Introduction

The NeurIPS 2020 Education Challenge was an international machine learning competition in which participants aimed to predict students' answers to assessments accurately, determine question quality, and identify a personalized sequence of questions for each student that best predicted the student's answers.

The assessments are in the form of diagnostic questions: the answers that the students give to these questions reveal the understanding of concepts by the students. The analysis of students' responses provides information about students learning levels and hence offer better recommendations for learning curriculum.

In this project, there are two tasks that are performed with NeurIPS 2020 Education Challenge datasets. Firstly, to predict the correctness of the answer selected by the student, and secondly, to predict the answer chosen (A, B, C or D).

Background

Machine learning techniques: The analysis is done by using classification, which is one of the most important aspects of supervised learning. SVM, Naive Bayes, Logistic regression, Decision tree and Random Forest are some classification methods used to find the best model for accurate predictions.

These techniques will bring fundamental advances to educational data mining technologies, particularly to analyze students' learning progress and recommend personalized learning curricula. These methods will be deployed in a real educational platform where they will improve the learning outcomes of millions of students.

Education with predicting students' performances enable personalized assessments for each student to improve learning outcomes. This study's real-world impact is to recommend questions of appropriate difficulty to the student that best fits their background and learning status. This also helps to determine common misconceptions that students have by clustering question-answer pairs which may indicate the same or related misconceptions.

Data Description

Data from two school years (September 2018 to May 2020) of students' answers to mathematics questions from Eedi is used to perform the predictions. Esdi is a leading educational platform which millions of students interact with daily around the globe. Eedi offers diagnostic questions to students from primary to high school (roughly between 7 and 18 years old). Each diagnostic question is a multiple-choice question with 4 possible answer choices, exactly one of which is correct. Currently, the platform mainly focuses on mathematics questions.

There are four datasets used for this analysis. Train_task_1_2.csv, answer_metadata_task_1_2.csv, question_metadata_task_1_2.csv, and student_metadata_task_1_2.csv.

Train_task_1_2.csv has columns 'QuestionId', 'UserId', 'AnswerId', 'IsCorrect', 'CorrectAnswer',and 'AnswerValue'.

This data does not seems to have enough information for the predictions, so this is merged with some of the metadata from answer, question and student dataset. Some feature engineering is also performed to get the "age" column and unwanted columns are removed. The final dataset has following columns and datatypes.

```
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   QuestionId     67854 non-null  int64
 1   UserId         67854 non-null  int64
 2   AnswerId       67854 non-null  int64
 3   IsCorrect      67854 non-null  int64
 4   CorrectAnswer  67854 non-null  category
 5   AnswerValue    67854 non-null  category
 6   GroupId        67854 non-null  int64
 7   QuizId         67854 non-null  int64
 8   Gender         67854 non-null  category
 9   SubjectId      67854 non-null  category
 10  age            67854 non-null  int64
```

The final dataset has the following columns as a dataframe.

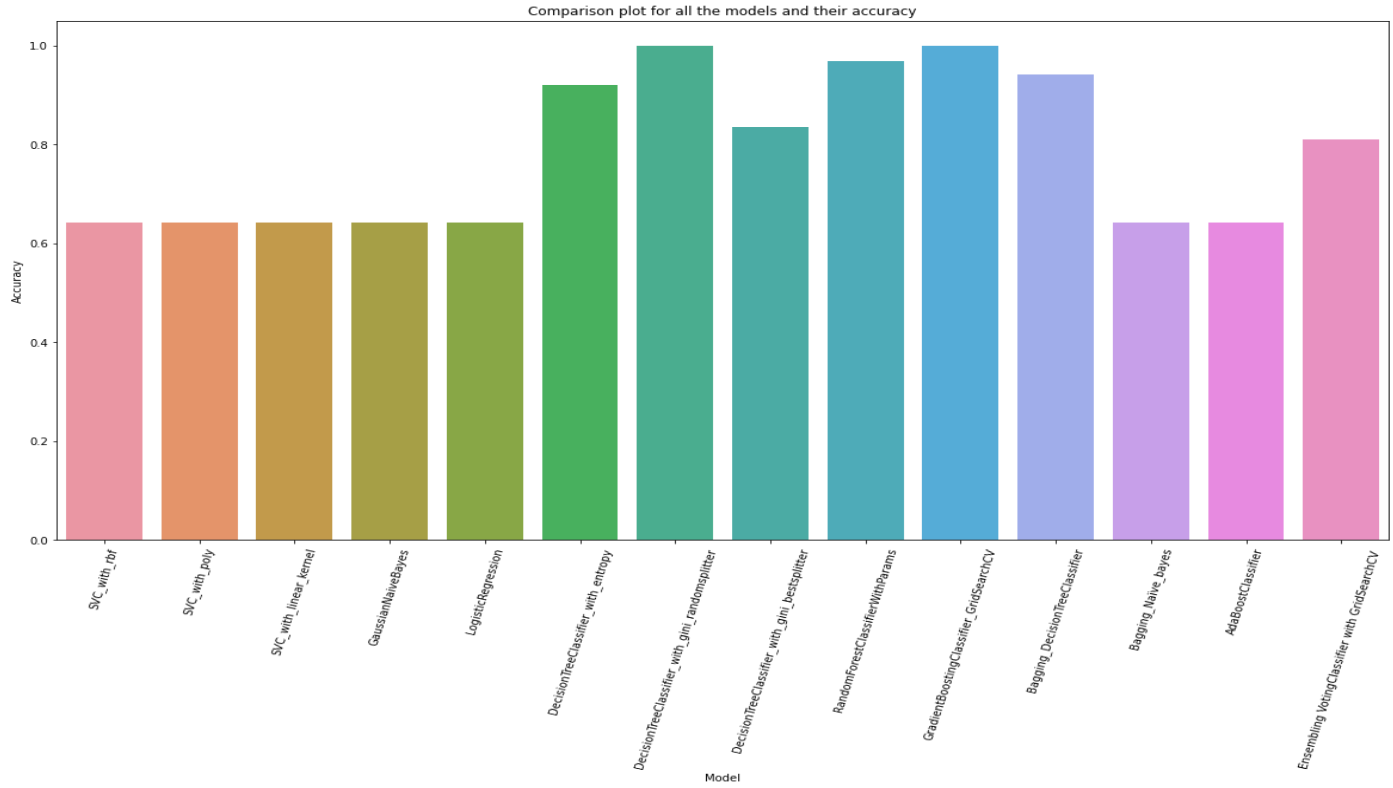
QuestionId	UserId	AnswerId	IsCorrect	CorrectAnswer	AnswerValue	GroupId	QuizId	Gender	SubjectId	age
24909	99434	10909007	1	3	3	10754	13392	1	570	18
18931	94775	10809477	0	1	2	3269	7990	2	927	18

Task 1: predict the correctness of answer chosen by a student

Following are the results for task 1 with the models executed and the respective metrics.

Model	Accuracy	Precision	Recall	F1 score	Confusion matrix	ROC	Perf counter
SVC_with_rbf	0.642653	0.642653	1.000000	0.782457	[[0, 4871], [0, 8760]]	0.496782	953.916000
SVC_with_poly	0.642653	0.642653	1.000000	0.782457	[[0, 4871], [0, 8760]]	0.498175	450.671853
SVC_with_linear_kernel	0.642653	0.642653	1.000000	0.782457	[[0, 4871], [0, 8760]]	0.505876	350.336138
GaussianNaiveBayes	0.642653	0.642653	1.000000	0.782457	[[0, 4871], [0, 8760]]	0.512481	0.016979
LogisticRegression	0.642726	0.642700	1.000000	0.782492	[[1, 4870], [0, 8760]]	0.510815	0.667368
DecisionTreeClassifier_with_entropy	0.920842	0.938363	0.938470	0.938417	[[4331, 540], [539, 8221]]	0.913805	0.864787
DecisionTreeClassifier_with_gini_randomsplitter	1.000000	1.000000	1.000000	1.000000	[[4871, 0], [0, 8760]]	1.000000	0.047348
DecisionTreeClassifier_with_gini_bestsplitter	0.835302	0.876111	0.866210	0.871133	[[3798, 1073], [1172, 7588]]	0.822963	0.901704
RandomForestClassifierWithParams	0.968308	0.953003	1.000000	0.975936	[[4439, 432], [0, 8760]]	0.999906	18.270244
GradientBoostingClassifier_GridSearchCV	1.000000	1.000000	1.000000	1.000000	[[4871, 0], [0, 8760]]	1.000000	776.446885
Bagging_DecisionTreeClassifier	0.942557	0.918038	0.999886	0.957215	[[4089, 782], [1, 8759]]	0.999853	296.784772
Bagging_Naïve_bayes	0.641479	0.642989	0.994064	0.780881	[[36, 4835], [52, 8708]]	0.501475	15.529298
AdaBoostClassifier	0.642579	0.642627	0.999886	0.782403	[[0, 4871], [1, 8759]]	0.507751	2.146856
Ensembling VotingClassifier with GridSearchCV	0.810579	0.772591	0.999429	0.871491	[[2294, 2577], [5, 8755]]	0.998046	361.723549

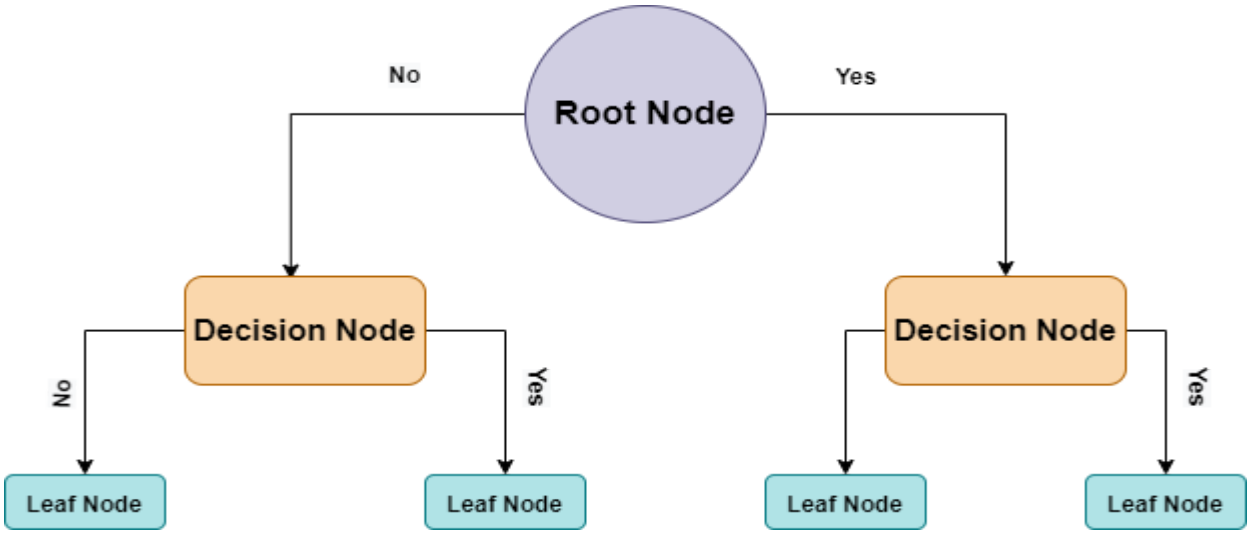
Below is the plotting for the accuracy of from the classification models.



The highest accuracy is of Decision Tree classification with criteria set to “gini” and splitter set to random.

Decision Trees usually mimic human thinking ability while making a decision and it is a tree-like structure, so it is easy to understand.

Decision tree starts root node. It represents the entire dataset, which further gets divided into two or more homogeneous sets. The final output nodes are leaf nodes, So the tree cannot be segregated further after getting a leaf node.
Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.



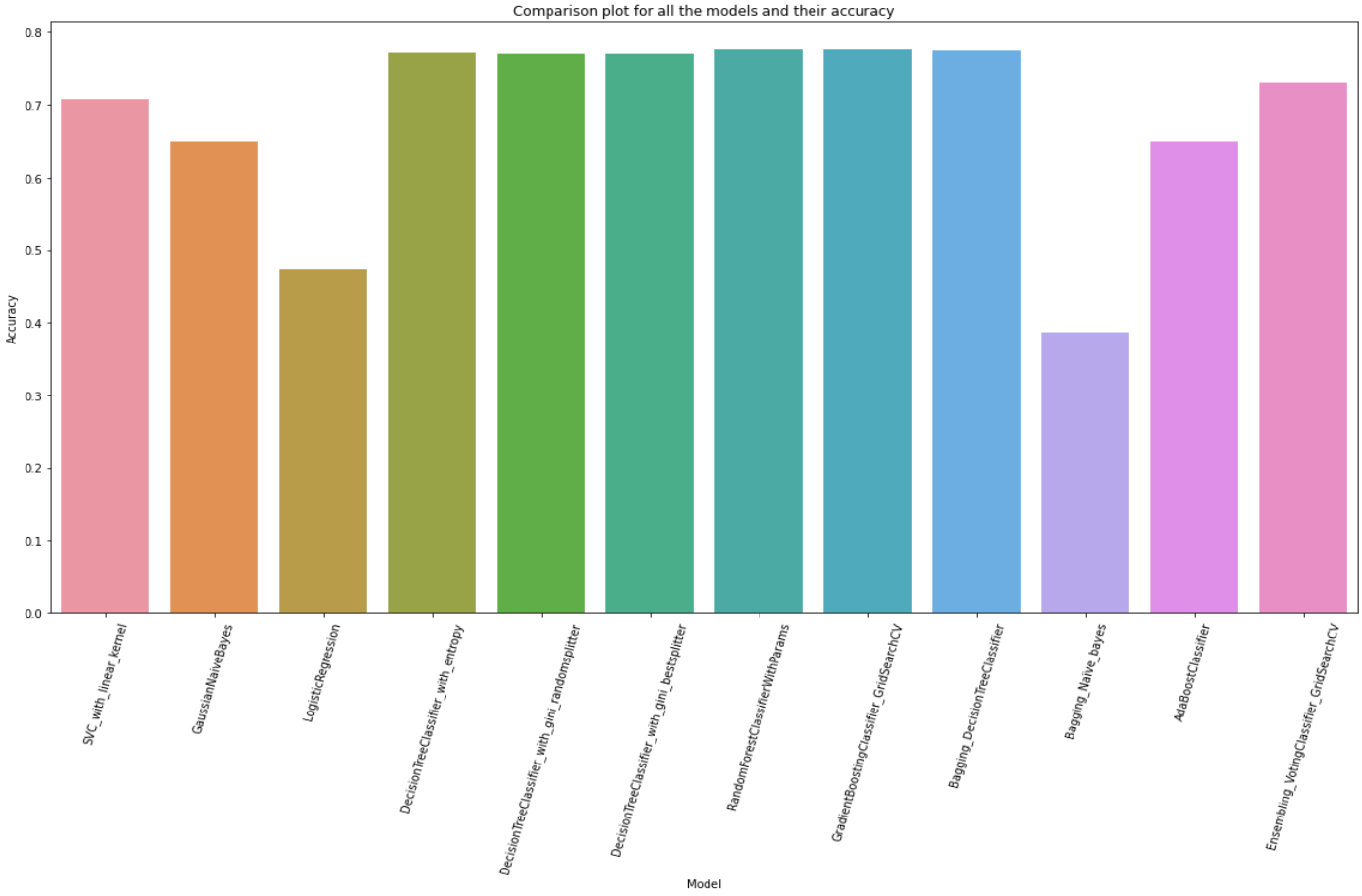
A decision tree can be “learned” by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions.
Decision trees can handle high-dimensional data. In general decision tree classifier has good accuracy. The criteria that best performed with decision tree in task 1 is “gini”. The Gini impurity is used to predict the likelihood of a randomly chosen example being incorrectly classified.

Task 2: Predict the answer chosen by a student

Following are the results for task 2 with the models executed and the respective metrics.

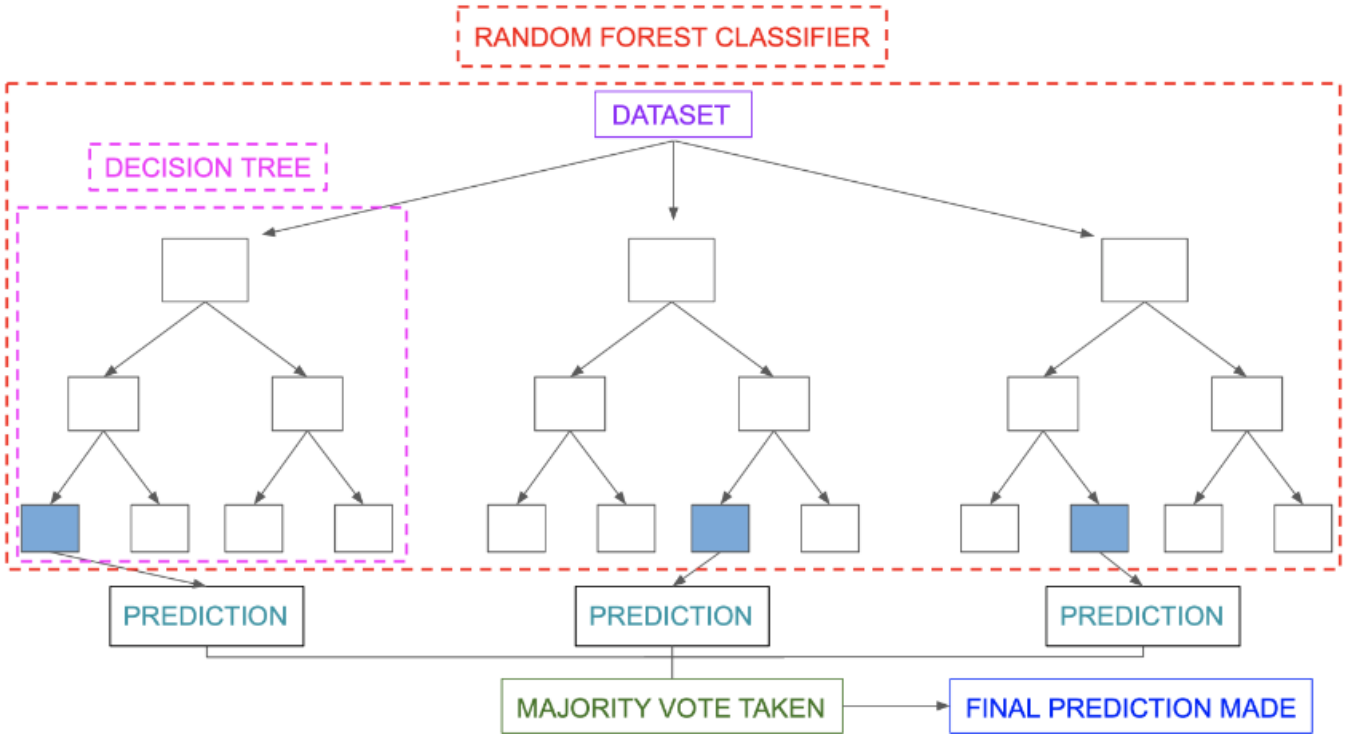
Model	Accuracy	Precision	Recall	F1 score	Confusion matrix	ROC	Perf counter
SVC_with_linear_kernel	0.708021	0.708021	0.708021	0.708021	[[2486, 394, 0, 365], [402, 2685, 0, 423], [84...	0.744163	573.571993
GaussianNaiveBayes	0.649375	0.649375	0.649375	0.649375	[[2141, 345, 394, 365], [402, 2233, 452, 423],...	0.767292	0.024081
LogisticRegression	0.474540	0.474540	0.474540	0.474540	[[2193, 222, 250, 580], [830, 1050, 749, 881],...	0.689920	3.046419
DecisionTreeClassifier_with_entropy	0.772112	0.772112	0.772112	0.772112	[[2490, 289, 234, 232], [268, 2693, 278, 271],...	0.848001	0.903397
DecisionTreeClassifier_with_gini_randomsplitter	0.770125	0.770125	0.770125	0.770125	[[2515, 263, 233, 234], [274, 2699, 274, 263],...	0.846752	0.084700
DecisionTreeClassifier_with_gini_bestsplitter	0.770567	0.770567	0.770567	0.770567	[[2491, 274, 263, 217], [263, 2721, 255, 271],...	0.846996	0.516160
RandomForestClassifierWithParams	0.776085	0.776085	0.776085	0.776085	[[2227, 581, 409, 28], [49, 3095, 326, 40], [7...	0.959894	722.467768
GradientBoostingClassifier_GridSearchCV	0.776085	0.776085	0.776085	0.776085	[[2227, 581, 409, 28], [49, 3095, 326, 40], [7...	0.959894	2908.598384
Bagging_DecisionTreeClassifier	0.774614	0.774614	0.774614	0.774614	[[2417, 351, 302, 175], [190, 2822, 301, 197],...	0.960657	134.474864
Bagging_Naïve_bayes	0.387344	0.387344	0.387344	0.387344	[[2466, 15, 758, 6], [2575, 21, 904, 10], [834...	0.740846	18.041013
AdaBoostClassifier	0.649227	0.649227	0.649227	0.649227	[[2142, 345, 394, 364], [402, 2233, 452, 423],...	0.770841	2.302063
Ensembling_VotingClassifier_GridSearchCV	0.730611	0.730611	0.730611	0.730611	[[2457, 107, 650, 31], [533, 2378, 562, 37], [...	0.932649	292.422798

Below is the plotting for the accuracy of from the classification models.



From the above results we can see that RandomForestClassifier with GridSearchCV performed the best with accuracy 0.77 and execution time 722 seconds. We can use this model to predict the answer value selected by a student.

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. The diagram below shows a simple random forest classifier.



Here GridSearchCV is used to find the best parameters.It is a process that searches exhaustively through a manually specified subset of the hyperparameter space of the targeted algorithm.The hyperparameters for the models were:

Max_depth = range(5,16,2)

Min_samples_split = range(200,1001,200)

cv=5

Conclusion

The analysis was performed to predict students' answers to assessments accurately, determine question quality, and identify a personalized sequence of questions for each student that best predicted the student's answers.

In this project, the two tasks are achieved by executing the classification models to find the best model for predicting the correctness of the answer and the answer value. The comparisons of the accuracy of models are plotted and predictions are generated with the best model identified based on the evaluation metrics. Best models for task 1 and task 2 are Decisions tree with "gini" and RandomForst classifier with GridsearchCV respectively.