CHAPTER 5

PREDICTION OF STUDENT POTENTIAL IN TECHNICAL PROGRAMS

5.1 Introduction

In this chapter, we show how to predict a student's potential for study in technical programs, using various ML classification methods and to evaluate the performance of these methods, based on their predictive accuracy and other evaluation measures. The classification techniques will use the different feature subsets generated by feature selection algorithms discussed in chapter 4.

The main objective is to assign a student to a predefined class, with minimum error rate. The most appropriate class is assigned to the student by the system based on his/her attributes.

The dataset used in this chapter has two parts: the student dataset for analysis of student potential, without considering the Education Boards and the dataset used for analysis of student potential based on the two Boards of Secondary Education, Rajasthan and Central. All the 21 attributes were used to determine the best classifier model.

As mentioned in Chapter 4, this research has proposed a framework to recommend classifier models for prediction of student potential. There are many classification algorithms from different areas in statistics and ML. So, it is very difficult to select the best algorithm. A brute force approach has been applied for this problem- to try all the classification methods on the student datasets and choose the one with the best result.

Brachman and Anand (1996) have concluded that the choice of the selection of algorithms is an exploratory process, dependent on the user's knowledge about the problem domain and the algorithms. Classification accuracy is the most important evaluation metric which is used to check the goodness of a classifier, but other metrics/criterion can also be used to evaluate the performance of prediction models.

Additionally, this chapter will discuss the data used in this study, the required tools and the results of applying ML techniques on the student dataset.

5.2 Applying the ML techniques to analyse student potential

The best classifier model is selected based on the values of the selected metrics of performance. This study has used five metrics: accuracy, precision, sensitivity, specificity and F-measure. The measures were calculated using a 10-fold cross validation. For performance evaluation, the student data was divided into two parts: 70% of data was used as training data and 30% data was used as testing data.

In the Literature review, it was shown that many ML techniques are used on a specific dataset for different tasks. In this study, different ML techniques were applied on the same student dataset and compared with each other to find out which has the highest predictive accuracy.

The ML techniques used in this research are: K-Nearest Neighbor, Naïve Bayes, Decision Tree, Support Vector Machine, Random Forest and Stochastic Gradient Boosting. The prediction model used the dataset and provided the evaluation results. Finally, the confusion matrix was made to show whether the data is correctly classified or not. All the models were built on the Jupyter Notebook, which provides a good ML environment.

Some specified Machine Learning models need information in a specified format. Dataset should be formatted in such a way that more than one Machine Learning algorithms are executed on any one data set and the best out of them is selected.

Both training dataset and test dataset are One-hot encoded, i.e., nominal variables are converted into numerical form to be provided to different Machine Learning algorithms for effective prediction. These classification algorithms are executed in python with 10-fold cross-validation. Tool python3 was used to run the different Machine Learning algorithms. matplotlib library was used to visualize the inner working of the model. The accuracy measure was used for evaluating the quality of the classifier. The purpose of accuracy is to achieve a higher value.

In the next section we discuss each technique separately.

K-Nearest Neighbor

The first model to be built was kNN. It is a classification method that has been used in education data mining (Baker & Inventado, 2014). KNN was applied on the student

data for analysis of student potential based on the attributes which were identified at the time of data collection. In KNN algorithm, we store all available cases and categorize new cases by a majority vote of its k neighbors. The case is assigned to the class which is most common amongst its K nearest neighbors. It measured by a distance function.

The distance functions which are used for assigning the case can be Euclidean, Manhattan, Minkowski or Hamming distance. The first three functions are used for continuous function and the last one (Hamming) for categorical variables. If K = 1, then the case is only allocated to the class of its nearest neighbor; at times, when we perform kNN modelling, selection of K is a challenging task.

The KNN model performance details are shown in Table 5.1 and 5.2. Table 5.1 shows the confusion matrix calculated by KNN model on the student dataset.

	Training	Test
True Positive (TP)	427.000000	205.000000
False Positive (FP)	205.000000	88.000000
False Negative (FN)	135.000000	42.000000
True Negative (TN)	574.00000	240.00000

Table 5.1 KNN Confusion Matrix

Metric	Training Dataset	Testing Dataset
Accuracy	0.746458	0.773913
Precision	0.675633	0.699659
Specificity	0.736842	0.731707
Sensitivity	0.759786	0.829960
F-measure	0.715243	0.759259

Table 5.2 KNN result evaluation for analysis of student potential

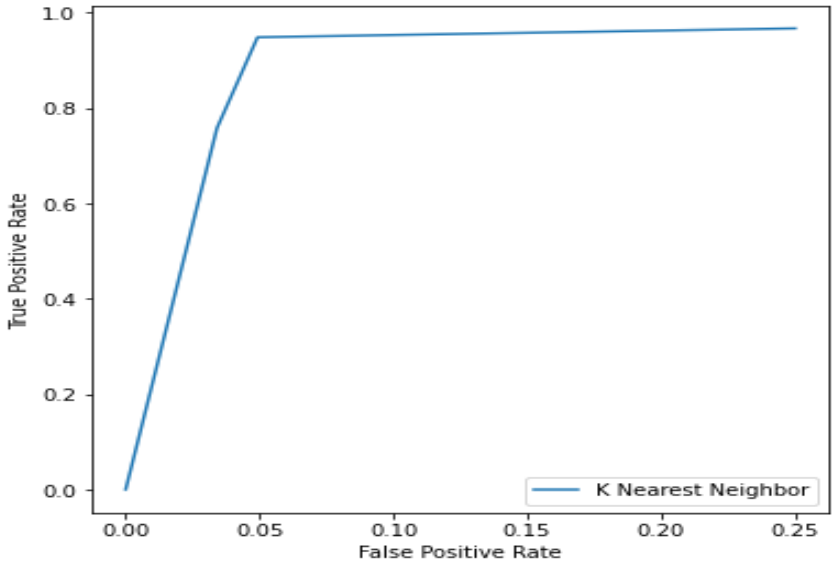

Figure 5.1 KNN ROC Curve

The result is shown in Table 5.2. Figure 5.1 shows the KNN ROC curve. The area of the curve is determined by the shape of each curve. A curve which shows a nearly 45-degree diagonal line means the model makes random guesses, while a rapidly increasing curve shows that there are more true positives and less false positives. The model which creates a bigger area under the curve results in better classifier performance.

Support Vector Machine

Support Vector Machine is a supervised machine learning algorithm for classification problems and has been used in education data mining. The algorithm plots each data item as a point in n-dimensional space (where n is number of features) with the value of each feature being the value of a particular coordinate. Details of the SVM model performance are shown in Tables 5.3 and 5.4.

	Training	**Test**
True Positive (TP)	427	205
False Positive (FP)	205	88
False Negative (FN)	135	42
True Negative (TN)	574	240

Table 5.3 SVM Confusion Matrix

The confusion matrix is evaluated by applying the SVM model on the student dataset, as shown in table 5.3. This method is simple and fast for solving large problems. Figure 5.2 shows the SVM ROC curve.

Metric	Training Dataset	Testing Dataset
Accuracy	0.74	0.76
Precision	0.689873	0.682594
Specificity	0.741425	0.719033
Sensitivity	0.747856	0.819672
F-measure	0.717695	0.744879

Table 5.4 SVM result evaluation for analysis of student potential

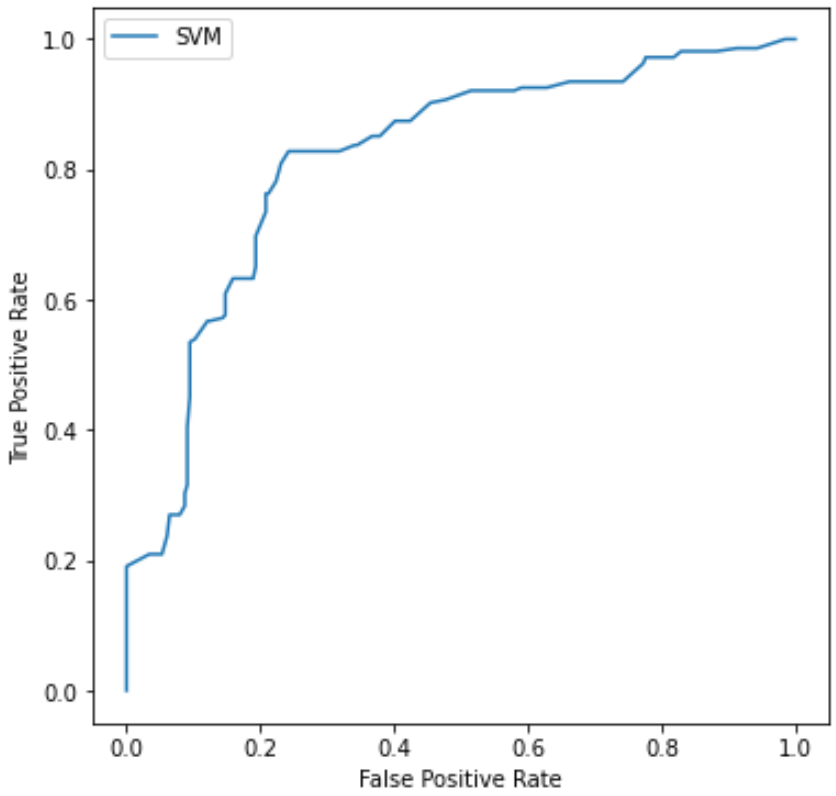

Figure 5.2 SVM ROC Curve

Naïve Bayes

Naive Bayes is a classification technique based on Bayes' theorem. It is an assumption of independence between predictors. In simple term, a Naive Bayes classifier predicts that the presence of a particular feature in a class is unrelated to the presence of any other feature. The Naïve Bayes model performance details are shown in Tables 5.5 and 5.6. Figure 4.3 shows its ROC curve.

	Training	Test
True Positive (TP)	420.000000	186.000000
False Positive (FP)	212.000000	107.000000
False Negative (FN)	181.000000	56.000000
True Negative (TN)	528.00000	226.00000

Table 5.5 Naïve Bayes Confusion Matrix

Metric	Training Dataset	Testing Dataset
Accuracy	0.716522	0.716522
Precision	0.664557	0.634812
Specificity	0.713514	0.678679
Sensitivity	0.698835	0.768595
F-measure	0.681265	0.695327

Table 5.6 Naïve Bayes result evaluation

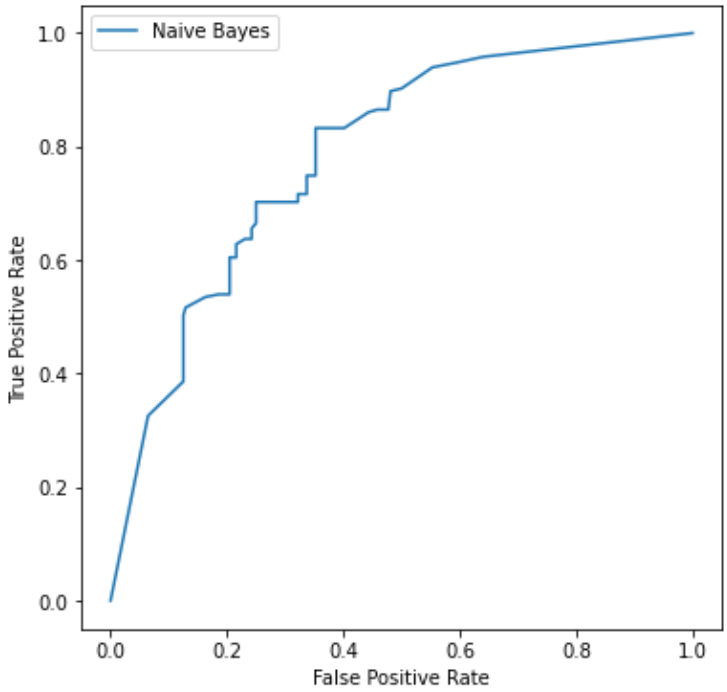

Figure 5.3 Naïve Bayes ROC Curve

Decision Tree

Decision tree is a supervised learning algorithm which is mostly used for classification problems. Decision tree is used for both categorical and continuous dependent variables. In this algorithm, we divide the population into two or more similar sets. In

a decision tree, a test on the attribute is represented by an internal node, each branch of the tree represents the outcome of the test and the leaf node represents a particular class label. The results of implementing Decision tree are presented in Tables 5.7 and 5.8.

	Training	**Test**
True Positive (TP)	528.000000	242.000000
False Positive (FP)	104.000000	51.000000
False Negative (FN)	115.000000	39.000000
True Negative (TN)	594.00000	243.00000

Table 5.7 Decision Tree Confusion Matrix

Metric	**Training Dataset**	**Testing Dataset**
Accuracy	0.836689	0.843478
Precision	0.835443	0.825939
Specificity	0.851003	0.826531
Sensitivity	0.821151	0.861210
F-measure	0.828235	0.843206

Table 5.8 Decision Tree result evaluation

It shows that the accuracy achieved by training dataset is 83% and testing dataset is 84%. The ROC curve by this model is shown in Figure 5.4.

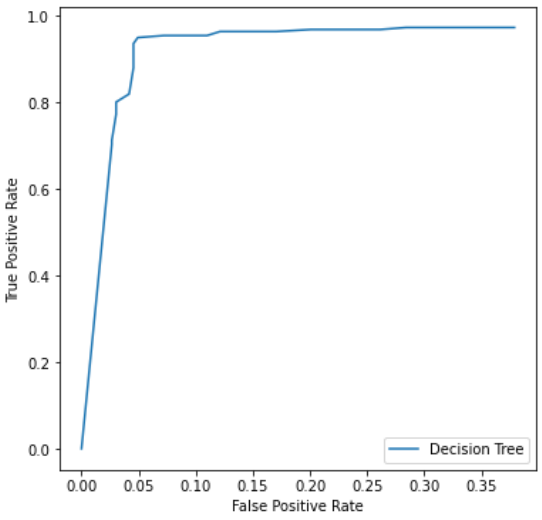

Figure 5.4 Decision Tree ROC Curve

Random Forest

Random Forest is an ensemble technique in which a random forest is created by a large dataset divided into several tree predictors. Each tree predictor works independently on

different samples and the results are combined to improve the performance of the classifier model. It follows the bagging procedure (also called bootstrapping) with some improvements. In this procedure, samples are taken from the training data repeatedly.

The confusion matrix which is evaluated by this model is shown in Table 5.9. Table 5.10 shows the result and Figure 5.5 shows the ROC curve.

	Training	**Test**
True Positive (TP)	520.000000	229.000000
False Positive (FP)	112.000000	64.000000
False Negative (FN)	119.000000	41.000000
True Negative (TN)	590.00000	241.00000

Table 5.9 Random Forest Confusion Matrix

Metric	**Training Dataset**	**Testing Dataset**
Accuracy	0.82774	0.817391
Precision	0.822785	0.781570
Specificity	0.840456	0.790164
Sensitivity	0.813772	0.848148
F-measure	0.818253	0.813499

Table 5.10 Random Forest result evaluation

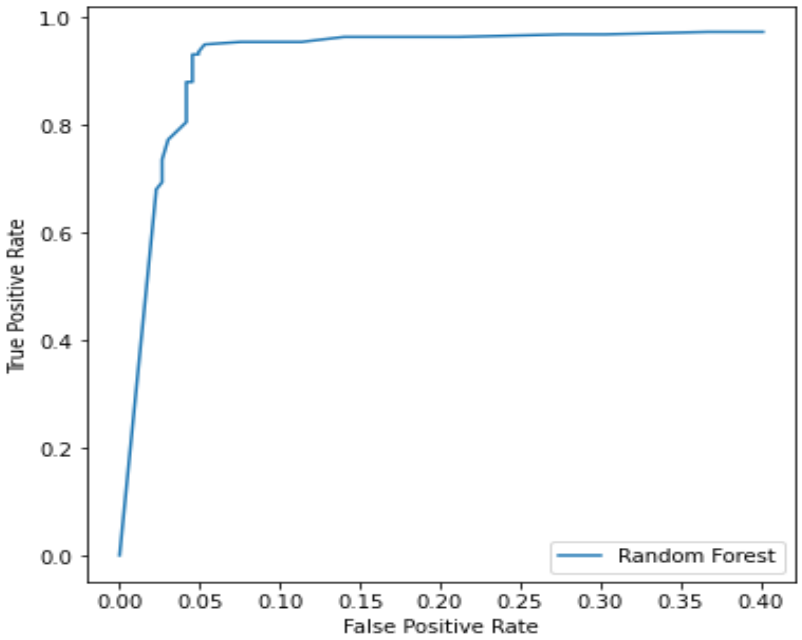

Figure 5.5 Random Forest ROC Curve

Stochastic Gradient Boosting

Stochastic Gradient Boosting (GB) is a boosting ensemble method. It is usually used for improving the performance of any classifier and reducing the error of the weak one. It is a successive process, where each succeeding model attempts to correct the errors of the previous one. GB was applied on the student data for analysis of student potential based on the attributes which were identified at the time of data collection.

The GB model performance details are given in Tables 5.11 and 5.12 and its ROC curve is shown in Figure 5.6.

	Training	**Test**
True Positive (TP)	590.000000	259.000000
False Positive (FP)	42.000000	34.000000
False Negative (FN)	40.000000	16.000000
True Negative (TN)	669.00000	266.00000

Table 5.11 GB Confusion Matrix

Metric	**Training Dataset**	**Testing Dataset**
Accuracy	0.938852	0.913043
Precision	0.933544	0.883959
Specificity	0.940928	0.886667
Sensitivity	0.936508	0.941818
F-measure	0.935024	0.911972

Table 5.12 GB result evaluation for analysis of student potential

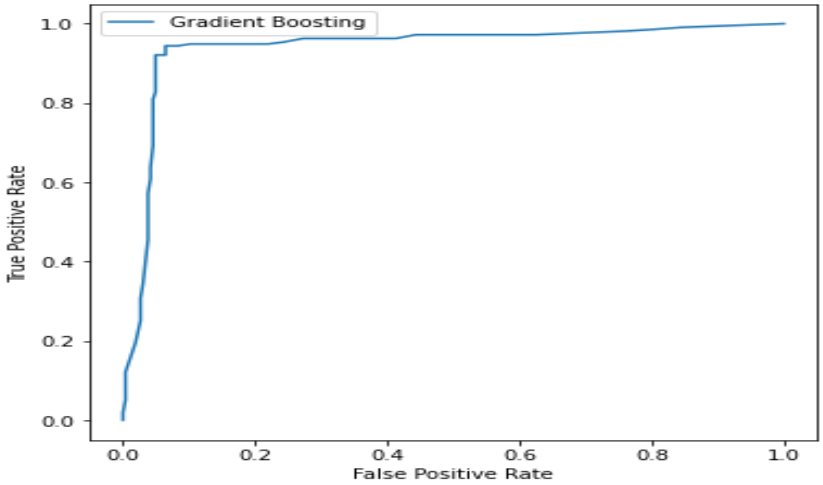

Figure 5.6 Gradient Boosting ROC Curve

5.3 Evaluation of the classifier results

In this subsection, we compare the results of experiments used to find the best classifier model for the analysis of student potential. Table 5.13 shows the correctly and incorrectly classified actual number of student records.

ML Prediction Models	Correctly Classified	Incorrectly Classified
K-Nearest Neighbor	1446	470
Decision Tree	1607	309
Random Forest	1580	336
Gradient Boosting	1784	132
Naïve Bayes	1360	556
Support Vector Machine	1532	357

Table 5.13 Correctly and Incorrectly Classified records

This table shows the comparison of the different classifier models performance in terms of actual number of records that were classified correctly or incorrectly. As shown, Gradient Boosting has the highest correctly classified number of records and also lowest misclassified records. Decision Tree had the second highest correctly classified. The higher the number of correctly classified records, the more suitable is the classifier for the used dataset. So, Gradient Boosting is the most suitable classifier for the used dataset in this study.

Analysis of the results based on the selected metrics is explained next.

ML Prediction Models	Accuracy	f1 score	Precision	Sensitivity	Specificity
K-Nearest Neighbor	0.7464	0.7152	0.6756	0.7597	0.7368
Decision Tree	0.8366	0.8282	0.8354	0.8211	0.8510
Random Forest	0.8277	0.8182	0.8227	0.8137	0.8404
Gradient Boosting	0.9388	0.9350	0.9335	0.9365	0.9409
Naïve Bayes	0.7069	0.6812	0.6645	0.6988	0.7135
Support Vector Machine	0.74	0.7176	0.6898	0.7478	0.7414

Table 5.14 Comparison based on training dataset

ML Prediction Models	Accuracy	f1 score	Precision	Sensitivity	Specificity
K-Nearest Neighbor	0.7739	0.7592	0.6996	0.8299	0.7317
Decision Tree	0.8434	0.8432	0.8354	0.8612	0.8265
Random Forest	0.8173	0.8182	0.8259	0.8137	0.8182
Gradient Boosting	0.913	0.9119	0.8839	0.9418	0.8866
Naïve Bayes	0.7165	0.6953	0.6348	0.7685	0.6786
Support Vector Machine	0.7617	0.7444	0.6825	0.8196	0.719

Table 5.15 Comparison based on testing dataset

The results of classifiers' performance, based on five metrics, are given in Table 5.15. The first metric is accuracy. It is the total number of correct classifications out of the total number of classifications done. The result show that a majority of the classifiers, three out of six, obtain an accuracy of over 80%. Gradient Boosting achieved the highest accuracy while Naïve Bayes classifier has the lowest accuracy.

The next important measure is f-measure. It is used to determine the performance of the different classes separately. It is the harmonic average of the recall and precision rates only for the technical class. The values of the classifiers, except Naïve Bayes, were reasonable. Gradient Boosting gave the highest value of 91.1% and NB gave the lowest value of 63.48%.

Specificity, the proportion of negatives that are correctly classified, ranges from 88% to 66%. Half of the models had probability of specificity below 70%. This is basically attributed to imbalanced class.

Another important metric is the recall value, also called sensitivity; it is the proportion of the true positives which are correctly classified. The result showed that the majority of the models are highly sensitive, with five out of six models obtaining a recall probability of over 80%. Gradient Boosting again achieved the highest recall probability of 88%, while NB has the lowest probability.

The precision metric measures the positive predictive value and ranges from 88% to 63%. Half of the classifiers had precision probability below 70%. Gradient Boosting has the highest precision value and NB has lowest value.

The ROC curve area is a reliable measure of classifier performance, remaining stable even for imbalance classes (Brown and Devis, 2006). The ROC values remain constant for the classes in each model as seen in all the six figures. The highest performance was achieved by Gradient Boosting and lowest by NB, as they have the highest and lowest areas under the curve.

The results thus show that out of the six classifiers, Gradient Boosting achieved the best metric value, making it the most appropriate classifier for the used data in this research.

The bar graph shown in Figure 5.8 also summarizes the six classifier models performance based on the five metrics. It also shows that the performance of Gradient Boosting is best.

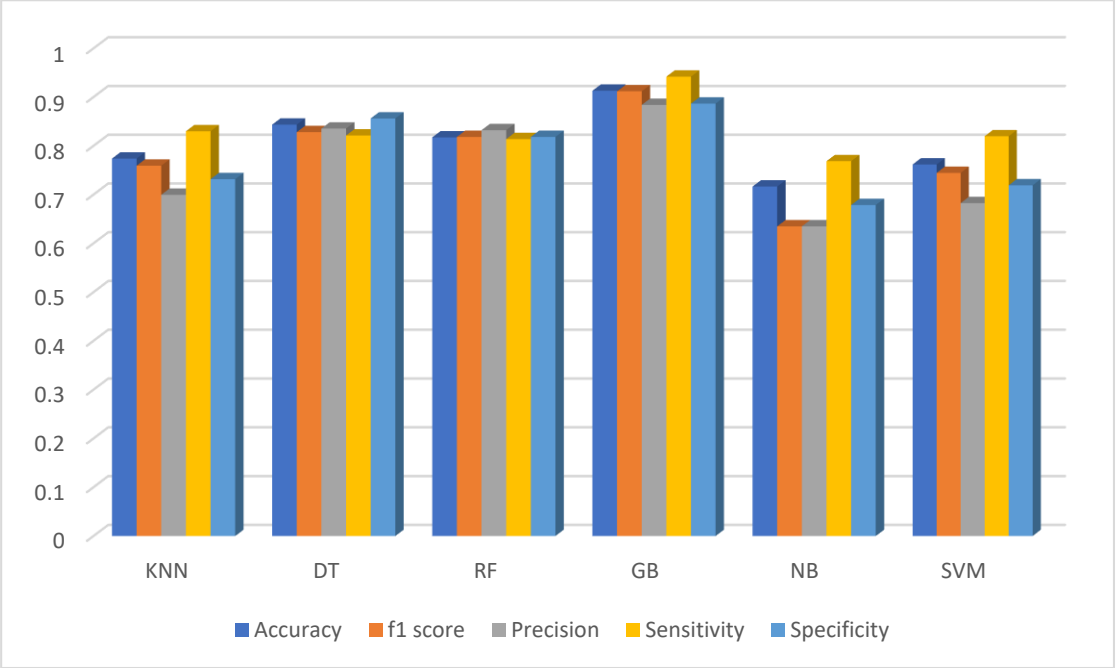

Figure 5.7 Bar Graph for Comparison Analysis of Classifier Methods

5.4 Finding the optimal feature subset from the student dataset

Feature selection is a process that is used in Machine Learning, wherein a subset of the factors is selected from the data for better prediction. This is an important step of pre-processing for avoiding the curse of dimensionality. This method tries to give a subset of features that are relevant to the target. This helps in getting a better prediction result for analysis of student potential and performance.

The filter method relies on the general features of the training data to select some attributes independently of any algorithm. Filter does not require re-execution for different algorithms. So, this research uses only filter algorithms. As mentioned in the

previous Chapter, we have applied four filter methods, Mutual Information Gain (MIG), Univariate (ANOVA) Test, Univariate ROC_AUC Test and Fisher Score & Chi2 Test. We discuss the results of this in the following sections.

Mutual Information Gain

The features ranked by MIG are shown in Table 5.16. The experiments were conducted in python3. The features order from most important to least important is shown in the second column.

The MIG in the Table showed that *StIntforJob* is the best attribute of the target class, followed by *Stream*, and *10th*, the last attribute is *MoE*. This rank was obtained through the 10-fold cross validation.

Original Dataset	**Dataset ranking by MIG**
0. Age	13. StIntforJob
1. Gender	11. Stream
2. Fedu	10. 10th
3. Foccu	19.WhythisCourse
4. Medu	6. Fincome
5. Moccu	2. Fedu
6. Fincome	12. Twelth
7. NoS	15. ST
8. Address	18. AreaofInt
9. MoE	3. Foccu
10. 10th	4. Medu
11. Stream	0. Age
12. Twelth	17.IntInfluence
13. StIntforJob	14. Health
14. Health	16. STS
15. ST	7. NoS
16. STS	5. Moccu
17.IntInfluence	8. Address
18. AreaofInt	1. Gender
19.Why this Course	9. MoE

Table 5.16 Features ranked by MIG

The elimination process aims to reduce the size of the input feature set and at the same time to retain the class discriminatory information for classification problems. The predictive accuracy of six classifiers based on feature selection obtained through MIG method is shown in Table 5.17 and compared with the accuracy obtained by Full Feature Selection (FFS).

Classifier models	Predictive Accuracy Through MIG	Predictive Accuracy Through FFS
KNN	0.776	0.7739
DT	0.7447	0.8434
RF	0.776	0.8173
GB	0.7447	0.913
NB	0.6979	0.7165
SVM	0.776	0.7617

Table 5.17 Comparing Predictive Accuracy of six classifiers with MIG filter

ANOVA

The analysis of variance (ANOVA) can be thought of as an extension to the t-test. The independent t-test is used to compare the means of a condition between 2 groups.

Analysis of variance (ANOVA) is a collection of statistical models and their associated estimation procedures (such as the "variation" among and between groups) used to analyse the differences among group means in a sample. This method was used to obtain an optimal feature subset.

The features ranked by ANOVA method are presented in Table 5.18.

Original Dataset	Ranking by ANOVA Method
0. Age	11. Stream
1. Gender	13. StIntforJob
2. Fedu	10. 10th
3. Foccu	15. ST
4. Medu	4. Medu
5. Moccu	6. Fincome
6. Fincome	2. Fedu
7. NoS	16. STS
8. Address	5. Moccu
9. MoE	19.WhythisCourse
10. 10th	12. Twelth
11. Stream	17.IntInfluence
12. Twelth	0. Age
13. StIntforJob	3. Foccu
14. Health	18. AreaofInterest
15. ST	7. NoS
16. STS	1. Gender

17.IntInfluence	8. Address
18. AreaofInt	14. Health
19.WhythisCourse	9. MoE

Table 5.18 Features ranked by ANOVA

An observation in this algorithm is that two features *StIntforJob* and *Stream* are in top two positions as in the MIG rank features. These two attributes obtained the highest average merit which was the most important attribute for student potential prediction.

The Predictive Accuracy was calculated through applied six classifiers based on selected feature subset, as given in Table 5.19.

Classifier models	**Predictive Accuracy Through ANOVA**	**Predictive Accuracy Through FFS**
KNN	0.9348	0.7739
DT	0.8541	0.8434
RF	0.9479	0.8173
GB	0.8333	0.913
NB	0.7005	0.7165
SVM	0.7604	0.7617

Table 5.19 Comparing Predictive Accuracy of Six classifiers with ANOVA filter

ROC

The Receiver Operator Characteristic (ROC) curve is well-known in evaluating classification performance. Owing to its superiority in dealing with imbalanced and cost-sensitive data, the ROC curve has been exploited as a popular metric to evaluate ML models. The existing ROC-based feature selection approaches are simple and effective in evaluating individual features.

The features ranking using ROC values is presented in Table 5.20. *StIntforJob, Stream* and *Fincome* are the first three features. It is to be noted that the first two features are similar to those ranked by MIG and ANOVA.

Original Dataset	Ranking by ROC Method
0. Age	13. StIntforJob
1. Gender	11. Stream
2. Fedu	6. Fincome
3. Foccu	4. Medu
4. Medu	2. Fedu
5. Moccu	10. 10^{th}
6. Fincome	0. Age
7. NoS	19.WhythisCourse
8. Address	3. Foccu
9. MoE	17.IntInfluence
10. 10th	5. Moccu
11. Stream	18. AreaofInt
12. Twelth	15. ST
13. StIntforJob	14. Health
14. Health	16. STS
15. ST	12. Twelth
16. STS	7. NoS
17.IntInfluence	8. Address
18. AreaofInt	1. Gender
19.WhythisCourse	9. MoE

Table 5.20 Features ranking by ROC

The predictive accuracy based on feature selection obtained through ROC method is shown in Table 5.21.

Classifier models	Predictive Accuracy Through ROC	Predictive Accuracy Through FFS
KNN	0.9348	0.7739
DT	0.8541	0.8434
RF	0.9479	0.8173
GB	0.8333	0.913
NB	0.7135	0.7165
SVM	0.75	0.7617

Table 5.21 Comparing Predictive Accuracy of Six classifiers with ROC filter

Fisher Score & Chi- Squared Method

Fisher Score is one of the most widely used supervised feature selection methods. However, it selects each feature independently according to their scores under the Fisher criterion, which leads to a suboptimal subset of features. This method was used for selecting optimal feature subset.

Table 5.22 shows the ranking features with this method for analysis of student potential in technical programs. The figure shows the result of the classifiers' performance. This

shows the more unsteady trend of the classifier's ability against the feature subsets obtained from filtered methods MIG, ANOVA, ROC, and CHI. The impact of filtered methods resulted in a marginal increase of the accuracy of the classifiers ANOVA and ROC against Full Feature Subset (FFS). The other classifiers underperformed on the filter subsets.

Original Dataset	Ranking by CHI Method
0. Age	13. StIntforJob
1. Gender	11. Stream
2. Fedu	10. 10th
3. Foccu	4. Medu
4. Medu	15. ST
5. Moccu	6. Fincome
6. Fincome	19.WhythisCourse
7. NoS	2. Fedu
8. Address	5. Moccu
9. MoE	12. Twelth
10. 10th	16. STS
11. Stream	17.IntInfluence
12. Twelth	0. Age
13. StIntforJob	3. Foccu
14. Health	18. AreaofInterest
15. ST	1. Gender
16. STS	7. NoS
17.IntInfluence	9. MoE
18. AreaofInt	8. Address
19.WhythisCourse	14. Health

Table 5.22 Feature ranked by CHI

The predictive accuracy measured by six classifiers when applied on the CHI filter method feature subset is shown in Table 5.23, along with the FFS accuracy.

Classifier models	**Predictive Accuracy Through CHI**	**Predictive Accuracy Through FFS**
KNN	0.7756	0.7739
DT	0.7513	0.8434
RF	0.776	0.8173
GB	0.7513	0.913
NB	0.72	0.7165
SVM	0.78	0.7617

Table 5.23 Comparing Predictive Accuracy of Six classifiers with CHI filter

Classifiers	Based on Accuracy Measure	
	Accuracy	**Ranking**
KNN-MIG	0.776	10
KNN-ANOVA	0.9348	2
KNN-ROC	0.9348	2
KNN-CHI	0.7756	11
KNN-FFS	0.7739	12
DT-MIG	0.7447	17
DT-ANOVA	0.8541	5
DT-ROC	0.8541	5
DT-CHI	0.7513	15
DT-FFS	0.8434	6
RF-MIG	0.776	10
RF-ANOVA	0.9479	1
RF-ROC	0.9331	3
RF-CHI	0.776	10
RF-FFS	0.8173	8
GB-MIG	0.7447	17
GB-ANOVA	0.8333	7
GB-ROC	0.8333	7
GB-CHI	0.7513	15
GB-FFS	0.913	4
NB-MIG	0.6979	22
NB-ANOVA	0.7005	21
NB-ROC	0.7135	20
NB-CHI	0.72	18
NB-FFS	0.7165	19
SVM-MIG	0.776	10
SVM-ANOVA	0.7604	14
SVM-ROC	0.75	16
SVM-CHI	0.78	9
SVM-FFS	0.7617	13

Table 5.24 Classification Methods results with Filter Methods

Table 5.24 showed the performance results of six classifier by predictive measures for filter-based feature subsets. From the filter classifier performance, it is observed that RF and KNN are the top ranked classifiers in terms of accuracy measure. It is seen that RF classifier scored the first rank against ANOVA feature subset method. Figure 5.8 shows the Bar Graph comparison between different classifier models based on the different filter methods.

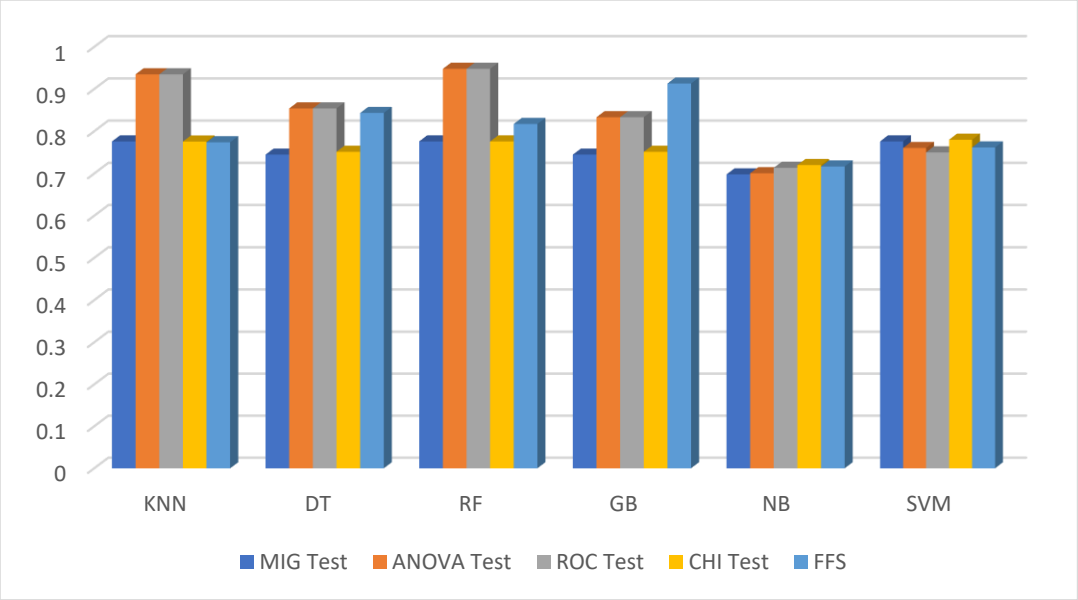

Figure 5.8 Comparison Analysis of Classification Methods with Filter Methods

5.5 Evaluation of students based on Education Boards

The previously used dataset of the students was reformed, based on the different higher secondary School Boards i.e., RBSE and CBSE. Data pre-processing was done on both the datasets. After data pre-processing, both the datasets have been compared, based on 8 attributes. The number of female students is much more in CBSE as compared to RBSE. Figure 5.9 represents the gender feature.

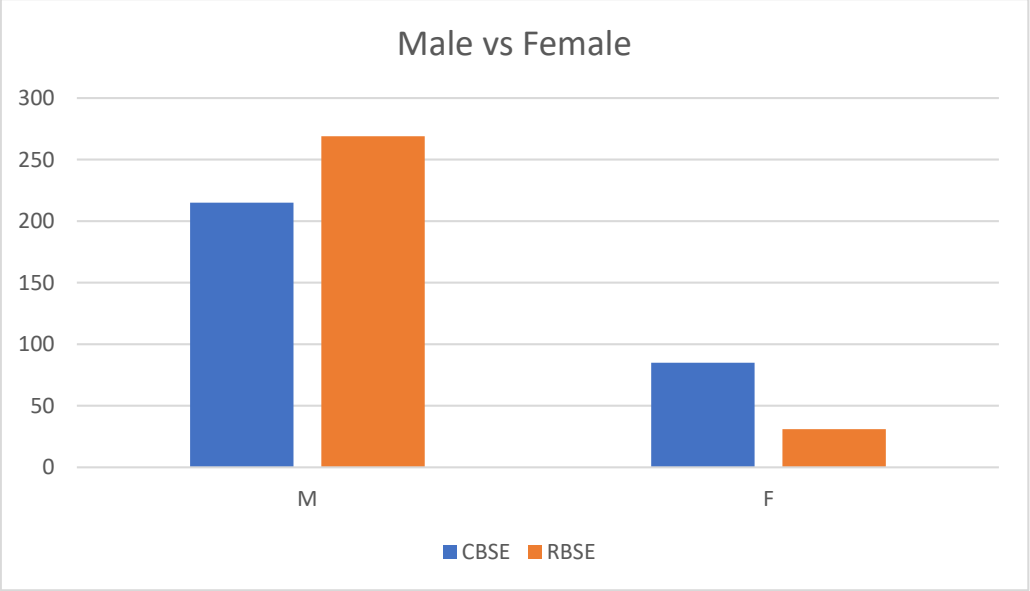

Figure 5.9 Gender Ratio

The medium of education also impacts choosing the higher programs. More English medium students take up technical programs, than Hindi medium, as shown in Figure 5.10.

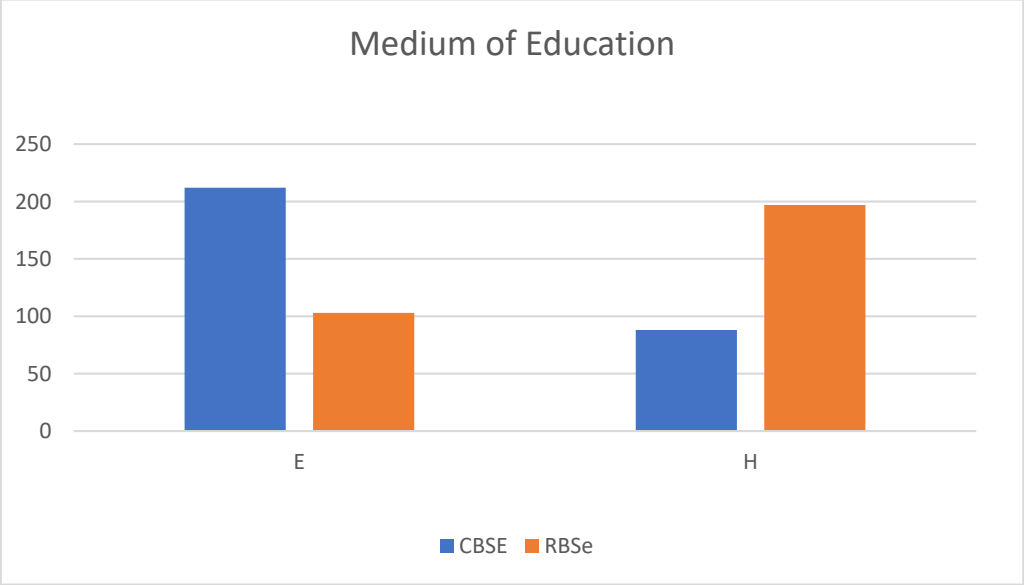

Figure 5.10 Data based on Medium of Education

Income is also an important attribute for choosing technical programs. High Income group students take admission in technical programs, as shown in Figure 5.11.

Figure 5.11 Data based on Family Income

This study concludes that more students choose technical programs from CBSE as compared to RBSE. Figure 5.12 shows the ratio between students from Technical and Non Technical programs.

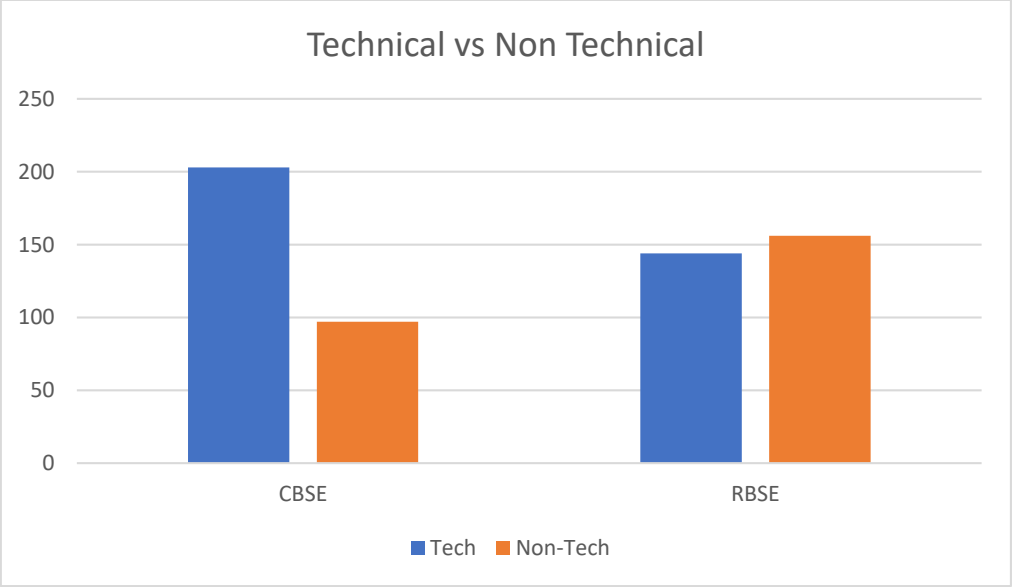

Figure 5.12 Students in Technical vs non-technical programs

Different Machine Learning algorithms have been applied on both the datasets. Python3 was used to implement different Machine Learning algorithms. This study calculated the accuracy of different classifiers to predict the student potential.

Ensemble Methods have also been used to provide better accuracy and produce one optimal predictive model. We have applied some Ensemble Methods on the student dataset and also RBSE and CBSE datasets. We have applied five algorithms, Bagged

Decision Trees, Random Forest, Voting Classifier, Adaboost and Stochastic Gradient Boosting, based on bagging and boosting. We calculated the accuracy of different Ensemble Methods for prediction of student potential.

5.5.1 Evaluation using Machine Learning Techniques

The Dataset was divided into two parts, 75% of which is used to train the models and 25% is held back as a test dataset. Both training dataset and test dataset are One-hot encoded. K is Chosen for K-fold cross-validation to estimate the accuracy of different models. Tool python3 is used to run different Machine Learning algorithms. Machine Learning algorithms have been applied on both datasets to calculate the accuracy of prediction. Five different ML algorithms, Decision Tree, Naive Bayes, Logistics Regression, K-Nearest Neighbor, and Support Vector Machine were used on the same dataset. The final classification accuracy is considered and compared with each other as shown in Tables 5.25 and 5.26. These tables show the results based on both datasets. It is found that Decision Tree gave higher accuracy from both the datasets.

ML Model	**Accuracy**
Logistics Regression (LR)	0.797288
K-Nearest Neighbor (KNN)	0.807401
Decision Tree (DT)	0.824237
Naïve Bayes (NB)	0.733277
Support Vector Machine (SVM)	0.817684

Table 5.25 Accuracy comparison for RBSE Dataset

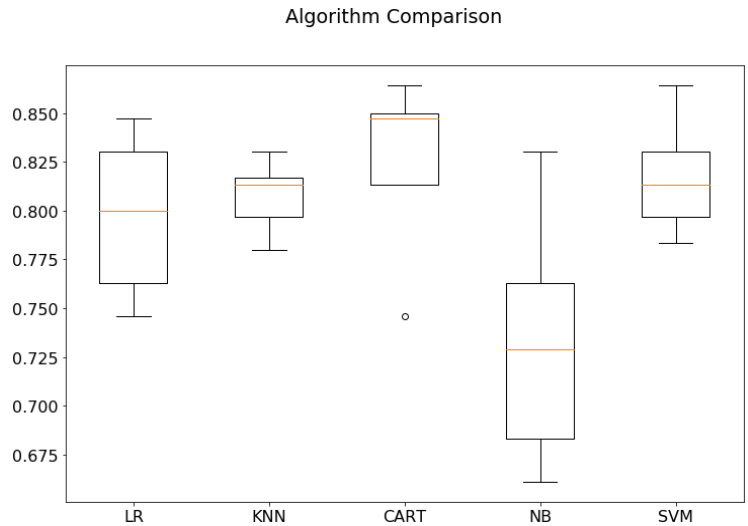

Figure 5.13 Comparison for RBSE Dataset

ML Model	Accuracy
Logistics Regression (LR)	0.840839
K-Nearest Neighbor (KNN)	0.828485
Decision Tree (DT)	0.868345
Naïve Bayes (NB)	0.477855
Support Vector Machine (SVM)	0.850023

Table 5.26 Accuracy comparison for CBSE Dataset

Whisker plots are also used, to show the accuracy scores of each cross-validation of 10 folds for each Machine Learning algorithm, using both the datasets. These are shown in Figures 5.13 and 5.14.

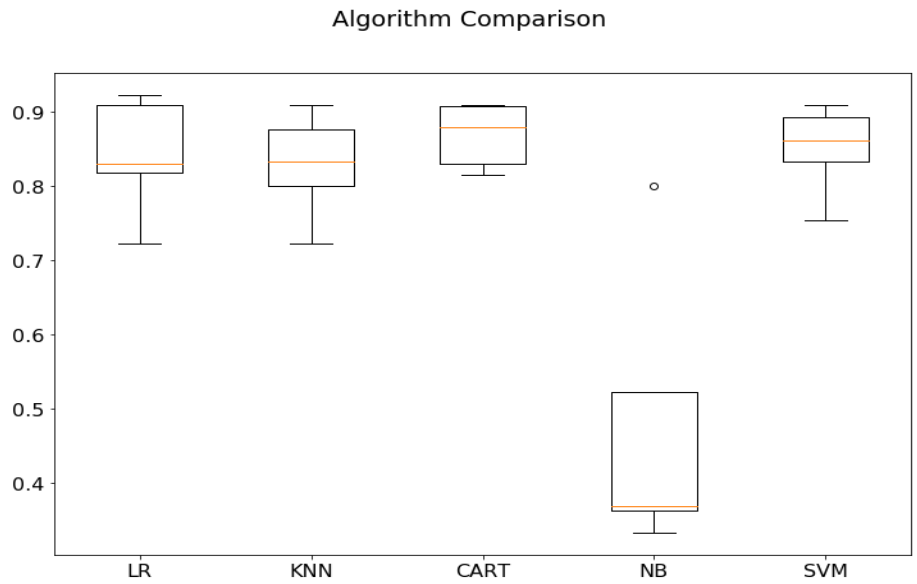

Figure 5.14 Comparison for CBSE Dataset

5.5.2 Evaluation using Ensemble Methods

The dataset for the study is split into training and test datasets in a ratio of 70:30 with 10-fold cross-validation, for estimating the performance. We have applied five algorithms: Bagged Decision Trees, Random Forest, Voting Classifiers, Adaboost and Stochastic Gradient Boosting based on bagging and boosting. Tables 5.27 and 5.28 show the performance accuracy of the five Ensemble models, while Figures 5.15 and 5.16 graphically show the different variations in the prediction. It is seen that Bagged Decision Tree gave higher accuracy for both the datasets.

Ensemble Model	Accuracy
Bagged Decision Trees	0.84827
Random Forest	0.84558
Voting Classifier	0.81269
Adaboost	0.79256
Stochastic Gradient Boosting	0.81019

Table 5.27 Accuracy Comparison for RBSE Dataset based on Ensemble Methods

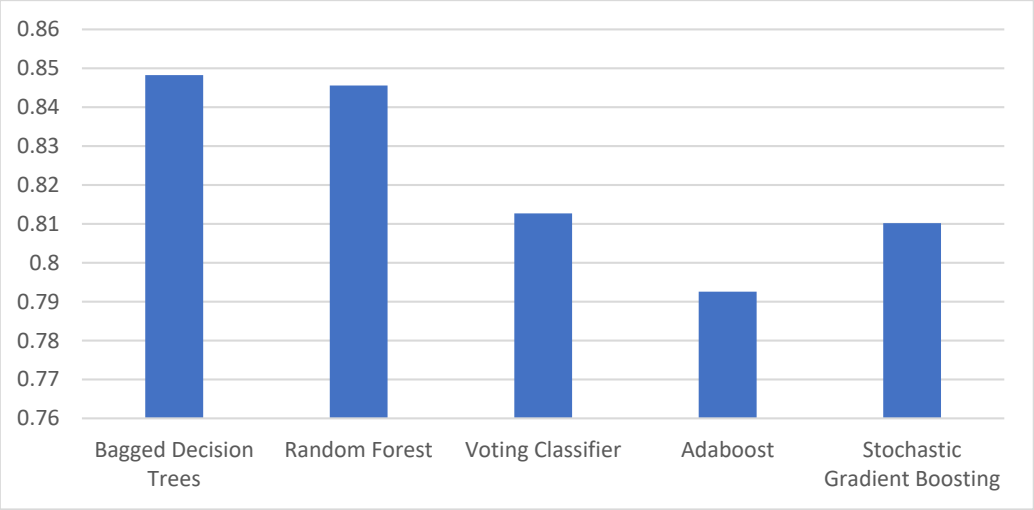

Figure 5.15 Graphical Comparision for RBSE dataset based on Ensemble Methods

Ensemble Model	Accuracy
Bagged Decision Trees	0.89974
Random Forest	0.89292
Voting Classifiers	0.83975
Adaboost	0.82373
Stochastic Gradient Boosting	0.86047

Table 5.28 Accuracy Comparison for CBSE Dataset based on Ensemble Methods

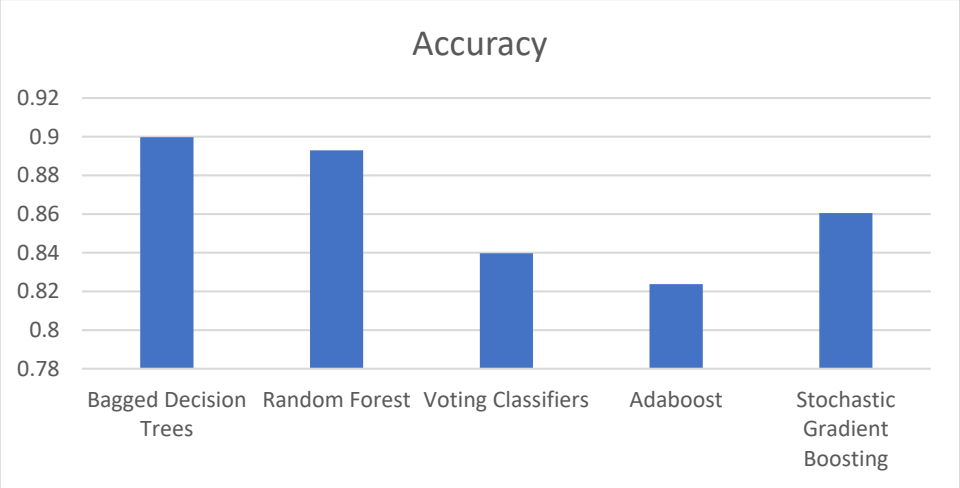

Figure 5.16 Graphical Comparison for CBSE dataset based on Ensemble Methods

5.7 Chapter Summary

This chapter focused on determining the best classifier model. It determined the classifier model performance with optimal feature subset, by applying different ML techniques. One of the objectives of the study has been achieved through applying different classifier models on student data to analysis of student potential. Another objective was fulfilled to apply classifier models based on Education Boards. The accuracy of different classifier models was also compared based on different metrics.

Different models were built including KNN, NB, SVM, DT, RF and GB and their performance was compared using six metrics: accuracy, precision, sensitivity, specificity, f-measure and ROC curve.

In chapter 6, we report on the performance of different classifier models to predict student performance. The aim of this comparison is to find the best classifier model which gives better predictive accuracy of performance.