The following is the process I went through to finish this assignment:

Merged both datasets using Unique-id as a connecting point
Create 10 new features using Python:

- Feature 1: Average Time taken to complete the actions that the students got correct. This is a great way to understand how long it took a student to understand and execute the questions they are getting correct
- Feature 2: The percent of recent actions students have gotten wrong. Can help us understand how much students are answering recent questions incorrectly to understand better how to help them going forward
- Feature 3: Change in average time per action. It helps us understand how long the time the student is spending on these actions is changing to help us better understand if students are getting faster or slower.
- Feature 4: Score based on correctness, time, and standard deviations. It helps us better understand if the actions have been efficient.
- Feature 5: Change in knowledge or proficiency from the previous time point. Can help us understand if students are growing, stagnating, or having more difficulty in their actions.
- Feature 6: Difference between the average correctness of bug-related actions and right actions. Can show patterns in students' abilities to fix their actions versus getting them correctly on the first try.
- Feature 7: Rate at which knowledge or proficiency changes over time. This can help us evaluate if students' knowledge is moving in a positive direction to indicate growth in understanding.
- Feature 8: Average change in time per action compared to the previous action. Helps us understand if students are getting faster or slower during the actions.
- Feature 9: Average duration of all actions, including right and wrong actions. Can help with understanding if the amount of time a student takes to complete an action is related to their understanding of the task and if they need more guidance. Also helps with understanding the efficiency of their solution processes.
- Feature 10: Calculating the change in recent correctness to better understand if students are gradually getting more questions correctly based on the help.

After creating the new features:

- I created a correlation matrix. I realized that two of my variables changeInKnowledge and knowledgeGrowthRate were resulting in the same data even though they were calculting them in different ways. So I decided to exclude one of them in the next steps.
- I then merged the original data set with the new features including the values I calculated the new features from the new dataset. These are the data-points that I merged in the new CSV file: "Unique-id", "namea_x", "OffTask", "Avgright", "Avgbug", "Avghelp", "Avgchoice", "Avgstring", "Avgnumber", "Avgpoint", "Avgpchange", "Avgtime", "AvgtimeSDnormed", "Avgtimelast3SDnormed", "Avgtimelast5SDnormed", "Avgnotright", "Avghowmanywrong-up", "Avghelppct-up", "Avgwrongpct-up", "Avgtimeperact-up", "AvgPrev3Count-up", "AvgPrev5Count-up", "Avgrecent8help", "Avg recent5wrong", "Avgmanywrong-up",

"AvgasymptoteA-up", "AvgasymptoteB-up", 'avgTimeCorrectAction', 'percentageRecent5Wrong', 'timePerActionChange', 'efficiencyScore', 'changeInKnowledge', 'avgBugVsAvgRight', 'totalWrongActions', 'avgActionDuration', 'recentCorrectnessChange', 'help', 'Pknow-2', 'pknow-1', 'time'

- I then chose a few ML algorithms and decided to calculate them. I first calculated the baseline with a Random Forest Classifier. I then performed cross-validation and then calculated the Random Forest Classifier using the new features data.
- Next, I performed a few more ML algorithms and the following are the results I got:

Baseline for Random Forest Classifier:
RFC Cohen's Kappa: 0.2776203966005666
RFC Accuracy: 0.9673202614379085
RFC F1 Score 0.2857142857142857
RFC Precision 1.0

Cross-validated Random Tree Classifier:
RFC Cross-validated Cohen's Kappa: 0.16010622352344023
RFC Cross-validated Accuracy: 0.9782234957020057
RFC Cross-validated F1 Score: 0.16636363636363635
RFC Cross-validated Precision: 0.4333333333333333

Baseline for Decision Forest Classifier:
DTC Cohen's Kappa: 0.5870445344129555
DTC Accuracy: 0.9738562091503268
DTC F1 Score 0.6
DTC Precision 0.75

Cross-validated Decision Forest Classifier:
DTC Cross-validated Cohen's Kappa: 0.19628662584245254
DTC Cross-validated Accuracy: 0.9621776504297994
DTC Cross-validated F1 Score: 0.21523809523809523
DTC Cross-validated Precision: 0.18867243867243866

Baseline for XGBClassifier:
XGBC Cohen's Kappa: 0.49
XGBC Accuracy: 0.9738562091503268
XGBC F1 Score 0.5
XGBC Precision 1.0

Cross-validated for XGBClassifier:
XGBC Cross-validated Cohen's Kappa: 0.18578412095032398
XGBC Cross-validated Accuracy: 0.9810888252148999
XGBC Cross-validated F1 Score: 0.18888888888888888
XGBC Cross-validated Precision: 0.8

Baseline for GaussianNB:
GNB Cohen's Kappa: 0.010921531162632614
GNB Accuracy: 0.3249283667621777
GNB F1 Score: 0.05161691790199776
GNB Precision: 0.02660360186938874

Cross-validated for GaussianNB:
GNB Cross-validated Cohen's Kappa: 0.010921531162632614
GNB Cross-validated Accuracy: 0.3249283667621777
GNB Cross-validated F1 Score: 0.05161691790199776
GNB Cross-validated Precision: 0.02660360186938874

I was a little disappointed to see my kappa drop all of the instances and was unable to figure out how to pick these backups. However, I was glad to see my accuracy score go up for XGBClassifier and Random Forest Classifier.

Here is my code and dataset for further context: [https://github.com/vedyakonda/CA2Code]