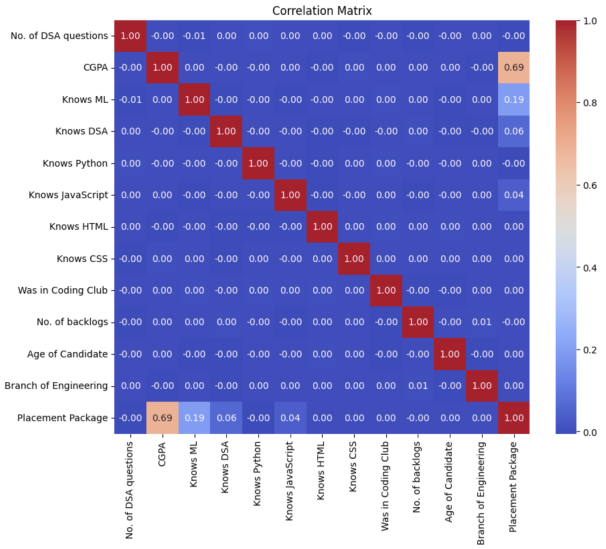

Fig. 5. Correlation Matrix

these models using the scikit-learn library. Our dataset was split into an 70-30 ratio.

a) Decision Tree: Decision Tree is a supervised machine learning algorithm utilized for both classification and regression tasks, with predominant usage in classification scenarios. Each data point in the n-dimensional space represents a data item, where each feature corresponds to a particular coordinate, with 'n' being the number of features. Classification occurs by identifying the hyperplane that effectively separates the classes.

Advantages:

- Effective when data exhibits clear separation patterns.
- Suitable for high-dimensional datasets. additional assumptions are not required
- Performs well even when the number of features exceeds the number of samples.

Disadvantages:

- Performance declines with large datasets due to increased training time.
- Susceptible to performance degradation in the presence of noisy data.
- Doesn't offer direct probability estimates; instead, it requires computationally intensive methods like cross-validation to estimate probabilities.

b) Random Forest: Among the classification algorithms available, Random Forest stands out for its effectiveness. It comprises multiple decision trees, each determining class labels based on features. Through voting, the final class label is determined, resulting in a more accurate prediction model.

Advantages:

- It can balance errors in data sets where classes are imbalanced
- Large data sets with higher dimensionality can be handled additional assumptions are not required
- It can handle thousands of input variables and could identify the most significant variables and as such, it is a good dimensionality reduction method

Disadvantages:

- It does more good of a job for classification problems rather than regression problems as it finds it harder to produce continuous values rather than discrete ones

c) SVM: SVM, or Support Vector Machine, is a supervised learning algorithm primarily used for classification tasks. It operates by finding a hyperplane in the n-dimensional feature space that effectively separates the different classes.

Advantages:

- This algorithm performs best when there is a clear margin of separation
- Effective in high dimensional spaces additional assumptions are not required
- If the number of dimensions is greater than the number of samples, the algorithm would be able to perform better

Disadvantages:

- Performance is affected when large data sets are used as the required training time is more.
- Performance is also affected when the data set has too much noise
- SVM doesn't directly provide probability estimates, rather a computationally intensive five-fold cross- validation is required

d) Logistic Regression: Logistic regression is a classification method ideal for binary outcomes. It establishes a linear decision boundary based on probability interpretation. Parameters are estimated by maximizing an expression using nonlinear optimization solvers. The goal is to predict the class of new data points, utilizing both quantitative and qualitative input features.

Advantages:

- Logistic Regression is good for linearly separable dataset
- It is efficient to train and easy to interpret and implement.
- It not only gives a measure of how relevant a predictor is, but also its direction of association.

Disadvantages:

- It is useful only for predicting discrete functions.
- It should not be used If the No. of observations in the dataset are lesser than the number of features.
- Assumption of linearity between the independent and dependent variables.

e) Naives bayes: Naive Bayes is a supervised machine learning algorithm predominantly used for classification tasks, though it can handle regression problems as well. Each data point in the n-dimensional space represents a data item, where each feature corresponds to a particular coordinate, with 'n' being the number of features. Classification is achieved by estimating the probability of each class given the input features, using Bayes' theorem.

Advantages:

- Efficient and simple algorithm that's easy to implement and understand.
- Performs well even with a small amount of training data.
- Can handle high-dimensional data well.

Disadvantages:

- May suffer from the "zero-frequency" problem if a category in the test data was not observed in the training data, resulting in a probability estimate of zero.
- Relatively simplistic model, which may not capture complex relationships in the data.

f) KNN: KNN, or k-nearest neighbors, is a straightforward algorithm used for classification and regression tasks. It relies on the assumption that similar data points are close to each other in the feature space. Distance, typically measured using Euclidean Distance, determines similarity between points. KNN operates by assigning labels based on the majority vote of its nearest neighbors.

Advantages:

- Building a model, tuning several parameters or making additional assumptions are not required
- This is a versatile algorithm, being able to be used in regression, classification and even search problems.

Disadvantages:

- The algorithm becomes significantly slower as the number of examples and/or predictors/independent variables increases.

G. Deployment in Cloud

We deployed our model on Streamlit at future-fit.streamlit.app to aid users in accessing its benefits. User feedback drove improvements in our app's design. Currently, thresholds for classification are set at 10LPY and 20LPY, with plans to enable user-defined thresholds in the next update. Addressing feedback, we aim to mitigate bias in the Random Forest Classifier and refine data coherence to minimize bias and overfitting in future iterations.

IV. Result and Analysis

The final result of performing various machine learning algorithms are mentioned in the table below.where we can see that random forest is the clear winner

Ml Model	Accuracy
Decision Tree(%)	84.70
Random Forest(%)	85.17
SVC(%)	81.03
Logisctic Regression(%)	79.16
Naives Bayes(%)	78.40
KNN(%)	71.92

TABLE I
Accuracy

We considered decision tree, Random Forest, SVM, Logistic Regression, Naive Bayes, KNN analysis.We end up with find the Random forrest performan the best in different performance measurment matrix. We considered F1, Precision,recall

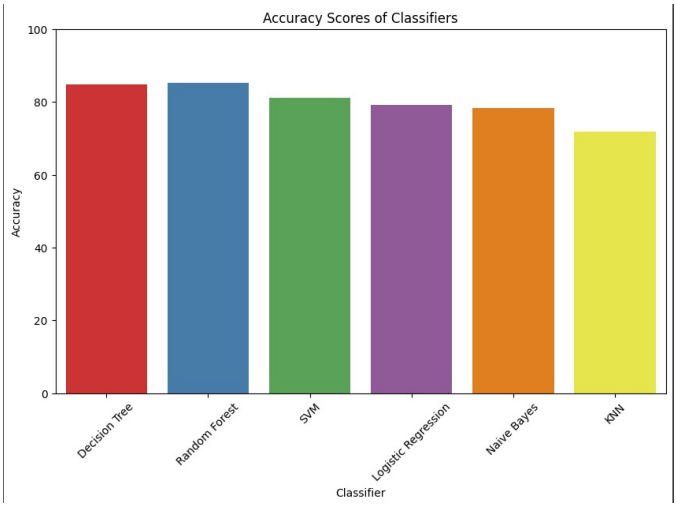

Fig. 6. Accuracy in Bars

,accuracy. Combinig all the four we find that random forest perform the best.We will show the score and pictures of the measurments

A. F1 Score

The F1-score, a harmonic mean of precision and recall, indicates that the model achieves better balance between precision and recall for the "Great" and "Poor" classes compared to the "Decent" class, with values ranging from 0.64 to 0.91.

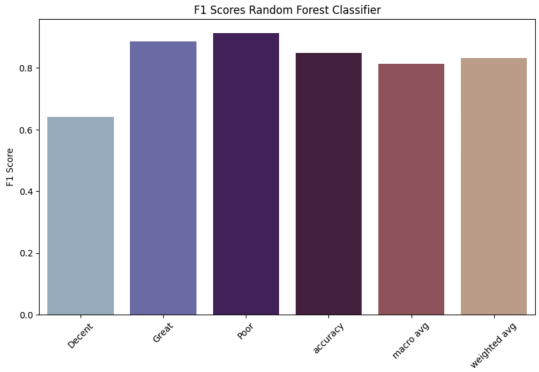

Fig. 7. Random Forest's F1 Score

B. Precision

Precision scores ranging from 0.80 to 0.93 show the model's ability to correctly classify instances of each class, with the "Great" class having the highest precision and the "Decent" class having the lowest.

C. Recall

With recall scores ranging from 0.49 to 0.99, the model exhibits varying degrees of success in identifying instances of each class, demonstrating particularly high recall for the "Great" class and lower recall for the "Decent" class.

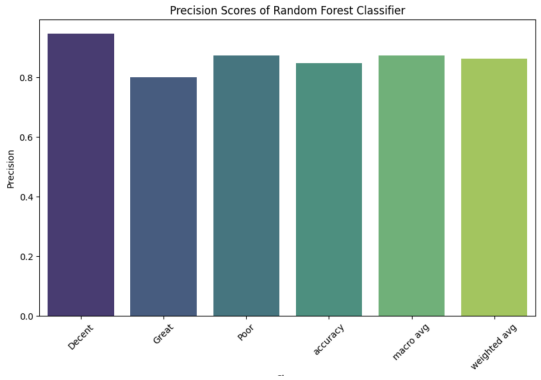

Fig. 8. Random Forest's precision Score

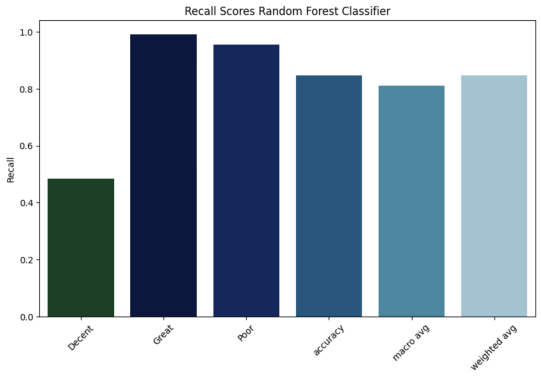

Fig. 9. Random Forest's recall Score

We trained and predicted the placement status of students based on the same dataset and found the True Positive, False Positive, False Negative, True Negative and accuracy of each algorithm.

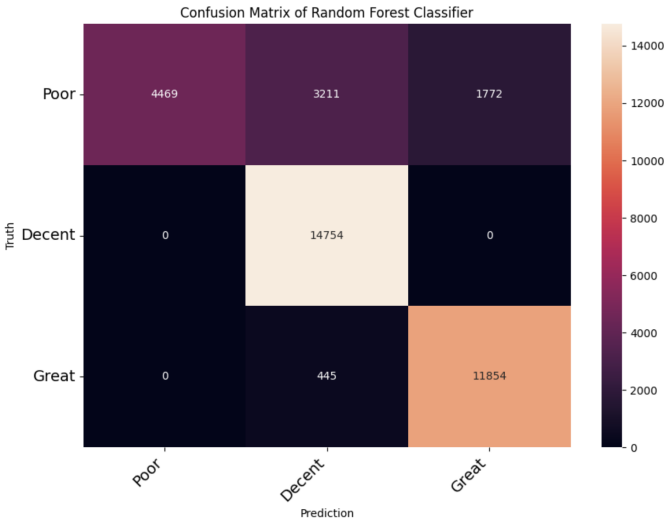

Fig. 10. Random Forest's Confusion Matrix

V. CONCLUSION

Placement prediction system is a system which predicts the placement status of final year BSC students. For data analysis and prediction different machine learning algorithms are used in the python environment. We analyse the accuracy of different algorithms and it is shown in the above table. It is clear that Random Forest gives an accuracy of 85.17 percent.Decision Tree is also good which gives an accuracy of 84.70 based on the given dataset. The accuracy of Machine learning algorithms may differ according to the dataset. From the result from our analysis it is clear that Logistic Regression, Random Forest, KNN are good for classification problems since they all give accuracy of above 75 percent . There can be many things to do in near future for the improvement. like Some recruiters consider other feature scores and history of backlogs which we didn't include in our dataset which can be modify according to the demand, the dataset can be improved by taking real life data.In such rare cases these results may change.

REFERENCES

[1] I. T. Jose, D. Raju, J. A. Aniyankunju, J. James, and M. T. Vadakkel, "Placement prediction using various machine learning models and their efficiency comparison," *International journal of innovative science and research technology*, 2020.

[2] S. Nagamani, K. M. Reddy, U. Bhargavi, and S. Kumar, "Student placement analysis and prediction for improving the education standards by using supervised machine learning algorithms," *J. Crit. Rev*, vol. 7, no. 14, pp. 854–864, 2020.

[3] A. J. Olalekan, P. O. Odion, M. E. Irhebhude, and H. Aminu, "Performance evaluation of machine learning predictive analytical model for determining the job applicants employment status," 2021.

[4] T. Guo, F. Xia, S. Zhen, X. Bai, D. Zhang, Z. Liu, and J. Tang, "Graduate employment prediction with bias," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 670–677.

[5] K. Saraswat, "Students placement prediction using machine learning algorithms," 2022.

[6] L. Sathish and T. Rani, "Student placement prediction using machine learning models (knn, svm, rf, logistic regression)," *Journal of Advances in Data Engineering*, vol. 10, 2023.