ISSN 2278-3091
Volume 8, No.6, November – December 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse145862019.pdf
https://doi.org/10.30534/ijatcse/2019/145862019

A Business Rule for a B-School using Machine Learning

Dr. Dhimant Ganatra[1], Dr. Dinesh Nilkant[2]
[1]Data Centre and Analytics Lab, Indian Institute of Management Bangalore-560076, India, Email: dhimanth@gmail.com
[2]CMS Business School, JAIN (Deemed-to-be University), Bangalore-560009, India, Email: dineshnilkant@cms.ac.in

ABSTRACT
Providing a successful placement to its students' is at the heart of a B-School's operation. Failure to place a student has a negative impact on the brand image of the B-School. A B-School is interested in selecting the right student from among the pool of eligible students who come with different abilities and competences. The right student is the one who exhibits the potential to get placed on successful completion of the MBA program. A wrong choice of a candidate at the time of admission will weigh heavily on the B-School's ranking. This paper proposes a strategy that can be used to identify potential placeable candidates based on their past academic performance. Decision Tree algorithm is used to generate a business rule that can be used at the time of admission to minimise the chance of selecting a candidate who is not likely to get placement.

Key words: Business Rule, Classification, Decision Tree, Machine Learning.

1. INTRODUCTION

1.1 The Placement Challenge

The placement report of a B-school is the deciding factor of that school's reputation not only in the education industry but also among companies visiting campuses to hire. India's tier II and tier III business schools are on tenterhooks bracing themselves for a tougher-than-usual placement season as they remain largely uninsulated from the ongoing slowdown in the economy [1]. The challenge has compounded further as the employability of MBA graduates has fallen by 3% year-on-year according to The India Skills Report 2019 [2]. Annually around 3,60,000 students graduate from 4,000 B-Schools in India, according to data from the All India Management Association (AIMA). However, only 60 per cent of these graduates find jobs [3]. Given this scenario is it possible for a B-School to be selective when it comes to onboarding applicants into their MBA program. A B-School which can afford to have a lower acceptance rate looks for a variety of trait among the prospective students. Apart from academic ability which is normally the topmost parameter, leadership, team player and communication skills, passion and clarity on goals feature on the list. An unwritten rule is that an admission office is required to keep tab on the placeability potential of the applicant. The objective of this paper is to provide the admission office with a statistical technique backed by data to support them in their decision making. There are many machine learning algorithms which can perform this task of classifying the pool of applicants into two classes – Placeable and Not Placeable. What is required in an algorithm is its ability to lay down a business rule which the admission team can easily interpret and act accordingly. The decision tree learning algorithm perfectly fits this requirement. Trees are easily interpretable even by a management professional as they can be displayed graphically.

1.2 Machine Learning Algorithms

Machine Learning algorithms are a division of Artificial Intelligence that mimics the human learning process. Machine learning is concerned with improving the prediction accuracy as opposed to statistical learning which is focussed on validating the model's assumptions and hypothesis testing. Machine learning algorithms are largely classified into supervised, unsupervised, reinforcement and evolutionary algorithms.

A. Supervised Learning

In supervised learning, the dataset will have values of predictor variables and the corresponding response variable. Supervised learning algorithms learn from the training dataset and predict the response variable for a new record with values of input variable. Many classical statistical learning methods such as linear regression and logistic regression, as well as modern approaches such as boosting, and support vector machines fall in the domain of supervised learning. In analytics, classification problems come into play when the response variable takes a discrete value. Here, the key objective is to predict the probability of an observation belonging to a particular class. The classes may be binary or may have multiple outcomes. Classification trees, logistic regression, discriminant analysis, neural networks, and support vector machines are among the several popular techniques used for solving a classification problem. Reference [4] carried out an in-detail evaluation of more than 175 classifiers spread across 17 families over the complete UC Irvine (UCI) machine learning classification database.

B. Decision Tree

Decision Tree Learning or Classification Trees is a powerful predictive analytics technique which is used for generating business rules. It is a relatively modern technique for fitting nonlinear models. In a classification tree, we start with a root node which consists of the complete data and thereafter make use of an intelligent strategy to split the node into multiple branches thus creating children nodes [5]. Children nodes comprise of more homogenous groups. The strategy used to split a node is what differentiates among the many decision tree techniques Classification and Regression Tree is one such classification tree technique.

2. LITERATURE REVIEW

There have been many studies which have applied decision tree algorithm in an educational setup, but none have connected admission to placement for a B-school. Classifier models have been developed to understand student success in exam and course completion using academic features. There are numerous papers on predicting student placement but most of them use complex algorithms and are built on data post enrolment into a course. Reference [6] show the use of decision tree classification model for university admission system. The model was built using records having 4 attributes and the class attribute with two values: Rejected and Accepted. Decision tree algorithms are applied on engineering students' past performance data by [7] to generate the model and this model was used to predict the students' performance. The model had an accuracy of 60.46% and used 16 predictor variables to predict the performance of a student in first semester examination. Decision trees method were used by [8] to analyse the relationships between the measures of high school achievement and successful completion of students' first math and English courses in community college. Placement rules that colleges can apply directly in their placement processes are developed and validated. In [9], supervised learning techniques are used to model and select the optimal academic characteristics of students to enhance their placement probability. Finding shows that the proposed hybrid CT–ANN model achieves greater accuracy in predicting students' placement than conventional supervised learning models. Student retention is an important issue for all university policy makers due to the potential negative impact on the image of the university and the career path of the dropouts. The article [10] attempts to bring in a new perspective by exploring the issue with the use of classification trees. In [11], it was found that classification trees and random forests identified factors and complex relationships not found by other statistical methods. Classification trees were able to easily illustrate complex structures in the data that otherwise would take many interaction terms to find using traditional regression techniques. Reference [12] concluded that the decision tree algorithm can be incorporated in the Academic Environment Model to assist lecturers and management to make informed decisions about student performance. In [13], the performance of five algorithms is compared in the prediction of student's performance and results demonstrated that Simple Linear Regression gave the best prediction accuracy.

3. CLASSIFICATION AND REGRESSION TREE (CART)

CART is an umbrella term – where if the outcome variable is discrete, it is called a classification tree while if the outcome variable is continuous then it is referred to as a regression tree. In order to split a node, Classification Tree uses impurity measures such as Entropy and Gini Impurity Index. A Regression Tree splits the node that minimises the Sum of Squared Errors. The steps to generate a classification and regression tree are as below [5]:

1. We start with the root node which has the complete training data.
2. We next decide on the measure of impurity to be used. The predictor variable which minimises the impurity when the parent node is split into children nodes is chosen.
3. We need to repeat step 2 for each subset of the data using the independent variables until all of them are exhausted or the stopping criteria is met.
4. Finally, we generate a business rule for the leaf node of the tree.

The choice of measure of impurity in step 2 is between Gini Impurity Index and Entropy. Few suggested stopping criteria for step 3 are number of levels of tree from the root node or minimum number of observations in parent/child node or minimum reduction in impurity index [14].

3.1 Gini Impurity Index

It is a measure of impurity that can be used to split the node in CART. Many commercially available software tools use this measure for splitting a node. As per [5], the Gini Impurity Index at node t for a two-class problem is given as in (1).

$$GI(t) = 2 \times \text{Proportion of values in Class 1} \times \text{Proportion of values in Class 2} \qquad (1)$$

For a two-class problem, GI(t) value will be a value between 0.0 and 0.5, where a higher value indicates higher impurity. A variable which provides for the maximum reduction in Gini impurity is selected at each step to split a node t into a right and a left node as shown in Figure 1.

According to [5], a predictor variable that maximises the function in (2) is selected for splitting at node t. In simple terms, the Gini index helps decide what feature to split the tree on.

$$Max[GI(t) - P_L GI(t_L) - P_R GI(t_R)] \qquad (2)$$

3.2 Entropy

Entropy is another measure of impurity which can be used in a classification tree to split a node. The entropy at node t for a K class problem is given by (3). Entropy lies between 0.0 and 1.0, a higher value indicating higher impurity at the node.

$$Entropy(t) = -\sum_{i=1}^{K} P(C_i|t) \times log_2 P(C_i|t) \qquad (3)$$

Where $P(C_i|t)$ = Proportion of observations which belong to class C_i at node t

Apart from the two impurity measures, a decision maker may also use Cost of Misclassification to split the data. In this approach, a penalty is assigned for misclassification of positives and negatives. In cases such as credit rating, cost-based classification is preferred [15]. Figure 2 shows a comparison of the impurity measures for binary classification problems. Here p is the proportion of the records that belong to one of the two classes [16].

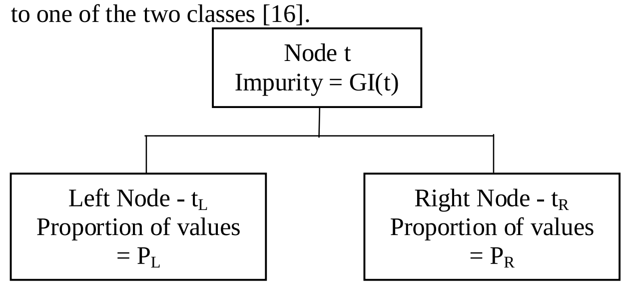

Figure 1: Splitting strategy in CART

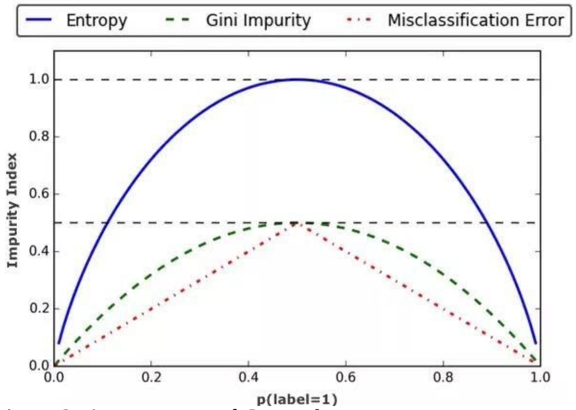

Figure 2: A comparison of Gini and Entropy impurity measures

4. METHODOLOGY

The objective here is to establish a procedure for classifying students into two-classes on the basis of scores on a set of independent variables using the classification tree algorithm. This follows the need of the B-School to differentiate among prospective students who will get placed from those who will not get placed. R language and environment (version 3.6.1) has been used for statistical computing and graphics [17].

4.1 Data Description

The dependent variable is a two-class categorical variable – Placement with labels as Placed and Not Placed. The 10 predictor variables are defined in Table 1. Classification tree is compatible with numerical as well as categorical data and the algorithm does not require that data be normalised, or dummy variables be created [18].

4.2 Sample Size

215 students who completed their MBA from a Bangalore based B-School have been selected for the study. All the 215 cases are included in the analysis as there is no missing data for any of the case. The sample size corresponds to a ratio of approximately 20 observations for each predictor variable which is considered satisfactory. The two-class sizes of 148 (Placed) and 67 (Not Placed) of the dependent variable also exceed the minimum size of 20 observations per category [19]. The dependent variable Placement has been encoded as 0 for students who are Placed and 1 for students who are Not Placed by the classifier algorithm.

Table 1: Predictor variables used for creating classification tree

Predictor Variable	Symbol	Variable Type	Description
Gender	GEN	Categorical	2 categories Male and Female
Percentage in SSC	SSC_P	Continuous	Measured in Percentage
Examination Board in SSC	SSC_B	Categorical	2 categories Central and Others
Percentage in HSC	HSC_P	Continuous	Measured in Percentage
Examination Board in HSC	HSC_B	Categorical	2 categories Central & Others
Stream in HSC	HSC_S	Categorical	3 categories - Arts, Commerce and Science
Percentage in Degree	DEG_P	Continuous	Measured in Percentage
Bachelor Degree course	DEG_T	Categorical	3 categories Comm. & Mgmt,, Science & Tech., and Others
Work Exp. after Degree	WorkEx	Categorical	2 categories Yes and No
Entrance Test Percentile	ETEST_P	Continuous	Measured in Percentage

Data description, descriptive statistics and crosstabs of the predictor variables by the outcome class are given in Table 2 for better understanding the data.

4.3 Splitting the Dataset

Validating the classification tree algorithm requires splitting the sample into two parts randomly, one used for estimation and the other for validation. While there is no hard-and-fast rule for determining the relative sizes, the 75–25 technique is used [20]. The split is as shown in Table 3.

Table 2(a): Classification table of nonmetric variables

Gender	Placed	Not Placed	Total
Male	100	39	139
Female	48	28	76
Total	148	67	215

SSC_B	Placed	Not Placed	Total
Central	78	38	116
Others	70	29	99
Total	148	67	215

HSC_B	Placed	Not Placed	Total
Central	57	27	84
Others	91	40	131
Total	148	67	215

HSC_S	Placed	Not Placed	Total
Arts	6	5	11
Comm.	79	34	113
Science	63	28	91
Total	148	67	215

DEG_T	Placed	Not Placed	Total
Comm. & Mgmt.	102	43	145
Science & Tech.	41	18	59
Others	5	6	11
Total	148	67	215

WorkEx	Placed	Not Placed	Total
Yes	64	10	74
No	84	57	141
Total	148	67	215

Table 3: Count of Train and Test set

	Placed	Not Placed	Total
Train set	111	50	161
Test set	37	17	54
Total	148	67	215

Table 2(b): Descriptive statistics of metric variables

Variable	Mean		Mean Difference	Sig. (2 tailed)	Std. Deviation		Skewness		Kurtosis	
	Placed	Not Placed			Placed	Not Placed	Statistic	Std. Error	Statistic	Std. Error
SSC_P	71.72	57.54	14.18	0.000	8.71	8.39	-.133	.166	-.608	.330
HSC_P	69.92	58.39	11.53	0.000	9.32	9.91	.164	.166	.451	.330
DEG_P	68.74	61.13	7.61	0.000	6.51	6.36	.245	.166	.052	.330
ETEST_P	73.24	69.58	3.66	0.049	13.69	11.99	.285	.166	-1.087	.330

5. CLASSIFIER MODEL

5.1 Decision Tree Classifier using Gini Criteria

Recursive Partitioning And Regression Trees (rpart) library [21] provides the algorithm to create the decision tree. rpart uses the formula interface but does not take interactions. Figure 3 shows the classification tree for the train dataset. Nodes that split to the left are the ones which meet the criteria while nodes to the right do not. Each node is labelled by the predicted class, either Placed or as Not Placed. The percentage value is to be read from left to right, with the probability of Placed being on the left.

The pre-pruning approach which involves defining an early stopping rule avoids generating overly complex subtrees. Here the algorithm is halted before generating a fully-grown tree. For the tree shown in Figure 3, a stopping criterion of 4 levels from the root node is used by defining the maxdepth. The tree so obtained is as displayed in Figure 4. At the expense of a little bias, it is preferred to have a smaller tree with lesser splits for a better interpretation and a lower variance [22].

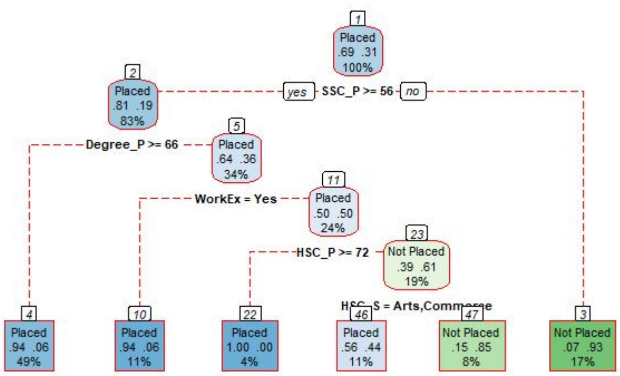

Figure 3: Classification tree for train dataset

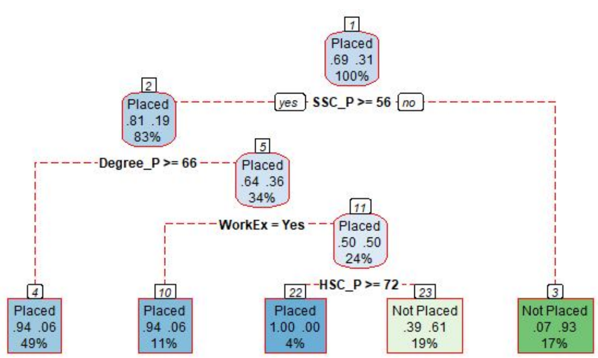

Figure 4: Classification tree for train dataset with 4 levels

From the classification tree of Figure 4, it can be interpreted that

a. In the first node which is the root node for all the data, there are 161 observations of which 111 are Placed and 50 are Not Placed cases. The Gini index at the node is 0.4278.
b. SSC_P is the most important variable for splitting the dataset as compared to other features, meaning that, the feature SSC_P provides the maximum reduction in Gini impurity.
c. At one of the leaf node 4, there are 79 observations of which 74 are Placed class and 5 are Not Placed class. The corresponding Gini index is 0.1128.

While stopping criteria are a relatively crude method of stopping tree growth, an alternative approach to stopping growth is to allow the tree to grow and then prune it back to an optimum size in a bottoms-up fashion [5]. The complexity parameter (cp) is used to control tree growth. If the cost of adding a variable is higher than the value of cp, then tree growth stops. The idea here is to allow the decision tree to grow fully and observe the cp value. Next, we prune/cut the tree with the optimal cp value as the parameter. Table 4 shows the cross-validation error for each nsplit which can be used to prune the tree. The optimal value of cp is the one with the least cross-validated error (xerror). Figure 5 shows the cp plot. Based on practical considerations in the given scenario so as to generate business rules, a deviation is made from the optimal cp value and a depth of 4 is considered. The resulting classification tree is as shown in Figure 4 earlier.

Table 4: Cross-validated error for each split

	cp	nsplit	rel error	xerror	xstd
1	0.460000	0	1.00	1.00	0.117426
2	0.046667	1	0.54	0.58	0.097522
3	0.040000	4	0.40	0.64	0.101271
4	0.010000	5	0.36	0.60	0.098813

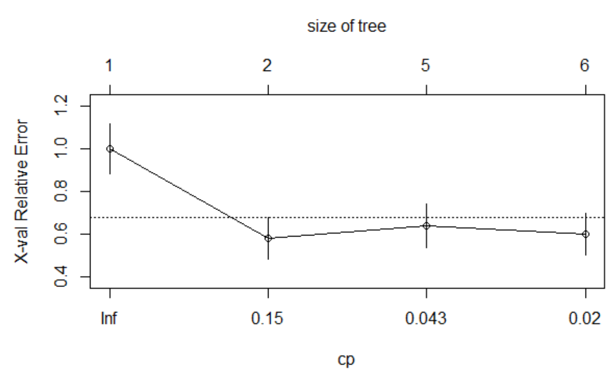

Figure 5: cp plot

The biggest advantage of using a classification tree is its ability to generate business rules that can be deployed for decision making. In Figure 4, nodes 3, 4, 10, 22, and 23 are the terminal nodes (leaf) as there are no branches emanating out of them. Table 5 lists the business rule relating to a leaf node and the corresponding support which indicates the proportion of data in that node. For decision-making, a business rule that has high accuracy as well as high support is desired [7].

From Figure 4 and Table 5, at node 4 of the tree we understand that if a student has scored more than 56% in SSC and more than 66% in Bachelor Degree, then there is more than 94% chance that the student is likely to be Placed. Apart from this class prediction, we are also interested in knowing the class proportions among the training observations that fall into that region. The support at node 4 is 49%.

5.2 Evaluating the performance of the classifier

A. Classification Table

After the model has been built, it can be applied to the test dataset to predict the class labels of previously unseen records. Such measure provides an unbiased estimate of its generalisation error [16]. Table 6 shows the classification table (confusion matrix) for the tree in Figure 4 applied to the test dataset. The confusion matrix for a binary classifier model is a table that evaluates the performance of the classification tree model showing the accuracy of classifying positives and negatives.

The accuracy of classifying Placed (negative) is 83.78%, whereas the accuracy of classifying Not Placed (positive) is 88.24%. The overall accuracy, which measures how often the classifier makes correct prediction is 85.19%. The model's performance cannot be decided based on overall accuracy particularly when the data is imbalanced. As per the accuracy paradox, a model with higher overall accuracy may not be a better model. Given the context, here, we need a higher accuracy in predicting positive classes (Not Placed, $Y_i = 1$) rather than negative classes (Placed, $Y_i = 0$) which is the case in Table 6.

Table 5: Business Rule and Support at leaf nodes

Node	Business Rule	Support
3	If SSC_P < 56, classify as Not Placed. Accuracy is 93%	17%
4	If SSC_P ≥ 56 AND Degree_P ≥ 66, classify as Placed. Accuracy is 94%	49%
10	If SSC_P ≥ 56 AND Degree_P < 66 AND WorkEx = Yes, classify as Placed. Accuracy is 94%	11%
22	If SSC_P ≥ 56 AND Degree_P < 66 AND WorkEx = No AND HSC_P ≥ 72, classify as Placed. Accuracy is 100%	4%
23	If SSC_P ≥ 56 AND Degree_P < 66 AND WorkEx = No AND HSC_P < 72, classify as Not Placed. Accuracy is 61%	19%

Table 6: Confusion Matrix based on Gini impurity

Actual	Predicted		Percentage Correct
	Placed	**Not Placed**	
Placed	31	6	83.78
Not Placed	2	15	88.24
Overall Percentage			85.19

We next compare the accuracy of the model for the test sample to the accuracy of the model for the train sample. The classification accuracy rate for the model using the training sample was 87.58%, compared to 85.19% for the test sample. As a rule, if classification accuracy of the test sample is within 10% of the training sample, this provides evidence of the utility of the model [23]. This supports a conclusion that the classification tree model based on the training sample would be effective in predicting scores for cases other than those included in the sample.

B. Sensitivity, Specificity, and Precision

The performance of the classification model is often measured using the four counts in the confusion matrix. Known as class statistics, they summarise the model performance for the positives and negatives separately [15]. Sensitivity and Specificity refers to the ability of a model to correctly classify positives and negatives respectively. In this case, sensitivity also known as Recall, measures how many of the actual Not Placed students are correctly predicted as Not Placed. From Table 6, sensitivity is 88.24%, meaning about 88% of the Not Placed students in the test dataset were correctly predicted as Not Placed. Specificity is the ability of the model to correctly classify the negatives, that is, when the actual value is negative, how often is the prediction correct. From Table 6, specificity is 83.78%, meaning less than 17% of all Placed students are predicted incorrectly as Not Placed. Precision measures how good the model is at assigning positives to the positive class. For our classifier model, the precision is 71.43%, meaning, almost 72% of the students predicted as Not Placed actually belonged to the Not Placed class.

C. F-Score

The F-Measure combines both precision and recall and is their harmonic mean [24]. The F-Score reaches the best value at 1 and worst score at 0. Where the cost of false positives and false negatives are very different, the F-score is superior as compared to the overall accuracy. In our case the F-Score is 0.7895.

D. Receiver Operating Characteristic curve (ROC curve)

ROC curve is used in order to understand the overall worth of a classification tree [25]. The ROC curve for the placement test dataset is shown in Figure 6.

In Figure 6, the area below the diagonal line which represents the case of not using a model is 0.5. The red line which is above the diagonal, captures sensitivity and 1 – specificity. The area under the ROC curve (AUC) is 0.893, indicating the proportion of concordance pairs in the data. Models with higher AUC are preferred. A good rule of thumb is that AUC of at least 0.7 is required for practical application of a model [18].

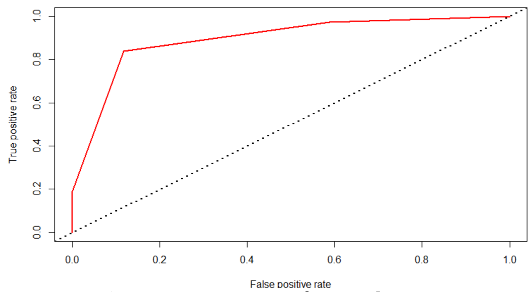

Figure 6: ROC Curve for test dataset

5.3 Decision Tree Classifier using Entropy Criteria

Figure 7 shows the classification tree for the train dataset and Table 7 shows the confusion matrix for the model applied to the test dataset. The model performs poorly on sensitivity as compared to the Gini impurity.

5.4 Cost Based Splitting Criteria

Cost of Misclassification criteria can also be used to split the data. In case of Placement, the cost of misclassifying a non-placeable student as placeable is way higher as compared to the cost of loosing out on a placeable candidate since (s)he got misclassified as non-placeable by the model. In the given context, we assume that there is a penalty of 1 for misclassifying $Y = 0$ (negative) as $Y = 1$ (positive) and a penalty of 5 for misclassifying $Y = 1$ (positive) as $Y = 0$ (negative). The Table 8 shows the penalty for misclassification.

Figure 8 shows the classification tree using penalty cost minimisation. The trees in Figure 8 and Figure 1 are different from level 1 itself. Table 9 shows the confusion matrix based on penalty cost classification. Though there is a reduction in the overall model accuracy as compared to Table 6 which is based on the Gini criteria, the accuracy of classifying positives (sensitivity) has increased to 94.12% compared to 88.24% earlier. This kind of trade-off between sensitivity and specificity can be seen in most classification problems [15].

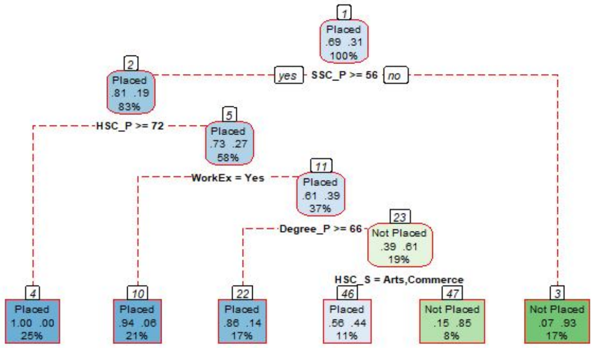

Figure 7: Classification tree based on Entropy

Table 7: Confusion Matrix based on Entropy

Actual	Predicted		Percentage Correct
	Placed	Not Placed	
Placed	34	3	91.89
Not Placed	6	11	64.71
Overall Percentage			83.33

Table 8: Penalty for Misclassification

Actual	Predicted	
	Placed	Not Placed
Placed	0	1
Not Placed	5	0

6. CONCLUSION

With increased power of computing infrastructure, the necessary simplifying assumptions of linearity and normality are staring to give way to nonparametric techniques. While trees are easy to interpret and fit the data nicely, they suffer from high variance. With a small change in the training data, the results that we get could be significantly different in the model [26].

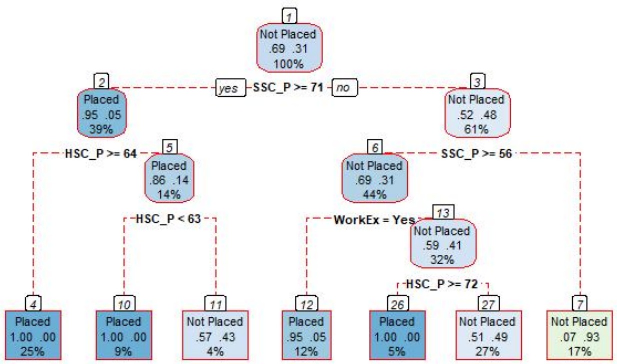

Figure 8: Classification tree based on penalty cost minimisation

Table 9: Confusion Matrix based on Cost of Misclassification

Actual	Predicted		Percentage Correct
	Placed	Not Placed	
Placed	26	11	70.27
Not Placed	1	16	94.12
Overall Percentage			77.77

Nevertheless, since they help create business strategies which other classification algorithms cannot coupled with the added advantage that trees can be visualized, they are best suited in the given context.

With an 85.19% overall accuracy, the classifier can serve as a reliable mechanism and aid the admission office in decision making. As indicated earlier, for a B-School it is imperative that it is able to differentiate the applicants into Placeable and Non-placeable classes. The cost-based splitting criteria with a high sensitivity of 94.12% adds strength by being able to avoid misclassification of positives which in this case means wrongly identifying a non-placeable applicant as placeable.

A graphical representation of the business rule which the admission office can follow is shown in Figure 9. This rule cannot be a sole criterion based on which an applicant is offered or denied admission into the program but can aid in the decision-making process along with other institute specific criteria.

Future work: Unfortunately, classification trees generally do not have the same level of predictive accuracy as other classification and regression approaches [22]. The predictive performance of trees can be substantially improved by aggregating many decision trees, using methods like bagging, random forests, and boosting [27]. There is scope to evaluate the increase in model accuracy using these methods.

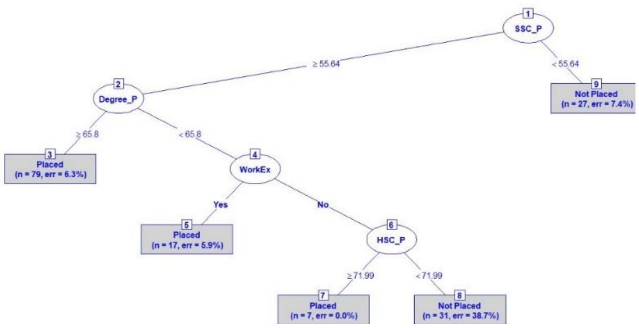

Figure 9: Business rule for the Placement dataset

REFERENCES

1. Economic Times (2019, November 02). **Slowdown has tier II, III B-schools on tenterhooks**. https://economictimes.indiatimes.com/jobs/slowdown-has-tier-ii-iii-b-schools-on-tenterhooks/articleshow/71860868.cms?from=mdr
2. United Nations Development Programme (2019, May 29). **The India Skills Report 2019**. https://www.undp.org/content/dam/india/docs/poverty/India-Skills-Report-2019.pdf
3. India Today (2019, October 26). **Step by step to the top**. https://www.indiatoday.in/magazine/education/story/20191104-step-by-step-to-the-top-1612697-2019-10-26
4. M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, **Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?**. *Journal of Machine Learning Research*, vol. 15(1), pp. 3133-3181, 2014.
5. L Breiman, JH Friedman, RA Olshen, and CJ Stone. **Classification and Regression Trees**. Chapman and Hall, 1984.
6. A. F. Mashat, M. M. Fouad, P. S. Yu, and T. F. Gharib. **A Decision Tree Classification Model for University Admission System**. *International Journal of Advanced Computer Science and Applications*, vol. 3(10), pp. 17-21, 2012.
7. R. R. Kabra, and R. S. Bichkar. **Performance Prediction of Engineering Students using Decision Trees**. *International Journal of Computer Applications*, vol. 36(11), pp. 8-12, 2011.
8. P. R. Bahr et al. **Improving Placement Accuracy in California's Community Colleges Using Multiple Measures of High School Achievement**. *Sage Journal*, vol. 47(2), pp. 178-211, 2019.
9. T. Chakraborty, S. Chattopadhyay, and A. K. Chakraborty. **A novel hybridization of classification trees and artificial neural networks for selection of students in a business school**. *OPSEARCH*, Springer, vol. 55(2), pp. 434-446, 2018. https://doi.org/10.1007/s12597-017-0329-2
10. C. H. Yu, S. DiGangi, A. Jannasch-Pennell and C. Kaprolet. **A Data Mining Approach for Identifying Predictors of Student Retention from Sophomore to Junior Year**. *Journal of Data Science*, vol. 8, pp. 307-325, 2010.
11. G. Mendez, T. Buskirk, S. Lohr, and S. Haag. **Factors associated with persistence in Science and Engineering majors: An exploratory study using classification trees and random forests**. *Journal of Engineering Education*, vol. 97(1), 2013.
12. L. H. Musawenkosi, and T. Bhekisipho, T. **Development of the Academic Model to Predict Student Success at VUT-FSASEC Using Decision Trees**. *International Journal of Computer and Information Engineering*, vol. 11(11), pp. 1188-1191, 2017.
13. A. Dhankhar, K. Solanki, A. Rathee, and Ashish. **Predicting Student's Performance by using Classification Methods**. *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8(4), pp. 1532-1536, 2019. https://doi.org/10.30534/ijatcse/2019/75842019
14. L. Rokach, O. Maimon, **Data Mining and Knowledge Discovery Handbook**. Boston, MA: Springer, 2005, pp. 165-192.
15. U. D. Kumar, **Business Analytics- The Science of Data Driven Decision Making**. New Delhi: Wiley India, 2017, ch.. 12.
16. P-N. Tan, M. Steinbach, and V. Kumar. **Introduction to Data Mining**. Pearson Education India, 2016, ch. 4.
17. **The R project for statistical computing**. https://www.r-project.org/
18. M Pradhan, U. D. Kumar, **Machine Learning using Python**. New Delhi: Wiley India, 2019, pp. 152-153.
19. D.R. Anderson, D.J. Sweeney, and T.A. Williams. **Statistics for Business and Economics**, 9th ed. New Delhi: Thomson South-Western, 2007.
20. J.E. Holden, W. H. Finch, and K. Kelley. **A Comparison of Two-Group Classification Methods, Educational and Psychological Measurement**. 71(5). Sage, 2011, pp. 870–901.
21. **Package rpart**. https://cran.r-project.org/web/packages/rpart/rpart.pdf
22. G. James, D. Witten, T. Hastie and R. Tibshirani. **An Introduction to Statistical Learning with Applications in R**. Springer, 2017, pp. 307-308.
23. G. Forman, and M. Scholz. **Apples to apples in cross-validation studies: Pitfalls in classifier performance measurement**. *ACM SIGKDD Explorations*, vol.12(1), pp. 49–57, 2010.
24. M. Sokolova, N. Japkowicz, and S. Szpakowicz. **Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation**. Australasian Joint Conference on Artificial Intelligence, Springer, 2006, pp. 1015-1021.
25. J. A. Hanley, and B. J. McNeil. **The meaning and use of the area under a receiver operating characteristic (ROC) curve**. *Radiology*, vol. 143(1), pp. 29- 36, 1982. https://pubs.rsna.org/doi/pdf/10.1148/radiology.143.1.7063747.
26. J. P. Lander. **R for everyone – Advanced Analytics and Graphics**. Pearson Education, Inc, 2016, pp. 310-312.
27. L. Breiman. **Random Forests**. *Machine Learning*, vol. 45 (1), Springer, pp. 5–32, 2001. https://doi.org/10.1023/A:1010933404324