PLACEMENT PREDICTION MODEL

Deotale Chinmay Jayendra

Roll No.- 60, GR No. – 11910759

Under the guidance of : **Prof. Ashwini Barbadekar Ma'am**

Department of Electronics and Telecommunication Engineering

Abstract -A placement predictor is to be designed to calculate the possibility of a student being placed in a company, subject to the criterion of the company. The placement predictor takes many parameters which can be used to assess the skill level of the student. While some parameters are taken from the university level, others are obtained from tests conducted in the placement management system itself. Combining these data points, the predictor is to accurately predict if the student will or will not be placed in a company.

Keywords — : Machine learning, Data Science, prediction, training, testing, SVM ,Logistic Regression,.

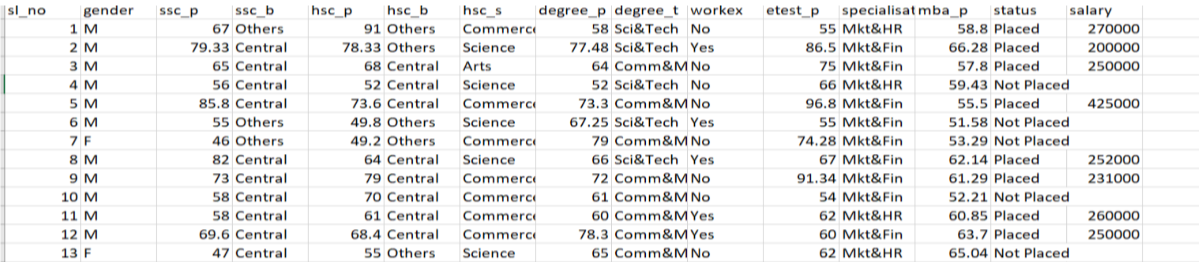
INTRODUCTION - Data from past students are used for training the predictor. But the problem was to find a suitable classification algorithm that could do the job with maximum accuracy for our data set. Different algorithms have different accuracy depending on the type of problem it has to solve and the data set it has to work with. So, we decided to select four algorithms, namely KNN, SVM, Logistic Regression and Random Forest and to compare the accuracy levels of each of these algorithms, with respect to our problem and data set. The result of this test would help us in determining which algorithm to use while implementing our predictor in the placement management system.

For this, we trained each of the algorithms with the data set that we acquired and tested it against some test data to find the accuracy of the algorithms. For each algorithm, we can easily obtain the True Positive, True Negative, False Positive and False Negative. With these four values, it was a matter of finding the accuracy using the accuracy equation.

We aim to develop a placement predictor as a part of making a placement management system at college level which predicts the probability of students getting placed and helps in uplifting their skills before the recruitment process starts. We are using machine learning for the placement prediction. We consider Support Vector Machine(SVM), Logistic Regression, to classify students into appropriate clusters and the result would help them in improving their profile. And accuracy of respected algorithms are noted and With the comparison of various machine learning techniques, this would help both recruiters as well as students during placements and related activities

A. Prediction system In this paper we use machine learning techniques to predict the placement status of students based on a dataset. The parameters in the dataset which are considered for the prediction are Quantitative scores, Logical Reasoning scores, Verbal scores, Programming scores, CGPA, No. of hackathons attended, No. of certifications and current backlogs number. The placement prediction is done by machine learning using Logical Regression, Random Forest, KNN, SVM.

B. Sample Dataset –



sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisat	mba_p	status	salary
1	M	67	Others	91	Others	Commerc	58	Sci&Tech	No	55	Mkt&HR	58.8	Placed	270000
2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000
3	M	65	Central	68	Central	Arts	64	Comm&M	No	75	Mkt&Fin	57.8	Placed	250000
4	M	56	Central	52	Central	Science	52	Sci&Tech	No	66	Mkt&HR	59.43	Not Placed	
5	M	85.8	Central	73.6	Central	Commerc	73.3	Comm&M	No	96.8	Mkt&Fin	55.5	Placed	425000
6	M	55	Others	49.8	Others	Science	67.25	Sci&Tech	Yes	55	Mkt&Fin	51.58	Not Placed	
7	F	46	Others	49.2	Others	Commerc	79	Comm&M	No	74.28	Mkt&Fin	53.29	Not Placed	
8	M	82	Central	64	Central	Science	66	Sci&Tech	Yes	67	Mkt&Fin	62.14	Placed	252000
9	M	73	Central	79	Central	Commerc	72	Comm&M	No	91.34	Mkt&Fin	61.29	Placed	231000
10	M	58	Central	70	Central	Commerc	61	Comm&M	No	54	Mkt&Fin	52.21	Not Placed	
11	M	58	Central	61	Central	Commerc	60	Comm&M	Yes	62	Mkt&HR	60.85	Placed	260000
12	M	69.6	Central	68.4	Central	Commerc	78.3	Comm&M	Yes	60	Mkt&Fin	63.7	Placed	250000
13	F	47	Central	55	Others	Science	65	Comm&M	No	62	Mkt&HR	65.04	Not Placed	

Figure 1 :Sample Dataset

II. Implementation Details

A. *Algorithm and Block Diagram*

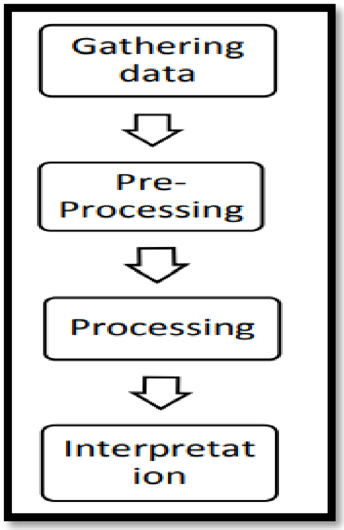


Figure 2. Flow chart of the technique

The whole approach is depicted by the above flowchart.

3.1 Data gathering : The sample data has been collected from our college placement department which consists of all the records of previous years students. The dataset collected consist of over 1000 instances of students.

3.2 Pre processing Data pre processing is a technique that is used to convert raw data into a clean dataset. The data is gathered from different sources is in raw format which is not feasible for the analysis. Pre-processing for this approach takes 4 simple yet effective steps.

3.2.1 Attribute selection :Some of the attributes in the initial dataset that was not pertinent (relevant) to the experiment goal were ignored. The attributes name, roll no, credits, backlogs, whether placed or not, b.tech % ,gender are not used. The main attributes used for this study are credit , backlogs , whether placed or not, b.tech %.

3.2.2 Cleaning missing values : In some cases the dataset contain missing values . We need to be equipped to handle the problem when we come across them. Obviously you could remove the entire line of data but what if you're inadvertently removing crucial information? after all we might not need to try to do that. one in every of the foremost common plan to handle the matter is to require a mean of all the values of the same column and have it to replace the missing data. The library used for the task is called Scikit Learn preprocessing. It contains a class called Imputer which will help us take care of the missing data.

3.2.3 Training and Test data Splitting the Dataset into Training set and Test Set Now the next step is to split our dataset into two. Training set and a Test set. We will train our machine learning models on our training set, i.e our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to examine how accurately it will predict. A general rule of the thumb is to assign 80% of the dataset to training set and therefore the remaining 20% to test set.

3.3 Processing Processing in this paper's sense is applying different algorithms to the data to find the best results .

3.3.1 Naive Bayes :

This is an easy technique for building classifiers: models that assign class labels to downside instances, painted as vectors of feature values, where class labels are drawn from a few finite set. There is no single algorithm for training such classifiers, however a family of algorithms which is based on a standard principle: all Naive Bayes classifiers assume that value of a particular feature is independent of the value of other feature, given class variable. For example, a fruit could also be thought of be an apple if it is red, round, and about 11 cm in diameter. A Naive Bayes classifier considers every of these options to contribute severally to the likelihood that this fruit is AN apple, no matter any potential correlations between the colour, roundness, and diameter features.

Working of Naive Bayes Algorithm

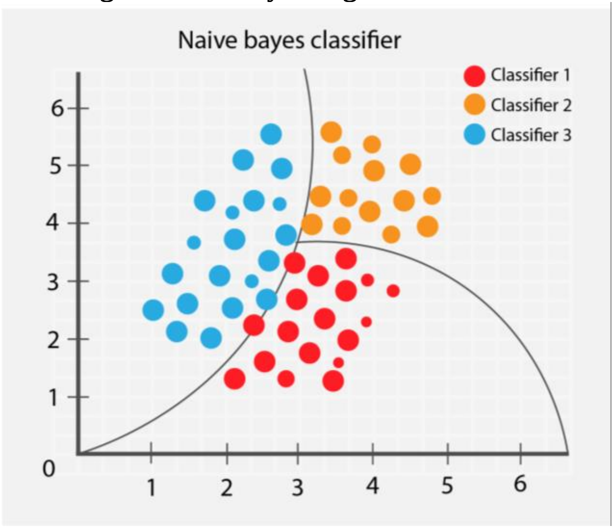


Figure 3. : Naïve Bayes Cluster

Step 1: Scan the dataset (storage servers) retrieval of required data for mining from the servers such as database, cloud, excel sheet etc.

Step 2: Calculate the probability of every attribute value. [n, n_c, m, p] Here for each attribute we calculate the probability of occurrence using the following formula. (mentioned in the next step). For each class (Course) we should apply the formulae. Step3:P(attributevalue(ai)/subjectvaluevj)=(n_c+mp)/(n+m) apply the above formulae Where: n = no. of training examples for which v = vj nc = no. of examples where v = vj and a = ai p = a priori estimate for P(aivj) m = the equivalent sample size

Step 4: Multiply the probabilities by p for each class, here we multiple the results of each attribute with p and final results are used for classification.

Step 5: Compare the values and classify the attribute values to 1 of the predefined set of class.

3.3.2 SVM

SVM stands for Support Vector Machine. It is also a supervised machine learning algorithm that can be used for both classification and regression problems. However, it is mostly used for classification problems. A point in the n-dimensional space is a data item where the value of each feature is the value of a particular coordinate. Here, n is the number of features you have.

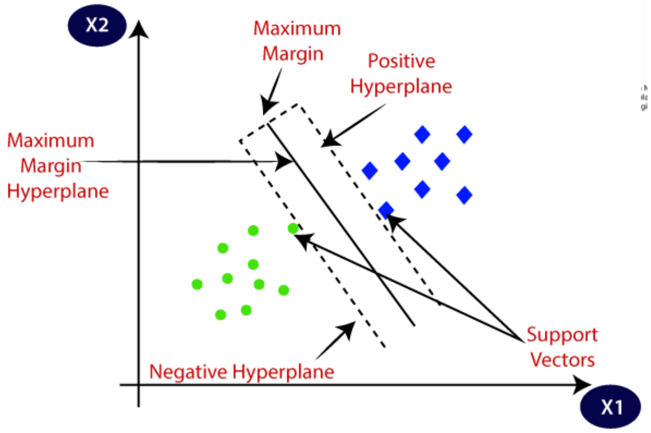


Fig. 4: Support Vector Machine

After plotting the data item, we perform classification by finding the hyper-plane that differentiates the two classes very well. Now the problem lies in finding which hyper-plane to be chosen such that it is the right one. Scikit-learn is a library in Python which can be used to implement various machine learning algorithms and SVM too can be used using the scikit-learn library.

¬ **Advantages:**

- This algorithm performs best when there is a clear margin of separation
- Effective in high dimensional spaces
- If the number of dimensions is greater than the number of samples, the algorithm would be able to perform better
- It is memory efficient

¬ **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.

3.2.3 . 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.

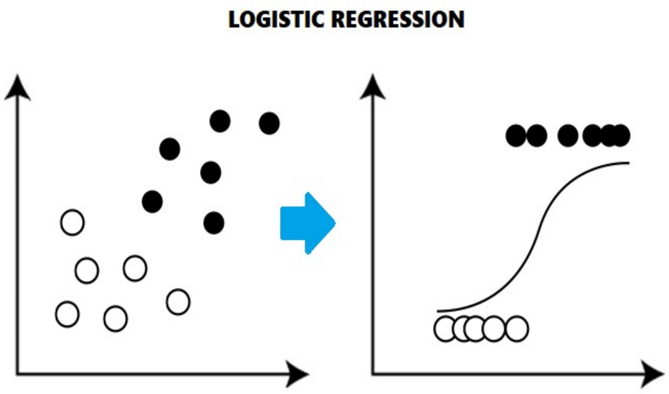


Fig. 5: Logistic Regression

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.

Instead of a hyperplane or straight line, the logistic regression uses the logistic function to obtain the output of a linear equation between 0 and 1. The function is defined as logistic(x)=1/(1+exp(-x)) Fig 2:- Logistic regression

¬ **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.
- Less prone to overfitting

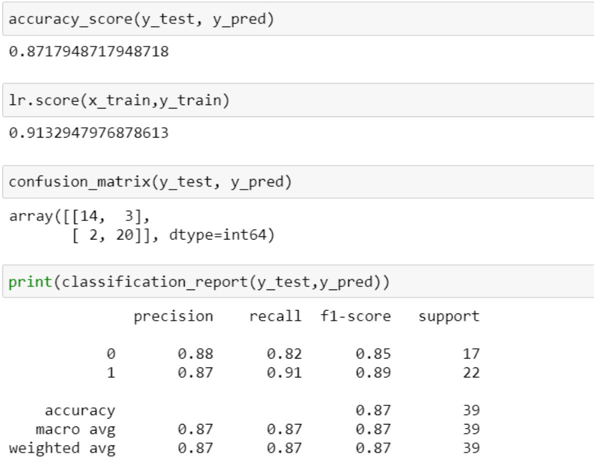
¬ **Disadvantages :**

- 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.

III. Results and Discussions

1) *Logistic regression*

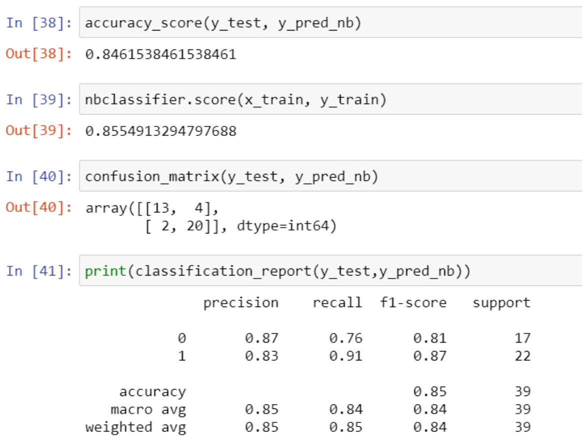


```
accuracy_score(y_test, y_pred)

0.8717948717948718

lr.score(x_train,y_train)

0.9132947976878613

confusion_matrix(y_test, y_pred)

array([[14,  3],
       [ 2, 20]], dtype=int64)

print(classification_report(y_test,y_pred))

              precision    recall  f1-score   support

           0       0.88      0.82      0.85        17
           1       0.87      0.91      0.89        22

    accuracy                           0.87        39
   macro avg       0.87      0.87      0.87        39
weighted avg       0.87      0.87      0.87        39
```

Figure 6 : accuracy model score of Logistic Regression

As seen in figure 6 Logistic regression have a very high model accuracy of 91.3% and also high recall and precision value

2) *SVM*

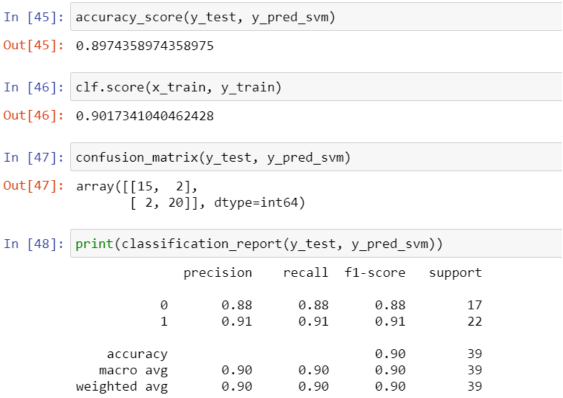


```
In [38]: accuracy_score(y_test, y_pred_nb)

Out[38]: 0.8461538461538461

In [39]: nbclassifier.score(x_train, y_train)

Out[39]: 0.8554913294797688

In [40]: confusion_matrix(y_test, y_pred_nb)

Out[40]: array([[13,  4],
                [ 2, 20]], dtype=int64)

In [41]: print(classification_report(y_test,y_pred_nb))

                       precision    recall  f1-score   support

                    0       0.87      0.76      0.81        17
                    1       0.83      0.91      0.87        22

             accuracy                           0.85        39
            macro avg       0.85      0.84      0.84        39
         weighted avg       0.85      0.85      0.84        39
```

Figure 7 : accuracy model score of Support Vector Machine

As seen in figure 4 Logistic regression have a high model accuracy of 86 % and also high recall and precision value. But the problem with SVM occurs because it is inconsistent with its result

3) *Naïve bayes*



```
In [45]: accuracy_score(y_test, y_pred_svm)

Out[45]: 0.8974358974358975

In [46]: clf.score(x_train, y_train)

Out[46]: 0.9017341040462428

In [47]: confusion_matrix(y_test, y_pred_svm)

Out[47]: array([[15,  2],
                [ 2, 20]], dtype=int64)

In [48]: print(classification_report(y_test, y_pred_svm))

                       precision    recall  f1-score   support

                    0       0.88      0.88      0.88        17
                    1       0.91      0.91      0.91        22

             accuracy                           0.90        39
            macro avg       0.90      0.90      0.90        39
         weighted avg       0.90      0.90      0.90        39
```

Figure 8 : accuracy model score of Naïve Bayes Model

As seen in figure 4 Naïve Bayes Model compliments the model perfectly with high Model accuracy as high as 90 % . It also shows very high and consistent Precision , Recall and F1-score

- The data is first trained and then tested with all Three algorithms and out of all SVM gave more accuracy with **89.75**, Logistic regression with **87.18** percent accuracy and Naïve bayes with accuracy of **85.6.**

IV. Future Scope

This model can be further modified using other Algorithms and Deploying for Practical uses

V. Conclusion

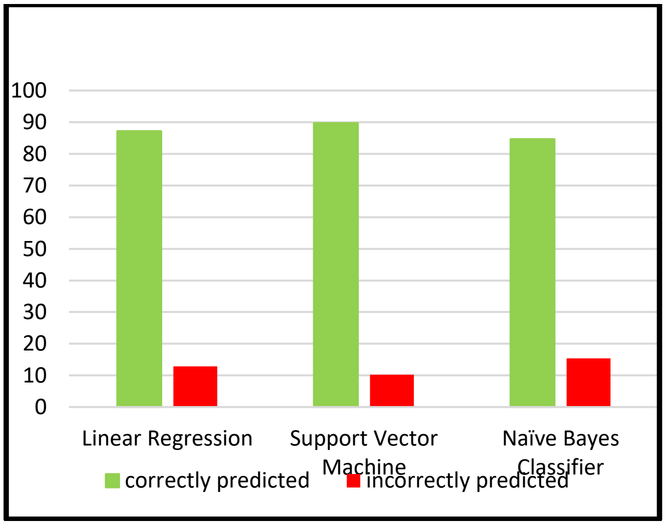


Fig. 9: Comparison between All three Models

- We conclude that Logistic Regression works better with better accuracy but difference in scores is highest among three
- Gaussian Naive Bayes was less accurate but the difference in known and unknown data was lesser.

- But, SVM gave better accuracy with least difference in score. So, Our final model would use SVM for Student Placement Prediction.

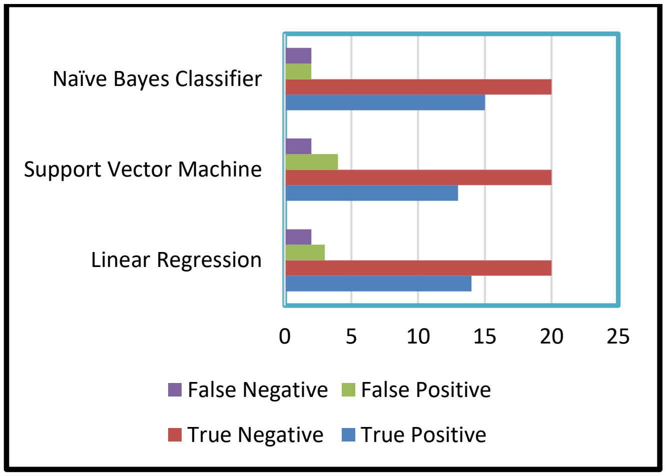


Figure 10. Comparison for TP,TN, FP,FN for all Algorithm

Acknowledgment

We wish to express our deep gratitude and sincere thanks to our guide and teacher "**Prof. (Dr.) Ashwinee Barbadekar** " for his invaluable guidance, constant encouragement and immense motivation which has sustained our efforts at all stages of this project.
We would also like to thank our parents for their constant support.

References

[1]. Shreyas Harinath, Aksha Prasad, Suma H and Suraksha A. Student Placement Prediction using Machine Learning, International Research Journal of Engineering and Technology (IRJIET) Volume : 06 Issue: 04 April 2019

[2]. Senthil Kumar Thangavel, Divya Bharathi P and Abhijith Shankar. Student Placement Analyzer: A recommendation System Using Machine Learning, International Conference on advanced computing and Communication systems (ICACCS-2017), Jan 06- 07,2017, Coimbatore, INDIA. [

3]. K. Sreenivasa Rao, N. Swapna, P. Praveen Kumar Educational data mining for student placement prediction using machine Learning algorithms Research Paper, International Research Journal of Engineering and Technology (IRJIET) 2018