Inspiring Excellence

Lab Project Report

CSE422 : Artificial Intelligence

Submitted by:
Habibun Nabi Hemel (22241042)
Ishrak Hamim Mahi(22101792)
Section - 11, Group - 04

Submitted to:
Md Mustakin Alam
Adjunct Lecturer
Department of Computer Science and Engineering

Labib Hasan Khan
Adjunct Lecturer
Department of Computer Science and Engineering

Date of submission: 27/04/24

CONTENTS

FutureFit: ML-Based Predictive Modeling for Campus Placement Success

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

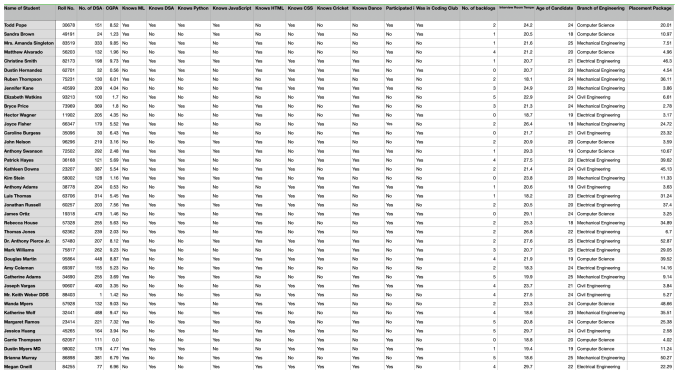
I. Introduction

Our project, FutureFit, uses machine learning to predict the job opportunities and potential salaries for BSc students based on various data. It's like a guide for students to track their goals and see what their future might look like. If a student finds out that the job package predicted by our model isn't satisfying, they can work extra hard to improve it. Our model provides insights into how certain actions can affect their future earnings, using data from past graduates. We came up with this idea because, as third-year students ourselves, we often worry about our future prospects. We wanted a tool that could give us a reality check on our current progress and motivate us to strive for a better future. So, FutureFit is designed to be that helpful tool for students like us, guiding us toward making informed decisions and encouraging us to work towards a successful career.

II. Dataset

A. 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 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.



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	Was in Coding Club	No. of backlogs	Interviews Known Human	Age of Candidate	Branch of Engineering	Placement Package
Todd Pope	99978	151	8.52	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes	2	24.2	24	Computer Science	20.81
Sandra Brown	49181	24	1.23	Yes	No	No	Yes	No	No	Yes	No	No	Yes	1	20.5	18	Computer Science	10.87
Mrs. Amanda Singleton	93519	333	9.80	No	Yes	Yes	No	Yes	Yes	No	No	No	No	1	21.6	25	Mechanical Engineering	7.51
Matthew Alvarado	56203	152	1.96	No	No	Yes	No	Yes	No	No	No	Yes	No	4	21.2	20	Computer Science	4.96
Christine Smith	82173	168	8.73	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	1	20.7	21	Electrical Engineering	45.3
Dustin Hernandez	62751	22	0.56	No	Yes	Yes	No	Yes	Yes	No	No	Yes	No	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	40089	209	4.04	No	No	No	No	Yes	Yes	Yes	Yes	Yes	No	3	24.9	23	Mechanical Engineering	3.86
Elizabeth Watkins	00213	100	1.7	No	No	No	No	No	Yes	Yes	No	No	No	0	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	2.76
Hector Wagner	11902	205	4.35	No	No	No	No	Yes	No	No	No	Yes	Yes	0	18.7	19	Electrical Engineering	3.17
Joyce Fisher	66547	179	5.52	Yes	Yes	No	No	No	No	Yes	No	Yes	No	2	20.4	18	Mechanical Engineering	24.72
Caroline Burgess	99386	39	6.40	Yes	Yes	Yes	No	Yes	No	No	No	No	Yes	0	21.7	21	Civil Engineering	20.32
John Nelson	99286	219	5.16	No	No	Yes	Yes	Yes	Yes	No	No	Yes	No	2	23.9	20	Computer Science	3.89
Anthony Swanson	72582	292	2.46	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	No	1	20.3	19	Computer Science	10.87
Patrick Hayes	36168	121	5.89	Yes	Yes	Yes	No	No	No	Yes	Yes	No	Yes	4	27.5	23	Electrical Engineering	39.82
Kathleen Downs	23257	367	5.24	No	Yes	Yes	No	Yes	Yes	No	Yes	No	No	2	21.4	24	Civil Engineering	45.10
Kim Stein	58002	128	1.16	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No	0	23.8	20	Mechanical Engineering	11.30
Anthony Adams	38778	204	5.53	No	No	Yes	No	No	Yes	Yes	Yes	Yes	Yes	1	20.6	18	Civil Engineering	3.83
Lois Thomas	65795	314	6.45	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	No	Yes	No	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	9.83	No	Yes	Yes	No	Yes	No	No	No	Yes	Yes	2	25.3	18	Mechanical Engineering	34.89
Thomas Jones	62382	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	No	Yes	Yes	Yes	Yes	No	Yes	2	27.6	25	Electrical Engineering	52.87
Mark Williams	75977	262	9.23	No	Yes	No	No	No	No	Yes	No	Yes	Yes	3	20.7	25	Electrical Engineering	29.85
Douglas Martin	95864	448	8.87	Yes	No	No	No	Yes	Yes	Yes	No	No	Yes	4	21.9	19	Computer Science	39.52
Amy Coleman	69097	195	5.23	No	No	No	No	Yes	Yes	No	No	No	No	2	18.3	24	Electrical Engineering	14.16
Catherine Adams	34680	255	5.89	Yes	No	Yes	No	Yes	No	No	No	Yes	No	5	19.9	25	Mechanical Engineering	9.14
Joseph Vargas	99087	406	3.39	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	4	23.7	21	Civil Engineering	3.84
Mr. Keith Weber DDS	88483	1	1.42	No	No	No	No	Yes	No	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	48.86
Katherine Wolf	32441	468	9.47	No	Yes	Yes	No	Yes	No	No	No	No	Yes	4	19.6	23	Mechanical Engineering	39.51
Margaret Ramos	25414	221	7.30	Yes	No	Yes	No	Yes	No	Yes	No	No	Yes	5	20.6	24	Computer Science	20.39
Jessica Hoang	45265	164	3.94	No	No	Yes	No	Yes	Yes	Yes	No	No	Yes	5	29.7	24	Civil Engineering	2.55
Carrie Thompson	62057	111	0.0	No	No	No	No	No	No	Yes	No	Yes	No	0	18.6	23	Computer Science	4.82
Dustin Myers MD	99082	178	4.77	Yes	No	Yes	No	No	No	Yes	No	Yes	Yes	1	19.4	19	Computer Science	11.24
Brianna Murray	86688	381	6.79	Yes	No	No	No	Yes	Yes	No	No	Yes	No	5	18.6	25	Mechanical Engineering	50.27
Megan Orecll	84265	77	8.96	No	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	4	29.7	22	Electrical Engineering	22.29

Fig. 1. Dataset

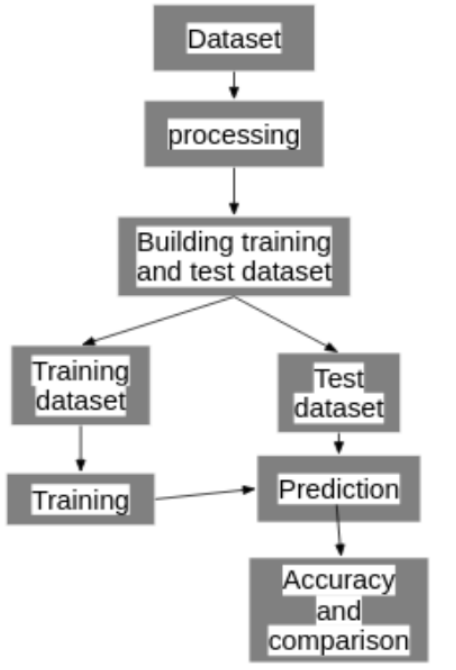


Fig. 2. Dataset used for Prediction and Analysis

B. 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. 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.

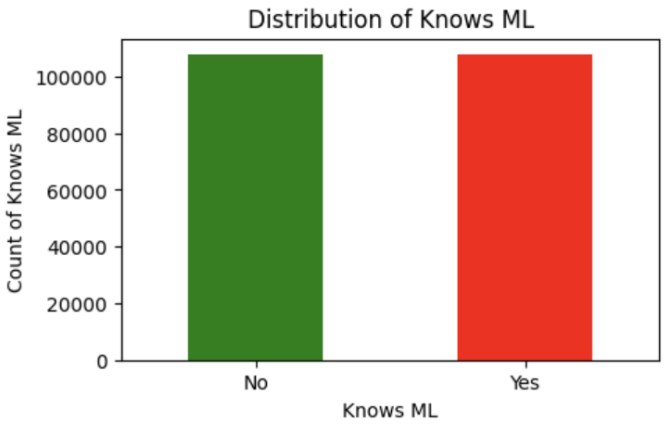


Fig. 3. Distribution of Data

C. Source

1. Link of the Dataset (Google Drive link): CLICK HERE
2. Reference: Kaggle Link of the Dataset CLICK HERE

D. Dataset Description

The dataset includes details like student names, roll numbers, academic scores, and skills, with 220,000 entries across 19 columns. It contains both numerical and categorical data, which we handle during data processing.The features ”Name,” ”Knows ML,” ”Knows DSA,” ”Knows Python,” ”Knows JavaScript,” ”Knows HTML,” ”Knows CSS,” ”Was in Coding Club,” and ”Branch of Engineering” are categorical. Conversely, ”No. of DSA questions,” ”CGPA,” ”No. of backlogs,” ”Age of Candidate,” and ”Placement Package” are numerical features. Although it’s suitable for regression analysis due to its continuous values, we’re using it for multi-class classification. Our aim is to categorize job placements into ”Great,” ”Decent,” or ”Poor” to evaluate career services’ effectiveness and enhance overall employment outcomes. We’ve divided the dataset based on salary ranges: below 10 LPA as ”Poor,” 10 to 20 LPA as ”Decent,” and over 20 LPA as ”Great.” Additionally, we employ a heatmap to visualize feature relationships, aiding in identifying irrelevant features.

E. 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.

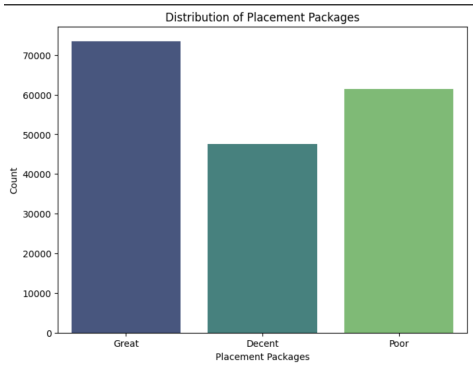


Fig. 4. Classes of Output

F. Data Pre-Processing

Data processing is the conversion of raw data into a meaningful format through steps like collection, cleaning, analysis, and interpretation, ultimately aiming to derive valuable insights and support decision-making.

G. 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.

H. 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.

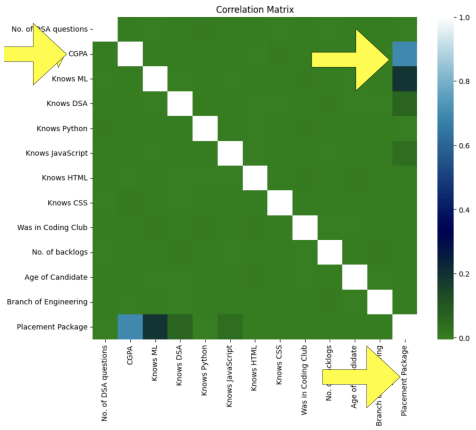


Fig. 5. Correlation Matrix

I. 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.

J. Problems

When it comes to converting categorical values into numerical ones, we have options like one-hot encoding and label encoding. However, we initially chose one-hot encoding and faced some challenges. One-hot encoding can make our data too big and sparse, which might slow down computations and waste memory, especially with large categorical variables. Also, it could cause problems like multicollinearity in some machine learning models because of the linear relationship between the encoded features.
For example, let's say we're dealing with a student's major subject, like Computer Science (CSE), Electrical and Electronics Engineering (EEE), Mechanical Engineering, and Civil Engineering. With one-hot encoding, we'd have to ask four questions: Are you from CSE? If yes, put 1 and 0 otherwise. Are you from EEE? If yes, put 1 and 0 otherwise. Are you from Mechanical Engineering? If yes, put 1 and 0 otherwise. Are you from Civil Engineering? If yes, put 1 and 0 otherwise. But if we switch to label encoding, we simplify it to just one question: "Are you from CSE?" If the answer is yes, we assign 1. This method saves time and is more straightforward compared to one-hot encoding.

III. Feature Scaling

In our work,we do not concern about feature scaling because we find that feature scaling may not be necessary for decision trees, random forests, logistic regression, k-nearest neighbors (KNN), and naive Bayes. These algorithms inherently handle feature scales effectively or are not sensitive to feature scales due to their underlying mechanisms. Decision trees and random forests base decisions on feature thresholds, while logistic regression adjusts coefficients to accommodate feature scale. KNN calculates distances between data points, primarily relying on relative distances, and naive Bayes assumes feature independence given the class label, making feature scale inconsequential to its probability calculations. Hence, we often omit feature scaling when employing these algorithms.

IV. Dataset Splitting

Data splitting allows us to assess model performance on unseen data, preventing overfitting and enabling hyperparameter tuning. It helps strike a balance between bias and variance while simulating real-world scenarios in model development.

A. 70% trainning set and 30% testing set

After splitting the data into a 70-30 train-test split using the trainTestSplit function from sklearn.modelSelection, we have separated the features (X) from the target variable (y). The training set (Xtrain and ytrain) contains 127,765 samples, while the test set (Xtest and ytest) contains 54,757 samples. Both the training and test sets include 12 features.

V. Model Training

We chose decision tree, random forest, logistic regression, naive Bayes, and k-nearest neighbors (KNN) algorithms for our analysis. We imported these models using the scikit-learn library. Our dataset was split into an 70-30 ratio, with 70 of the data used to train the models and 30 reserved for testing purposes.

A. Prediction System

This study employs machine learning methods to forecast students' placement outcomes using a dataset. The predictive factors in the dataset include the number of DSA questions attempted, CGPA, proficiency in programming languages (ML, DSA, Python, JavaScript, HTML, CSS), participation in coding clubs, number of backlogs, candidate's age, and engineering branch. Placement status prediction is accomplished through machine learning algorithms such as Logistic Regression, Random Forest, KNN, and SVM.

B. 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.

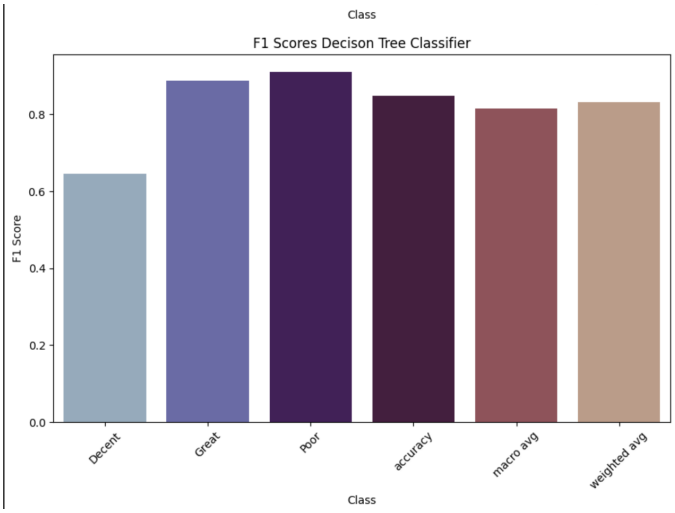


Fig. 6. Decision Tree's F1 Score

Choosing the appropriate hyperplane is critical in Decision Tree classification. Scikit-learn, a Python library, facilitates the implementation of various machine learning algorithms, including Decision Trees.

Advantages:

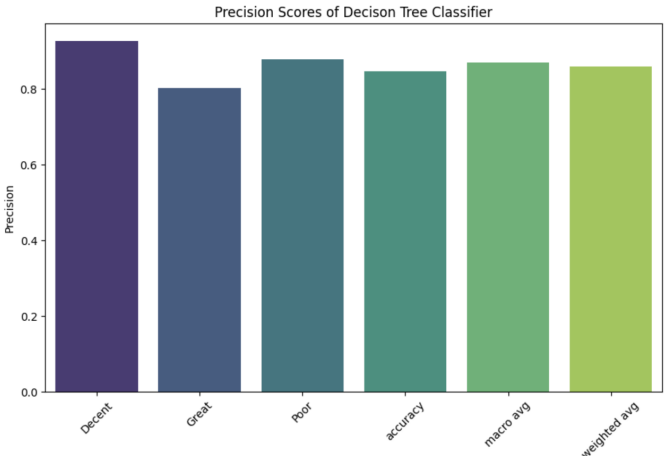


Fig. 7. Decision Tree's Precision Score

- 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.
- Efficient memory usage.

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.

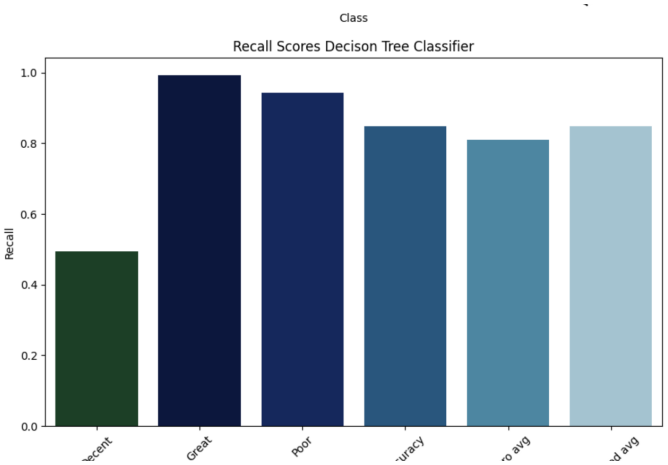


Fig. 8. Decision Tree's Recall Score

C. Random Forest

We have a plethora of classification algorithms at our disposal, including, but not limited to, SVM, Logistic regression, decision trees and Naive Bayes classifier, just to name a few. But, in the hierarchy of classifiers, the Random Forest Classifier sits near the top. The random forest classifier is a group of individual decision trees and so, we shall look into how decision trees work.

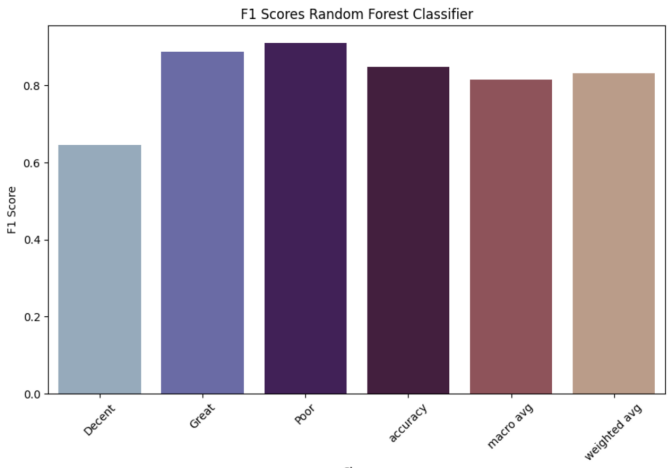


Fig. 9. Random Tree's F1 Score

It is basically a flowchart-like structure in which each node excluding the leaf node is a test on a feature (i.e, what will be the outcome if some activity, such as flipping a coin, is done), leaf nodes are used to represent the class label (the decision taken after all features are computed) and branches represent the conjunctions of features that lead to those class labels. The classification rules of a decision tree are the paths from the root node to the leaf node.

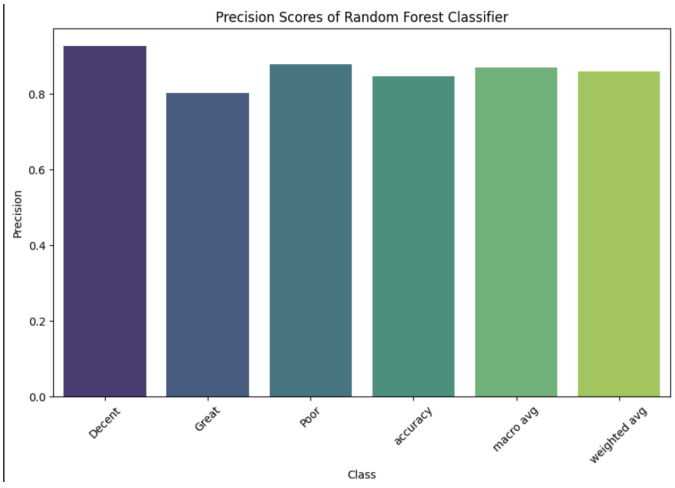


Fig. 10. Random Tree's Precision Score

So then, now let us look into random forest classifiers. As mentioned earlier, it is a collection of decision trees. The basic idea behind random forest is "the wisdom of the crowds". It is a powerful concept wherein a large number of uncorrelated models, or in this case trees, operating as a group, would provide a much more solid output than any of the constituent

models.
So, in a random forest, each individual tree with different properties and classification rules would try to find an appropriate class label for the problem. Each tree would give out its own answer. A voting is done within the random forest to see which class label received the most votes. The class label with the most votes would be considered the final class label for the problem. This provides a more accurate model for class label prediction.

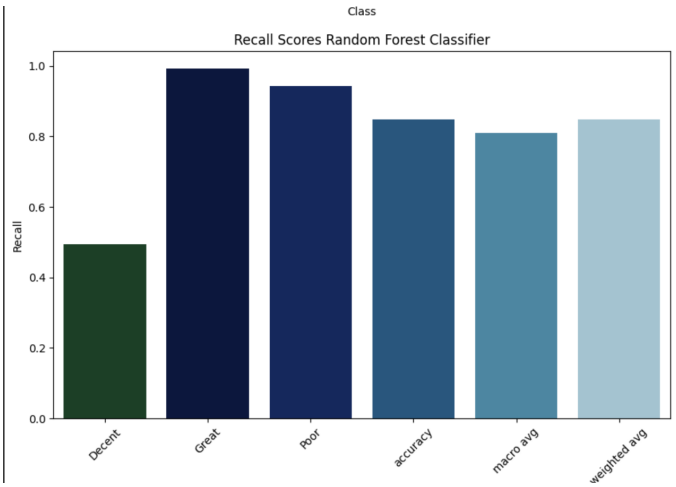


Fig. 11. Random Tree's Recall Score

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

D. Logistic Regression

Logistic regression is a classification technique and it is very good for binary classification. It's decision boundary which is generally linear derived based on probability interpretation. The results are in a nonlinear optimization problem for parameter estimation. Parameters can be estimated by maximising the expression using any nonlinear optimization solver.

The goal of this technique is given a new data point, and predict the class from which the data point is likely to have originated. Input features can be quantitative or qualitative.

Advantages:

- Logistic Regression is good for linearly separable dataset
- It is efficient to train and easy to interpret and implement.

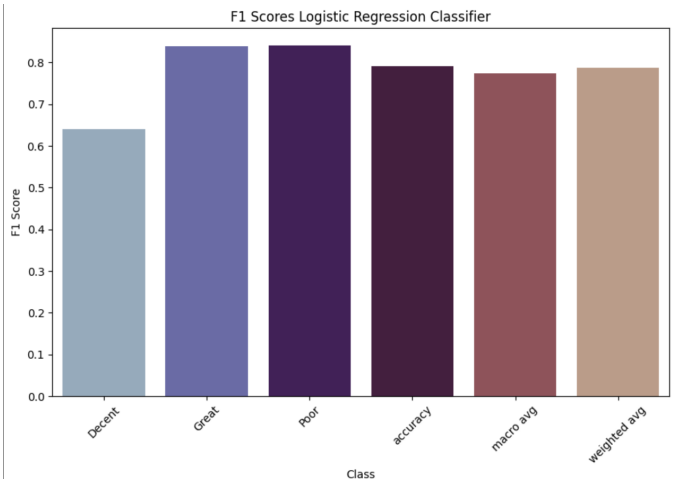


Fig. 12. Logistic Regression's F1 Score

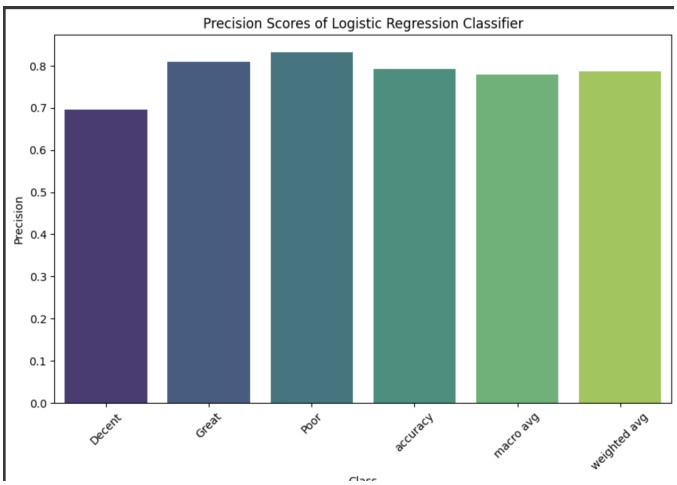


Fig. 13. Logistic Refression's Precision Score

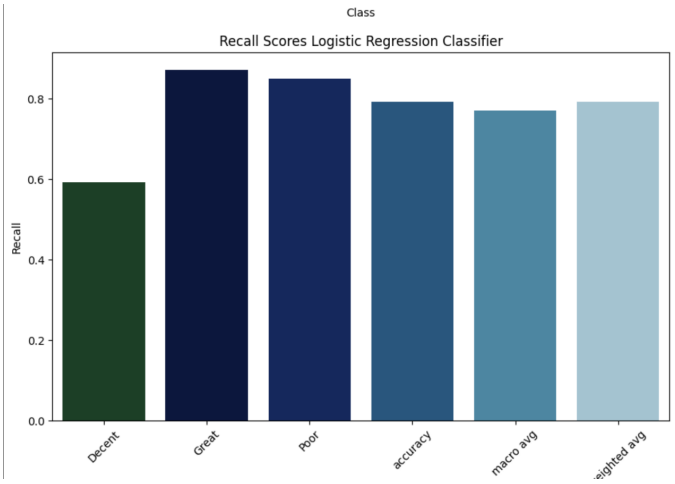


Fig. 14. Logistic Regression's Recall Score

- It not only gives a measure of how relevant a predictor is, but also its direction of association.
- Less prone to overfitting.

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

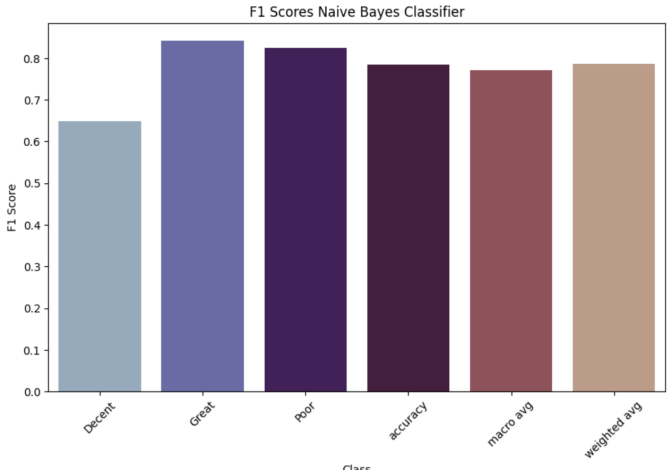


Fig. 15. Naives Bayes's F1 Score

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.

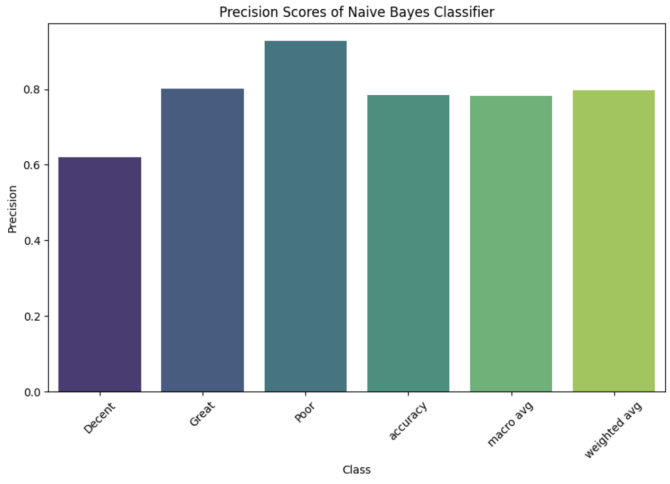


Fig. 16. Naives Bayes's Precision Score

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:

- Assumes that features are independent, which may not hold true in real-world scenarios.
- 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.

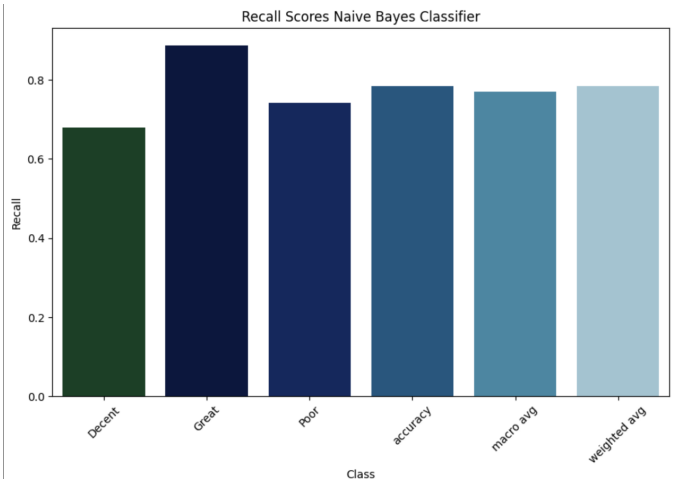


Fig. 17. Naives's Recall Scores

F. KNN

KNN stands for k-nearest neighbors. This is a simple algorithm that can be used to solve classification and regression type problems. It is a supervised machine learning algorithm, meaning labels are used.

The basic working of this algorithm revolves around the concept that similar things are always in close proximity within each other. So, for this algorithm to provide any fruitful results, this is an assumption that is taken. Similarity in KNN is expressed using distance, closeness or proximity. A mathematical approach is taken for distance, which is usually the Euclidean Distance as it is the common and familiar choice.

Advantages:

- This is a fairly simple and easy-to-implement algorithm
- 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.

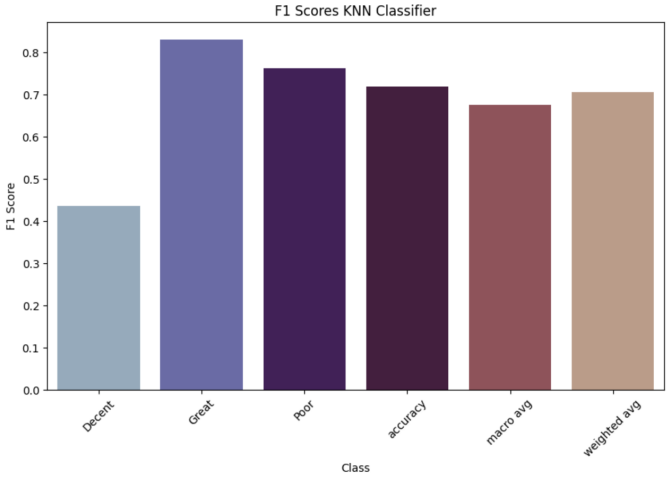


Fig. 18. KNN's F1 Score

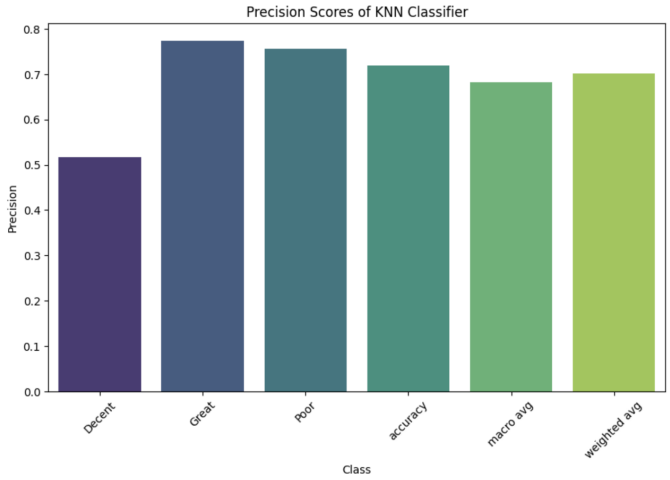


Fig. 19. KNN's Precision Score

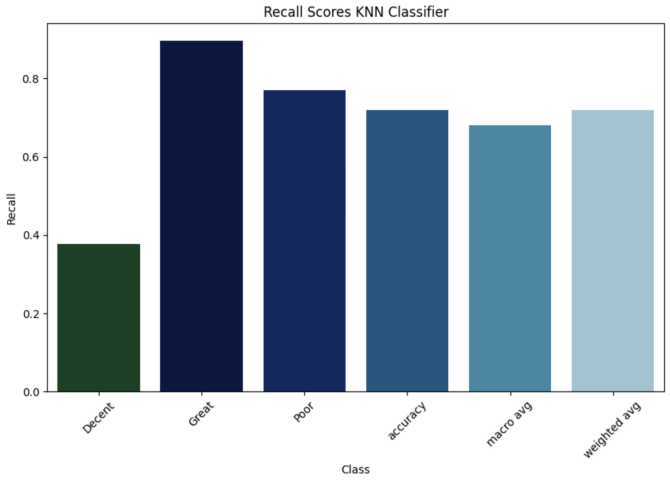


Fig. 20. KNN's Recall Score

VI. Result and Comparision analysis

The table summarizes the accuracy scores of various machine learning models. Both Decision Tree and Random Forest models achieve high accuracy rates, with Random Forest slightly outperforming Decision Tree. Logistic Regression and Naive Bayes models exhibit lower but still respectable accuracy scores. However, the KNN model demonstrates the lowest accuracy among the models listed. Overall, Random Forest appears to be the most accurate model among those evaluated.

A. accuracy

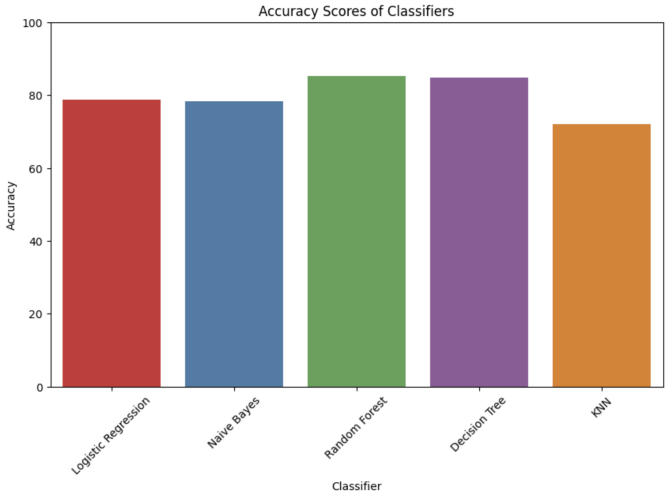


Fig. 21. Accuracy in Bars

The table-1 below illustrates the outcomes of applying different machine learning algorithms, with Random Forest emerging as the clear winner. Through thorough analysis encompassing Decision Tree, Random Forest, Logistic Regression, Naive Bayes, and KNN, Random Forest consistently outperformed its counterparts across various performance metrics, including F1 score, precision, recall, and accuracy. By integrating these measurements comprehensively, Random Forest demonstrated superior performance. Additionally, we will provide visual representations of these metrics to further elucidate the comparative performance of the algorithms.

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

TABLE I
ACCURACY

B. Confusion matrix

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

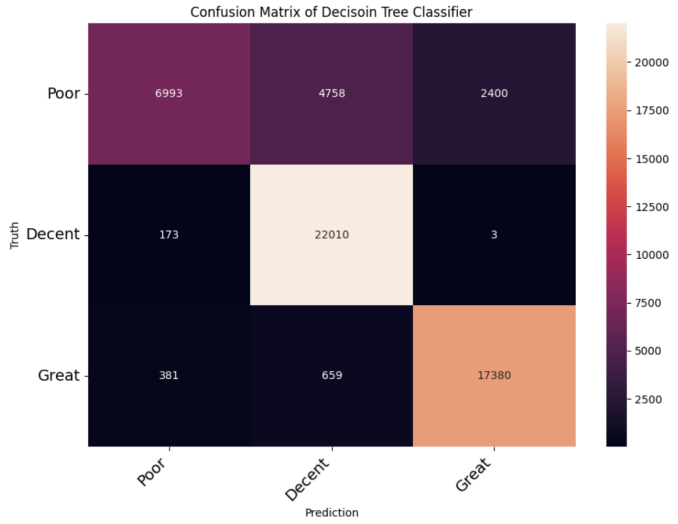


Fig. 22. Decision Tree's Confusion Matrix

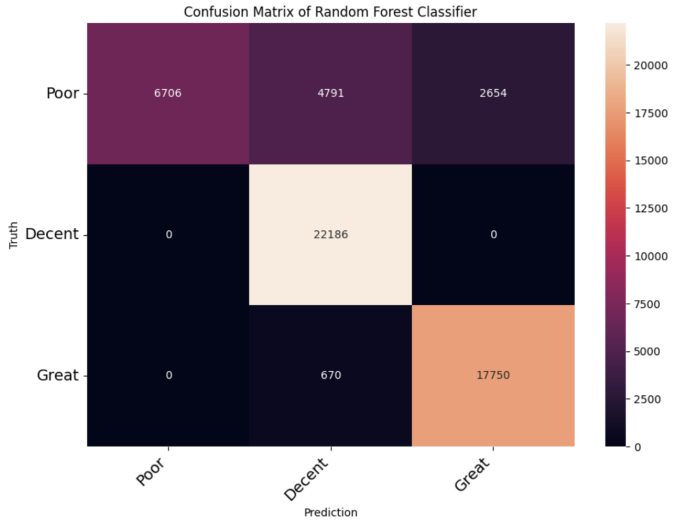


Fig. 23. Random Forest's Confusion Matrix

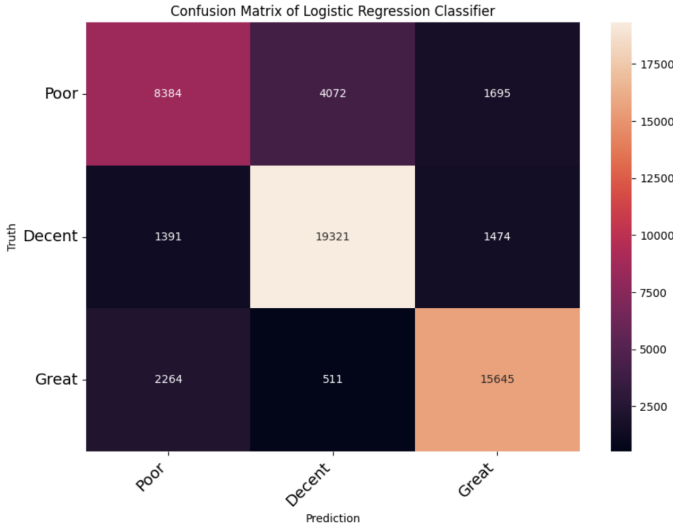


Fig. 24. Logistic Regression's Confusion Matrix

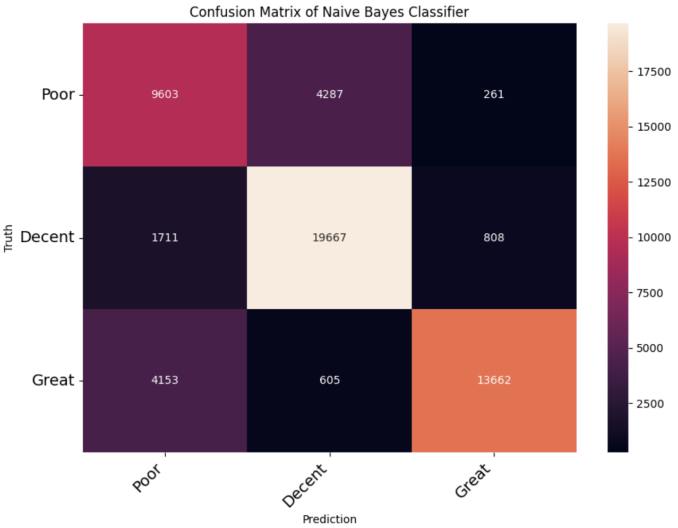


Fig. 25. Naives Bayes's Confusion Matrix

algorithm.A confusion matrix is a table that is used to evaluate the performance of a classification model. It summarizes the predictions of a model on a classification problem by comparing predicted labels with true labels. The matrix consists of rows and columns representing actual and predicted classes, respectively. Each cell in the matrix shows the count or proportion of observations that fall into a particular combination of predicted and actual classes, providing insights into the model's performance, including its accuracy, precision, recall, and other metrics.

C. F1 Score

The F1 score is a metric that combines precision and recall into a single value, providing a balance between them. It is particularly useful in evaluating the performance of classification models, taking into account both the number of true positive predictions and the ability to avoid false positives and false negatives.

D. Precision

Precision entails the meticulousness and refinement in ensuring that measurements, calculations, or descriptions are as accurate and exact as possible, minimizing errors and uncertainties.

E. Recall

In machine learning, recall refers to the proportion of actual positive cases (instances belonging to a specific class) that were correctly identified by a classification model. It indicates the model's ability to capture all relevant instances of a certain class within a dataset, thereby minimizing false negatives.

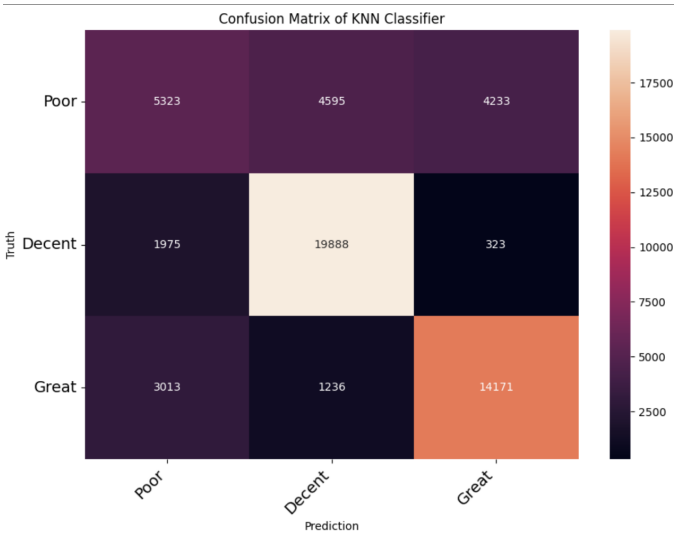


Fig. 26. KNN's Confusion Matrix

VII. 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.