Predicting Post-Calculus I Math Confidence

ISYE 7406 Course Project

Group 143: Emily Dobar

2024-04-01

Introduction

Math anxiety, a feeling of “worry of fear about performing math calculations,” is a common issue that affects many different people all over the world, regardless of age or race or area of study or work (West (2022)). For many, math anxiety can begin as young as kindergarten and follow them into post-graduate studies and into their careers. It can lead to lower math confidence and fuel avoidance of further math courses, which can significantly limit future career options and success.

Research has proven that instructors can play a key role in influencing student anxiety and self-confidence levels in their courses (Khasawneh, Gosling, and Williams (2021)). A teacher’s attitudes towards math can directly translate to their students, suggesting that a teacher exhibiting confidence and enjoyment in teaching math could directly lead to an increase in student enjoyment and confidence in math (Christensen and Knezek (2020)). With increased math confidence, students could be encouraged and develop lifelong skills that could benefit them in their future academic and professional roles.

Math confidence can be defined as “a student’s perception of their ability to attain good results and their assurance that they can handle difficulties in mathematics” (Pierce and Stacey (2004)). Math confidence is important and can help students both in and out of school embrace and value mistakes, develop self-reliance and perseverance, take healthy risks and handle new problems, work both independently and collaboratively, and make cross-content connections (Audet (2018)). Since instructors can be significant influences in student math confidence, exploring student perceptions of instructor quality and pedagogical practices in their math courses could allow for a better understanding of their relationships with student math confidence (Zakariya (2022)).

Ellis et al. (2016) contributed to student STEM education persistence research by performing an analysis on gender disparities in STEM persistence after taking introductory mathematics courses, in particular Calculus I. The data for their research was anonymized and made publicly available for download (One (2024)). The data was collected through a national survey under the Mathematical Association of America and targeted 2- and 4-year college students across the United States who were taking Calculus I and was collected both at the beginning and end of the semester-long course, with students responding to survey constructs to gauge their individual math identities and perceptions before and after taking the Calculus I course (Ellis, Fisdick, and Rasmussen (2016)).

The data collected included student self-rated math confidence based on rating how strongly they agree or disagree with the statement “I am confident in my mathematical abilities” both at the beginning and end of their Calculus I courses. Two other survey constructs were included in the overall data set: one focused on students rating the quality of their Calculus I instructors and another focused on the student-centered pedagogical practices the instructors implemented in their Calculus I courses. Using the available survey data, this research will explore the relationships between the pre-post differences of student self-rated math confidence before and after their Calculus I courses (an introductory course in most colleges/universities) and student perceptions of instructor quality and pedagogical practices as a means of attempting to predict post-Calculus I math confidence change in college and university students.

Exploratory Data Analysis

The raw data set contains 37 columns with 13,409 student responses. After cleaning the data, including reducing the data to include only the pre and post *Confidence* items, as well as the *Instructor Quality* and *Pedagogical Practices* items, and removing observations containing NA values, the data set was approximately 22% of the original, well within the suggested same size of the data population to be representative (Charan and Biswas (2013)), with 18 columns of 2903 student responses.

Table 1 summarizes the *Instructor Quality* items and response distributions. *Instructor Quality* was determined by 8 questions that students rated on the post survey. For analysis, construct questions were given variable names, which are indicated in parentheses after the items. On average, students tended to rate items between a 4 and 5 on a 6-point Likert scale (1, Strongly Disagree to 5, Strongly Agree), suggesting that most students felt as though their instructors were of relatively high quality. The highest average rating was a 5.06 out of 6 for *Appointments*, indicating that most students felt their Calculus I instructors were available to make appointments outside of office hours. *Discouraged* was reverse-coded, so a lower average score is more desirable for that item, and with an average rating of 2.22 out of 6, it suggests that most students did not feel discouraged from continuing to take calculus.

Table 1. Instructor Quality Items (n = 2903)

My Calculus instructor…	Mean\(^{1,2}\)		1	2	3	4	5	6
allowed time for me to understand difficult ideas (Time)	4.34		4%	6%	12%	26%	35%	17%
asked questions to determine if I understood what was being discussed (AskedQs)	4.41		3%	6%	9%	26%	39%	16%
discouraged me from wanting to continue taking calculus (Discouraged)	2.22		40%	30%	12%	8%	5%	4%
discussed applications of calculus(Applications)	4.77		1%	3%	7%	20%	43%	26%
helped me become a better problem solver (ProblemSolver)	4.38		3%	5%	11%	29%	33%	18%
listened carefully to my questions and comments (Listened)	4.77		2%	3%	6%	19%	42%	27%
provided explanations that were understandable (Explanations)	4.61		3%	5%	8%	21%	39%	24%
was available to make appointments outside of office hours, if needed (Appointments)	5.06		1%	1%	4%	16%	40%	38%

[1] Scale: 1 - Strongly Disagree, 2 - Disagree, 3 - Slightly Disagree, 4 - Slightly Agree, 5 - Agree, 6 - Strongly Agree
[2] Reverse-Coded: 'Discouraged'

Table 2 summarizes the *Pedagogical Practices* items and response distributions. *Pedagogical Practices* was determined by 8 questions that students rated on the post survey. For analysis, construct questions were given variable names, which are indicated in parentheses after the items. The average ratings vary for the *Pedagogical Practices* items, suggesting different Calculus I classroom experiences for the students. The highest average rating was 5.25 on a 6-point Likert scale (1, Not at all to 6, Very often), for *Lecture*, suggesting that most students experienced instructor lectures in their Calculus I classes. The lowest average was 1.60 out of 6 for *Presentations*, suggesting that very few students were required to give presentations as part of their Calculus I classes. Student ratings indicate Calculus I instructors tend to favor students asking questions, working on specific problems, and working individually, while activities like presentations, group work, and having students explain their thought processes were not as common.

Table 2. Pedagogical Practices Items (n = 2903)

During class time, how frquently did your instructor…	Mean\(^{1}\)		1	2	3	4	5	6
ask questions? (AskQuestions)	4.62		2%	4%	10%	24%	33%	27%
ask students to explain their thinking? (ExplainThinking)	3.64		14%	15%	15%	21%	21%	14%
have students give presentations? (Presentations)	1.60		74%	10%	6%	5%	3%	2%
have students work individually on problems or tasks? (Individually)	3.71		16%	11%	14%	21%	21%	17%
have students work with one another? (WorkTogether)	2.96		34%	15%	12%	14%	14%	12%
hold a whole-class discussion? (Discussion)	3.25		25%	15%	13%	16%	16%	15%
lecture? (Lecture)	5.25		1%	2%	5%	12%	25%	56%
show how to work specific problems? (SpecificProblems)	4.98		1%	3%	6%	18%	32%	40%

[1] Scale: 1 - Not at all to 6 - Very often

Table 3 summarizes the means and distributions of the pre and post math confidence questions asked of student participants. Students were asked to rate themselves on the statement "I am confident in my mathematical abilities" on a 6-point Likert scale (1, Strongly Disagree to 6, Strongly Agree) both before and after completing a Calculus I course. Interestingly, a statistically significant decrease in math confidence from before to after taking Calculus I presented itself, as students rated themselves on average 4.92 out of 6 before Calculus I and 4.43 out of 6 after, suggesting that Calculus I may be a turning point for students and their personal math confidence levels.

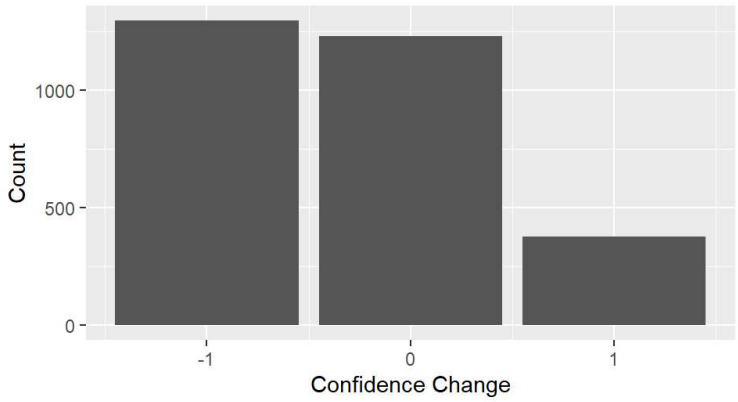
Table 3. Pre-Post Math Confidence (n = 2903)

I am confident in my mathematical abilities.	Mean\(^{1}\)	p		1	2	3	4	5	6
1. Before Calculus I	4.92	0****		1%	2%	5%	20%	42%	31%
2. After Calculus I	4.43			3%	6%	9%	26%	42%	15%

[1] Scale: 1 - Strongly Disagree, 2 - Disagree, 3 - Slightly Disagree, 4 - Slightly Agree, 5 - Agree, 6 - Strongly Agree

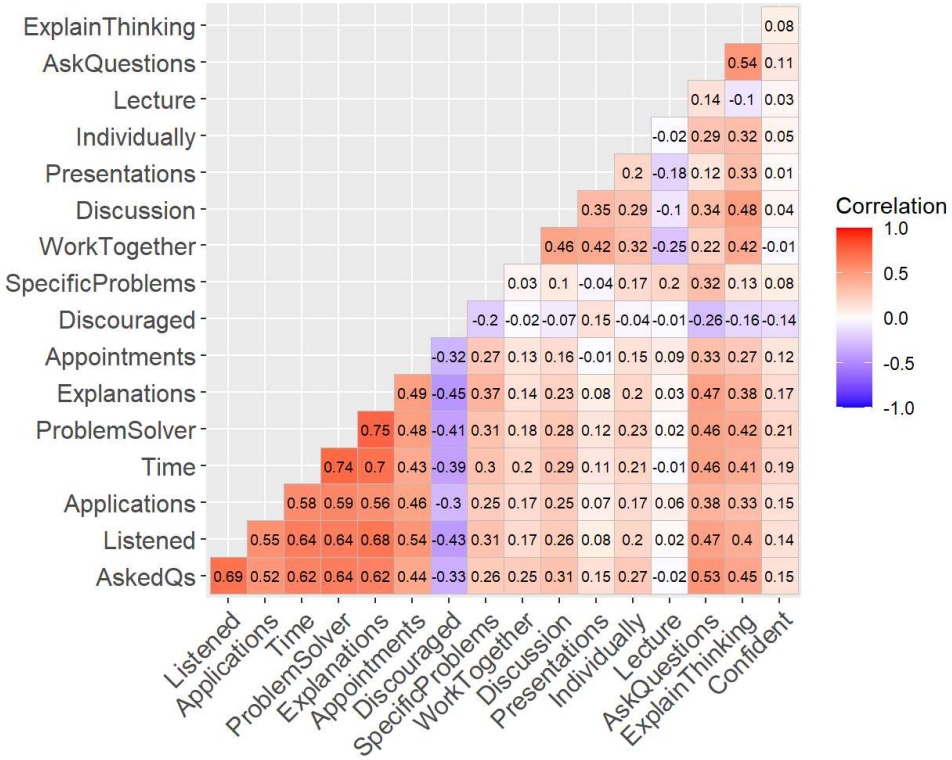
The pre and post *confidence* items were used to create a categorical response variable, *confidence change*, where the pre values were subtracted from the post values. *Confidence change* equals -1 when a confidence decrease was present from pre to post, 1 when a confidence increase was present from pre to post, and 0 when there was no change in confidence from pre to post. Figure 1 visualizes the distribution of the new *confidence change* variable. Most students seemed to have either a decrease or no change in confidence from pre to post, with less than 500 students experiencing an increase in confidence from pre to post.

Figure 1. Confidence Change Distribution



Correlations were calculated across all predictor variables and the new *confidence change* response variable (notated as *Confident* for analysis) and shown in Figure 2. Most variables are positively correlated, with the highest correlation appearing between the *Instructor Quality* items. *Discouraged*, unsurprisingly, is consistently negatively correlated with most items; however, it has a mild positive correlation with *Presentations*, suggesting that students who felt discouraged from continuing in calculus also experienced presentations assigned in class. *Lecture* has generally weak correlations with everything, with the correlations with *WorkTogether* and *Presentations*, being slightly stronger and negatively correlated, suggesting that courses with presentations or group work may have less lecture time. *Confident* does not seem to have strong correlation with any specific items, with the strongest correlations falling with *ProblemSolver* and *Time*.

Figure 2. Variable Correlations



Figures 3-6 show boxplots of *Confident* against four *Instructor Quality* items. For *Listened* and *Applications*, the means seem to stay the same across each *Confident* response option. The means for *AskedQs* and *Time* for students who experienced a decrease in math confidence are lower then for the other two *Confident* options, which have the same means. No confidence change presents the widest range for both *Listened* and *Applications*, while a confidence increase has the widest range for *AskedQs* and *Time*.

Figure 3. AskedQs

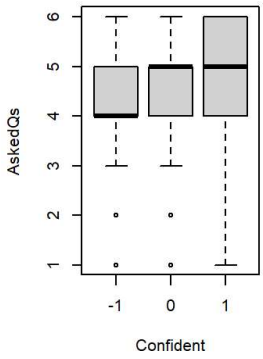


Figure 4. Listened

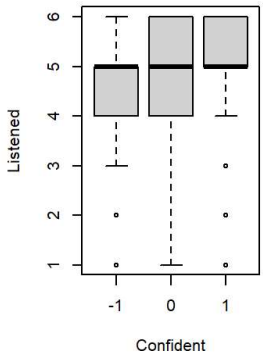


Figure 5. Applications

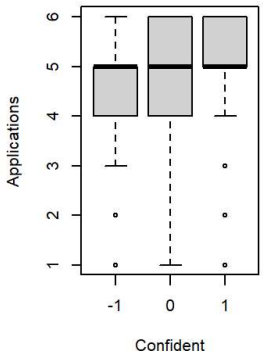
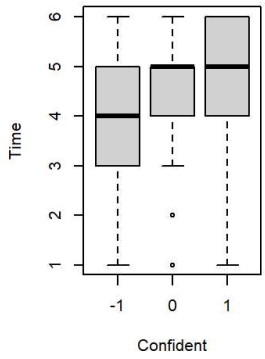


Figure 6. Time



The remaining four *Instructor Quality* items are visualized against *Confident* in boxplots in Figures 7-10. There appears to be no difference in means across *Confident* response options for *Explanations*, *Appointments*, and *Discouraged*, but the average for *ProblemSolver* for confidence decrease is much lower than for the other two response options. For *Appointments*, most response options seemed to fall in the confidence decrease category of *Confident*, as the plot is much wider than the other two options. *Discouraged* seemed to follow the same pattern as *Appointments* in terms of *Confident* distribution. On the other hand, confidence decrease has the smallest distribution for *Explanations*.

Figure 7. ProblemSolver

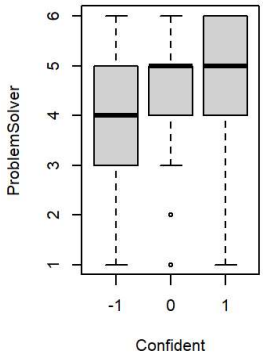


Figure 8. Explanations

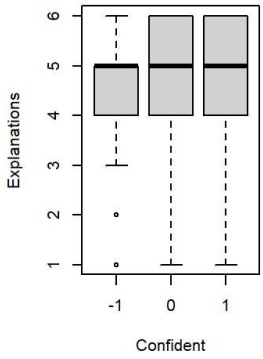


Figure 9. Appointments

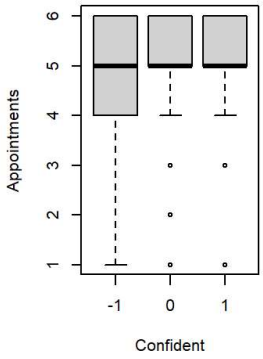
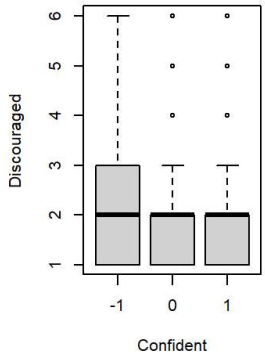


Figure 10. Discouraged



Figures 11-14 show boxplots of *Confident* against four *Pedagogical Practices* items. The means seem to be the same across *SpecificProblems* and *Presentations*. For *SpecificProblems*, the students who experienced an increase in confidence seem to be a much smaller group than the other two *Confident* options, and for *Presentations*, the smallest group seems to be those who have no change in confidence.The distributions are much wider for the *Confident* categories for both *WorkTogether* and *Discussion*. The mean for those with no confidence change seem to have a much smaller average score for *WorkTogether* than for the other two *Confident* categories, while for *Discussion*, students with an increase in confidence seem to have a higher average rating than the other two *Confident* categories.

Figure 11. SpecificProblems

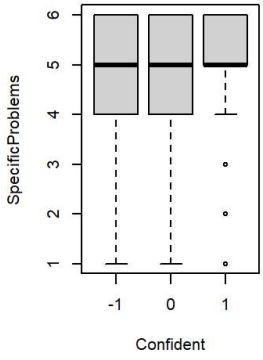


Figure 12. WorkTogether

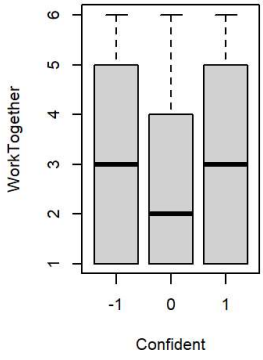


Figure 13. Discussion

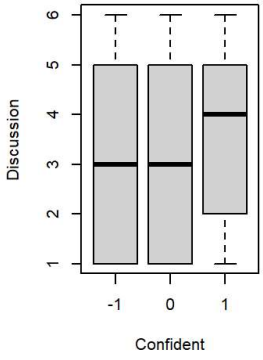
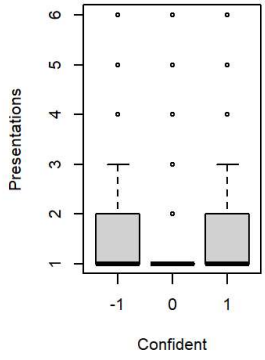


Figure 14. Presentations



The remaining four *Pedagogical Practices* items are visualized against *Confident* in boxplots in Figures 15-18. There appears to be no difference in means across *Confident* response options for any of the variables. For *Lecture*, distributions for each *Confident* category seem to be identical. For *Individually* and *ExplainThinking*, the students with increased confidence seem to present a smaller distribution of responses than the other two *Confident* categories, while for *AskQuestions*, the students who experienced a decrease in confidence experienced the smallest distribution.

Figure 15. Individually

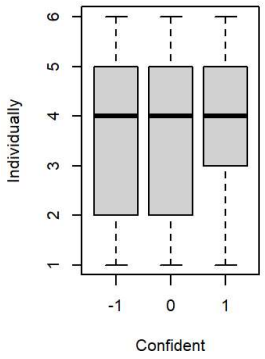


Figure 16. Lecture

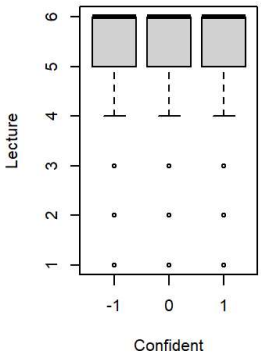


Figure 17. ExplainThinking

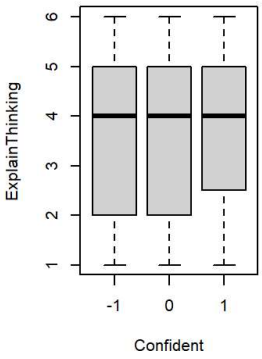
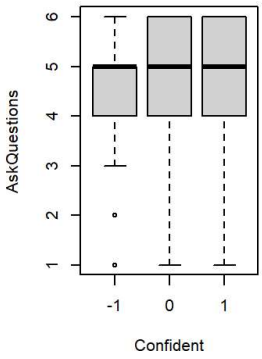


Figure 18. AskQuestion



Methods

The data was split into training and test sets with a 70%-30% split and then divided into three separate sets: *Instructor Quality*, *Pedagogical Practices*, and *Combined Constructs* in order to build separate models for comparison. The training set was used to build the following models: Linear Discriminant Analysi (LDA), Quadratic Discriminant Analysis (QDA), Naive Bayes, Logistic Regression, and k-Nearest-Neighbors (KNN) with 8 different values for k. After training the models, the test data was used to test each model's performance.

1. Linear Discriminant Analysis (LDA)

The first model was developed using linear discriminant analysis (LDA), which is a linear classifier and assumes a Gaussian, or normal, distribution and also assumes all features have the same covariance matrix.

2. Quadratic Disciminant Analysis (QDA)

The second model was developed using quadratic discriminant analysis (QDA), which is similar to LDA but does not assume that each variable has the same covariance matrix.

3. Naive Bayes

The third model was developed using a Naive Bayes classifier, which is a probabilistic model based on Bayes' Theorem that assumes independence of all features.

4. Logistic Regression

The fourth model was developed using logistic regression, which is another classification model that performs similarly to Naive Bayes but is more complex and works better on larger datasets than Naive Bayes, which is more suited to smaller datasets.

5. k-Nearest-Neighbor Algorithm (KNN)

The final 8 models were developed using a k-nearest-neighbors (KNN) classifier, which focuses on examining the k nearest points to each point as a means of classifying points. The models were each developed using the training dataset and several different values of k: 1, 3, 5, 7, 9, 11, 13, and 15.

Results

After testing and training the results, the training and test errors for each data set for each model were compiled for comparison. Then, each data set's models were run through 100 Monte Carlo cross-validation iterations, and the average cross-validated test errors and variances were compared. After selecting the best model for each data set, the results were compared within each set's models using t-tests and Wilcoxon tests to explore potential statistically significant differences between the selected model against the others.

Instructor Quality

Table 4 summarizes the model training and test errors for each model using the *Instructor Quality* data set. The KNN models performed much better than the others, except for KNN15, suggesting that as k increases, the KNN models begin performing similarly to the other four. In terms of training error, QDA had the highest error at 0.4894, which is only marginally higher than the training errors for LDA, Naive Bayes, Logistic Regression, and KNN15. However, the highest test error was yielded by the Naive Bayes model at 0.5259, which was only marginally higher than LDA, QDA, Logistic Regression, and KNN15. Interestingly, KNN1 had the lowest error for both training and test sets, suggesting it may be the best-performing model. However, these results are before performing cross-validation, so it's possible the results, which are ideal with 0.0000 error, are inflated by chance.

Table 4. Instructor Quality Training and Test Errors

Model	Train	Test
LDA	0.4691	0.5066
QDA	0.4894	0.5211
Bayes	0.4744	0.5259
LogReg	0.4691	0.5102
KNN1	0.0000	0.0000
KNN3	0.0468	0.0590
KNN5	0.0584	0.0794
KNN7	0.0685	0.1011
KNN9	0.0980	0.1227
KNN11	0.1047	0.1215
KNN13	0.1163	0.1288
KNN15	0.4691	0.5066

[a] Red indicates highest value. Green indicates lowest value.

Table 5 summarizes the average cross-validated model test errors and variances using the *Instructor Quality* data set. All variances were small, with marginal differences between highest and lowest values. KNN models still performed much better than the other four classification models, though the increase in test error as the value of k increases is not as significant or similar to the other classification models as before cross-validation, suggesting the previous error may have been inflated due to chance. The QDA model still performed the worst on the data with the highest test error of 0.5099, echoing the original training error results. However, KNN3 performed best on the data with the lowest test error 0.0875, which differs from the original test error suggesting KNN1 performed best, further supporting that the original values may have been inflated due to chance.

Table 5. Instructor Quality Average Cross-Validated Test Error and Variances

Model	Error	Variance
LDA	0.4836	0.0002
QDA	0.5099	0.0002
Bayes	0.4965	0.0002
LogReg	0.4830	0.0002
KNN1	0.0955	0.0001
KNN3	0.0875	0.0001
KNN5	0.0901	0.0001
KNN7	0.0965	0.0001
KNN9	0.1055	0.0002
KNN11	0.1123	0.0002
KNN13	0.1196	0.0002
KNN15	0.1276	0.0002

[a] Red indicates highest value. Green indicates lowest value.

After identifying KNN3 as the best model for the *Instructor Quality* data, t-tests and Wilcoxon tests were run to explore differences between the KNN3 results and the rest of the tested models. Table 6 summarizes the statistical test results. There were statistically significant differences between KNN3's performance against all other models using both t-tests and Wilcoxon tests, further validating that KNN3 is the best model for using the *Instructor Quality* data to predict math confidence change.

Table 6. Instructor Quality Statistical Test Comparisons

Model	t-test	Wilcoxon
LDA	0.000	0.0000
QDA	0.000	0.0000
Bayes	0.000	0.0000
LogReg	0.000	0.0000
KNN1	0.000	0.0000
KNN5	0.001	0.0009
KNN7	0.000	0.0000
KNN9	0.000	0.0000
KNN11	0.000	0.0000
KNN13	0.000	0.0000
KNN15	0.000	0.0000

[a] Yellow indicates statistical significance at alpha = 0.05.

Pedagogical Practices

Table 7 summarizes the model training and test errors for each model using the *Pedagogical Practices* data set. Again, the KNN models overall performed better than the other four model types. The Naive Bayes model had the highest training error at 0.5270, though it was only marginally higher than the training errors for LDA, QDA, and Logistic Regression. However, the highest test error was from the QDA model with 0.5628, which was also only marginally higher than the test errors of LDA, Naive Bayes, and Logistic Regression. Interestingly, KNN1 had the lowest error for both training and test sets, suggesting it may be the best-performing model. However, these results are before performing cross-validation, so it's possible the results, which are ideal with 0.0000 error, are inflated by chance.

Table 7. Pedagogical Practices Training and Test Errors

Model	Train	Test
LDA	0.5202	0.5142
QDA	0.5216	0.5628
Bayes	0.5270	0.5415
LogReg	0.5202	0.5107
KNN1	0.0000	0.0000
KNN3	0.1034	0.1398
KNN5	0.1287	0.1742
KNN7	0.1510	0.2026
KNN9	0.1714	0.2050
KNN11	0.1909	0.2156
KNN13	0.2006	0.2263
KNN15	0.2060	0.2334

[a] Red indicates highest value. Green indicates lowest value.

Table 8 summarizes the average cross-validated model test errors and variances using the *Pedagogical Practices* data set. All variances were small, with marginal differences between highest and lowest values. KNN models still performed much better than the other four classification models and were not much different from the initial errors, suggesting chance wasn't as much an influence on the errors. Rather than the Naive Bates, the QDA model performed worst on the data with the highest test error of 0.5402. However, KNN1 still performed best on the data with the lowest test error 0.1973, which differs from the original test error suggesting that, while the initial results were accurate in model selection, the error was a bit inflated by chance.

Table 8. Pedagogical Practices Average Cross-Validated Test Error and Variances

Model	Error	Variance
LDA	0.5287	0.0002
QDA	0.5402	0.0002
Bayes	0.5381	0.0002
LogReg	0.5289	0.0002
KNN1	0.1973	0.0001
KNN3	0.1981	0.0001
KNN5	0.2065	0.0002
KNN7	0.2184	0.0002
KNN9	0.2271	0.0001
KNN11	0.2334	0.0002
KNN13	0.2366	0.0002
KNN15	0.2385	0.0002

[a] Red indicates highest value. Green indicates lowest value.

After identifying KNN1 as the best model for the *Pedagogical Practices* data, t-tests and Wilcoxon tests were run to explore differences between the KNN1 results and the rest of the tested models. Table 9 summarizes the statistical test results. There were statistically significant differences between KNN1's performance against all other models using both t-tests and Wilcoxon tests, except for the t-test p-value against KNN3. While there was not a statistically significant difference in the mean test error of KNN1 and KNN3, the results still validate that KNN1 is the best model for using the *Pedagogical Practices* data to predict math confidence change.

Table 9. Pedagogical Practices Statistical Test Comparisons

Model	t-test	Wilcoxon
LDA	0.0000	0.0000
QDA	0.0000	0.0000
Bayes	0.0000	0.0000
LogReg	0.0000	0.0000
KNN3	0.4284	0.0000
KNN5	0.0000	0.0009
KNN7	0.0000	0.0000
KNN9	0.0000	0.0000
KNN11	0.0000	0.0000
KNN13	0.0000	0.0000
KNN15	0.0000	0.0000

[a] Yellow indicates statistical significance at alpha = 0.05.

Combined Constructs

Table 10 summarizes the model training and test errors for each model using the *Combined Constructs* data set. Again, the KNN models overall performed better than the other four model types. The Naive Bayes model had the highest training error at 0.5075, though it was only marginally higher than the training errors for LDA, QDA, and Logistic Regression. Similarly, it also has the highest test error of 0.5332, which was also only marginally higher than the test errors of LDA, QDA, and Logistic Regression. Interestingly, KNN1 had the lowest error for both training and test sets, just as with the two previous data sets, suggesting it may be the best-performing model. However, these results are before performing cross-validation, so it's possible the results, which are ideal with 0.0000 error, are inflated by chance.

Table 10. Combined Constructs Training and Test Errors

Model	Train	Test
LDA	0.4687	0.4882
QDA	0.4827	0.5036
Bayes	0.5075	0.5332
LogReg	0.4662	0.4882
KNN1	0.0000	0.0000
KNN3	0.1996	0.2026
KNN5	0.2326	0.2358
KNN7	0.2598	0.2488
KNN9	0.2627	0.2701
KNN11	0.2657	0.2820
KNN13	0.2700	0.2713
KNN15	0.2826	0.2784

[a] Red indicates highest value. Green indicates lowest value.

Table 11 summarizes the average cross-validated model test errors and variances using the *Combined Constructs* data set. All variances were small, with marginal differences between highest and lowest values. KNN models still performed much better than the other four classification models and were not much different from the initial errors, suggesting chance wasn't as much an influence on the errors. Rather than the Naive Bates, the QDA model performed worst on the data with the highest test error of 0.5259. However, KNN15 performed best on the data with the lowest test error 0.3350, even with a marginally higher variance.

Table 11. Combined Constructs Average Cross-Validated Test Error and Variances

Model	Error	Variance
LDA	0.4834	0.0002
QDA	0.5226	0.0002
Bayes	0.5259	0.0003
LogReg	0.4838	0.0002
KNN1	0.3697	0.0002
KNN3	0.3542	0.0003
KNN5	0.3489	0.0002
KNN7	0.3444	0.0002
KNN9	0.3387	0.0002
KNN11	0.3369	0.0002
KNN13	0.3351	0.0002
KNN15	0.3350	0.0003

[a] Red indicates highest value. Green indicates lowest value.

After identifying KNN15 as the best model for the *Combined Constructs* data, t-tests and Wilcoxon tests were run to explore differences between the KNN15 results and the rest of the tested models. Table 12 summarizes the statistical test results. There were statistically significant differences between KNN15's performance against all other models using both t-tests and Wilcoxon tests, exceptagainst KNN11 and KNN13, suggesting that even with the smallest test error, the differences between KNN15, KNN11, and KNN13 are negligible. While there was not a statistically significant difference in the mean test error of KNN15, KNN11, and KNN3, the results still validate that KNN15 is the best model for using the *Combined Constructs* data to predict math confidence change.

Table 12. Combined Constructs Statistical Test Comparisons

Model	t-test	Wilcoxon
LDA	0.0000	0.0000
QDA	0.0000	0.0000
Bayes	0.0000	0.0000
LogReg	0.0000	0.0000
KNN1	0.0000	0.0000
KNN3	0.0000	0.0000
KNN5	0.0000	0.0000
KNN7	0.0000	0.0000
KNN9	0.0028	0.0071
KNN11	0.1089	0.2302
KNN13	0.9567	0.9162

[a] Yellow indicates statistical significance at alpha = 0.05.

Comparison

After training, testing, and validating the models for each of the data sets, the average cross-validated test errors for each model across all three data sets were compared and summarized in Table 13. QDA performed worst on the Instructor Quality and Pedagogical Practices data, while Bayes performed worst on the Combined Constructs data. Of all three data sets, *Pedagogical Pratices* yielded the highest test error of all models with its QDA model with an error of 0.5402. *Instructor Quality* models consistently performed better across all model types for all data sets and had the lowest overall test error with model KNN3's test error of 0.0875, suggesting the *Instructor Quality* data may be better for predicting change in math confidence.

Table 13. Model Test Error Comparisons

Model	Instructor	Pedagogical	Combined
LDA	0.4836	0.5287	0.4834
QDA	0.5099	0.5402	0.5226
Bayes	0.4965	0.5381	0.5259
LogReg	0.4830	0.5289	0.4838
KNN1	0.0955	0.1973	0.3697
KNN3	0.0875	0.1981	0.3542
KNN5	0.0901	0.2065	0.3489
KNN7	0.0965	0.2184	0.3444
KNN9	0.1055	0.2271	0.3387
KNN11	0.1123	0.2334	0.3369
KNN13	0.1196	0.2366	0.3351
KNN15	0.1276	0.2385	0.3350

[a] Red indicates highest value. Green indicates lowest value.

After identifying the *Instructor Quality* KNN3 model as the best model overall, t-tests and Wilcoxon tests were run to explore differences between the *Instructor Quality* KNN3 model results, *Pedagogical Practices* KNN1 model results, and *Combined Constructs* KNN15 model esults, as these were the best-performing models for each data set. Table 14 summarizes the statistical test results. There were statistically significant differences between the *Instructor QUality* KNN3 model's performance against both other models using both t-tests and Wilcoxon tests, which suggests that *Instructor Quality* data is the best for predicting post-Calculus I math confidence change using the KNN3 classification model.

Table 14. Selected Model Statistical Test Comparisons

Model	t-test	Wilcoxon
Pedagogical	0.0000	0.0000
Combined	0.0000	0.0000

[a] Yellow indicates statistical significance at alpha = 0.05.

Conclusion

Using a response variable created from student self-rated math confidence based on rating how strongly they agree or disagree with the statement "I am confident in my mathematical abilities" both at the beginning and end of their college/university Calculus I courses, it was possible to accurately and effectively predict change in confidence from pre- to post-Calculus I using *Instructor Quality* data where students rated the overall perceived quality of their instructors. Across three separate data sets (*Instructor Quality*, *Pedagogical Practices*, and a *Combined Constructs* set made up of the two), twelve models were trained and tested for each, exploring the best-possible classification method for predicting math confidence change. Overall, KNN models performed best of all model types tested for any data set, and the best-performing model across all 36 tested was a KNN model with k = 3 using the *Instructor Quality* data, with an average cross-validated test error of 0.0875, which is very low and suggests that the model performs well on the data in predicting the created math confidence change response variable. The results of this analysis give insight into the student-instructor relationship, as students having a relationship with their instructors and perceiving them as high-quality and providing them space and time to ask questions, seek support outside of class, and fully understand the classroom content has a higher predictive power in students experiencing a change in their math confidence than simply exploring instructor student-centered pedagogical practices or even a a combination of pedagogy and instructor quality.

It is important to note the limitations of this analysis. The original data set had 37 columns with 13,409 student responses, but due to only using the two specified survey constructs and removing any observations containing NA, the data was reduced down to 18 columns and 2,903 observations, which is approximately 22% of the original. While 22% is an acceptable sample size to be representative of the whole data set, most students were removed from the data set, which could cause a significant bias to become present in the analysis. Also, since the data was anonymized and had most identifiable and demographic information, such as year in school, institution, and race/ethnicity, the analysis is limited in terms of generalization, as without this information, it's impossible to disaggregate the data to explore generalization for future populations. During this analysis, too, variable selection was not performed, as the focus was on the whole constructs, rather than subsets, which allows the models to possibly contain extra noise or chance due to potentially-unnecessary variables/data included in the training and testing. The analysis also did not explore other measures of model performance outside of average cross-validated test error and variance. It's possible that results could vary greatly when using other performance metrics.

In the future, it would be important to continue the research and try to rectify some of the limitations, such attempting a new round of data collection to obtain demographic and institutional data for further analysis and generalization. It could also be beneficial to explore other similar data sets that measure math confidence alongside different survey constructs and compare to these results in order to further validate results. Math confidence change in this analysis was created as a categorical variable in order to test different classification models; however, it could be interesting to compare results when predicting a continuous math confidence change variable with other types of models than just classification. There are also other classification methods, such as random forest, that could be used to further compare against the *Instructor Quality* KNN3 model performance. As mention, the original data set also included many other variables, such as SAT and ACT scores, gender identity, and primary field of study, which could be interesting to explore as additional predictors or grouping variables within the analysis. Lastly, the data set also included a pre-post question asked both before and after Calculus I about intention to persist, "Do you intent to take Calculus II?", which could be interesting to do as a repeat analysis using the same methods to see if the results are similar for both pre-post results with both math confidence change and intention to persist in math.

References

Audet, Lauren. 2018. “Encouraging Mathematical Confidence.” *Encouraging Mathematical Confidence*. https://blog.heinemann.com/thinking-together-encouraging-mathematical-confidence (https://blog.heinemann.com/thinking-together-encouraging-mathematical-confidence).

Charan, J., and T. Biswas. 2013. “How to Calculate Sample Size for Different Study Designs in Medical Research?” *Indian Journal of Psychological Medicine* 35: 121–26. https://doi.org/https://doi.org/10.4103/0253-7176.116232 (https://doi.org/10.4103/0253-7176.116232).

Christensen, Rhonda, and Gerald Knezek. 2020. “Indicators of Middle School Students’ Mathematics Enjoyment and Confidence.” *School Science and Mathematics* 120 (8): 491–503. https://doi.org/10.1111/ssm.12439 (https://doi.org/10.1111/ssm.12439).

Ellis, J., B. K. Fisdick, and C. Rasmussen. 2016. “Women 1.5 Times More Likely to Leave STEM Pipeline After Calculus Compared to Men: Lack of Mathematical Confidence a Potential Culprit.” *PLOS ONE* 11. https://doi.org/https://doi.org/10.1371/journal.pone.0157447 (https://doi.org/10.1371/journal.pone.0157447).

Khasawneh, Eihab, Cameron Gosling, and Brett Williams. 2021. “What Impact Does Maths Anxiety Have on University Students?” *BMC Psychology* 9 (1). https://doi.org/10.1186/s40359-021-00537-2 (https://doi.org/10.1186/s40359-021-00537-2).

One, PLOS. 2024. “PLOS One.” *Journal Information*. PLOS One. https://journals.plos.org/plosone/s/journalinformation (https://journals.plos.org/plosone/s/journalinformation).

Pierce, Robyn, and Kaye Stacey. 2004. “A Framework for Monitoring Progress and Planning Teaching Towards the Effective Use of Computer Algebra Systems.” *International Journal of Computers for Mathematical Learning* 9 (1): 59–93. https://doi.org/10.1023/b:ijco.0000038246.98119.14 (https://doi.org/10.1023/b:ijco.0000038246.98119.14).

West, Mary. 2022. “Math Anxiety: Definition, Symptoms, Causes, and Tips.” Edited by Joslyn Jelinek. *Medical News Today*. MediLexicon International. https://www.medicalnewstoday.com/articles/math-anxiety-definition-symptoms-causes-and-tips#:~:text=The%20term%20%E2%80%9Cmath%20anxiety%E2%80%9D%20describes,doing%20math%20overwhelms%20working%20memory. (https://www.medicalnewstoday.com/articles/math-anxiety-definition-symptoms-causes-and-tips#:~:text=The%20term%20%E2%80%9Cmath%20anxiety%E2%80%9D%20describes,doing%20math%20overwhelms%20working%20memory.

Zakariya, Yusuf F. 2022. “Improving Students’ Mathematics Self-Efficacy: A Systematic Review of Intervention Studies.” *Frontiers in Psychology* 13 (September). https://doi.org/10.3389/fpsyg.2022.986622 (https://doi.org/10.3389/fpsyg.2022.986622).

Appendix

```
set.seed(7406)

#read data
file <- "data.xlsx"
dat <- read_excel(file, na = "NA")

#isolate variables and remove NAs
dat_conf <- dat[,c(9, 21:37)] %>%
  na.omit()

# create response variable
dat_conf$Confident <- dat_conf$Q6Post_Confident - dat_conf$Q29_Confident
dat_conf$Confident <- with(dat_conf, ifelse(Q6Post_Confident > Q29_Confident, 1,                                                    i
felse(Q6Post_Confident < Q29_Confident, -1, 0)))

#remove pre-post values
df <- dat_conf[,-c(1:2)]

#split into train and test sets
#https://www.statology.org/train-test-split-r/
sample <- sample(c(TRUE, FALSE), nrow(df), replace=TRUE, prob=c(0.7,0.3))
df_train  <- df[sample, ]
df_test   <- df[!sample, ]

#empty variables to hold train and test errors
TrainErr <- NULL;
TestErr  <- NULL;

TrainErr2 <- NULL;
TestErr2  <- NULL;

TrainErr3 <- NULL;
TestErr3  <- NULL;


### Method 1: LDA
##instructor quality
mod1 <- lda(df_train[,1:8], df_train$Confident);
pred1 <- predict(mod1,df_train[,c(1:8)])$class;
TrainErr <- c(TrainErr, mean( pred1  != df_train$Confident));
pred1test <- predict(mod1,df_test[,c(1:8)])$class;
TestErr <- c(TestErr,mean(pred1test != df_test$Confident));
##pedagogical practices
mod1 <- lda(df_train[,9:16], df_train$Confident);
pred1 <- predict(mod1,df_train[,c(9:16)])$class;
TrainErr2 <- c(TrainErr2, mean( pred1  != df_train$Confident));
pred1test <- predict(mod1,df_test[,c(9:16)])$class;
TestErr2 <- c(TestErr2,mean(pred1test != df_test$Confident));
##combined
mod1 <- lda(df_train[,1:16], df_train$Confident);
pred1 <- predict(mod1,df_train[,c(1:16)])$class;
TrainErr3 <- c(TrainErr3, mean( pred1  != df_train$Confident));
pred1test <- predict(mod1,df_test[,c(1:16)])$class;
TestErr3 <- c(TestErr3,mean(pred1test != df_test$Confident));

### Method 2: QDA
##instructor quality
mod2 <- qda(df_train[,c(1:8)], df_train$Confident)
pred2 <- predict(mod2,df_train[,c(1:8)])$class
TrainErr <- c(TrainErr, mean( pred2!= df_train$Confident))
TestErr <- c(TestErr, mean(predict(mod2,df_test[,c(1:8)])$class != df_test$Confident))
##pedagogical practices
mod2 <- qda(df_train[,c(9:16)], df_train$Confident)
```

```
pred2 <- predict(mod2,df_train[,c(9:16)])$class
TrainErr2 <- c(TrainErr2, mean( pred2!= df_train$Confident))
TestErr2 <- c(TestErr2, mean(predict(mod2,df_test[,c(9:16)])$class != df_test$Confident))
#combined
mod2 <- qda(df_train[,c(1:16)], df_train$Confident)
pred2 <- predict(mod2,df_train[,c(1:16)])$class
TrainErr3 <- c(TrainErr3, mean( pred2!= df_train$Confident))
TestErr3 <- c(TestErr3, mean(predict(mod2,df_test[,c(1:16)])$class != df_test$Confident))

### Method 3: Naive Bayes
##instructor quality
mod3 <- naiveBayes(df_train[,c(1:8)], df_train$Confident)
pred3 <- predict(mod3, df_train[,c(1:8)]);
TrainErr <- c(TrainErr, mean( pred3 != df_train$Confident))
TestErr <- c(TestErr,  mean( predict(mod3,df_test[,c(1:8)]) != df_test$Confident))
##pedagogical practices
mod3 <- naiveBayes(df_train[,c(9:16)], df_train$Confident)
pred3 <- predict(mod3, df_train[,c(9:16)]);
TrainErr2 <- c(TrainErr2, mean( pred3 != df_train$Confident))
TestErr2 <- c(TestErr2,  mean( predict(mod3,df_test[,c(9:16)]) != df_test$Confident))
##combined
mod3 <- naiveBayes(df_train[,c(1:16)], df_train$Confident)
pred3 <- predict(mod3, df_train[,c(1:16)]);
TrainErr3 <- c(TrainErr3, mean( pred3 != df_train$Confident))
TestErr3 <- c(TestErr3,  mean( predict(mod3,df_test[,c(1:16)]) != df_test$Confident))

### Method 4: (multinomial) logisitic regression
##instructor quality
mod4 <- multinom(Confident ~., data=df_train[,c(1:8, 17)], trace = F)
TrainErr <- c(TrainErr, mean( predict(mod4, df_train[,c(1:8)])  != df_train$Confident))
TestErr <- c(TestErr, mean( predict(mod4,df_test[,c(1:8)]) != df_test$Confident))
##pedagogical practices
mod4 <- multinom(Confident ~., data=df_train[,c(9:17)], trace = F)
TrainErr2 <- c(TrainErr2, mean( predict(mod4, df_train[,c(9:16)])  != df_train$Confident))
TestErr2 <- c(TestErr2, mean( predict(mod4,df_test[,c(9:17)]) != df_test$Confident))
##combined
mod4 <- multinom(Confident ~., data=df_train, trace = F)
TrainErr3 <- c(TrainErr3, mean( predict(mod4, df_train[,c(1:16)])  != df_train$Confident))
TestErr3 <- c(TestErr3, mean( predict(mod4,df_test[,c(1:16)]) != df_test$Confident))

### Method 5: KNN
##instructor quality
kk <- cbind(1,3,5,7,9,11,13,15)
cverror2 <- NULL
cverror3 <- NULL
for (i in 1:length(kk)){
  xnew2 <- df_train[,c(1:8, 17)];
  ypred2.train <- knn(df_train[,c(1:8,17)], xnew2, df_train$Confident, k=kk[i]);
  temperror2<- mean(ypred2.train != df_train$Confident)
  cverror2 <- cbind(cverror2,temperror2)
  #Test error
  xnew3 <- df_test[,c(1:8, 17)];
  rbind(TrainErr, cverror2)
  ypred3.test <- knn(df_test[,c(1:8,17)], xnew3, df_test$Confident, k=kk[i]);
  temperror3 <- mean(ypred3.test != df_test$Confident)
  cverror3 <- cbind(cverror3,temperror3)
}
TrainErr <- c(TrainErr, cverror2)
TestErr <- c(TestErr, cverror3)
##pedagogical practices
kk <- cbind(1,3,5,7,9,11,13,15)
cverror2 <- NULL
cverror3 <- NULL
for (i in 1:length(kk)){
  xnew2 <- df_train[,c(9:17)];
  ypred2.train <- knn(df_train[,c(9:17)], xnew2, df_train$Confident, k=kk[i]);
  temperror2<- mean(ypred2.train != df_train$Confident)
```

```
  cverror2 <- cbind(cverror2,temperror2)
  #Test error
  xnew3 <- df_test[,c(9:17)];
  rbind(TrainErr, cverror2)
  ypred3.test <- knn(df_test[,c(9:17)], xnew3, df_test$Confident, k=kk[i]);
  temperror3 <- mean(ypred3.test != df_test$Confident)
  cverror3 <- cbind(cverror3,temperror3)
}
TrainErr2 <- c(TrainErr2, cverror2)
TestErr2 <- c(TestErr2, cverror3)
##combined
kk <- cbind(1,3,5,7,9,11,13,15)
cverror2 <- NULL
cverror3 <- NULL
for (i in 1:length(kk)){
  xnew2 <- df_train;
  ypred2.train <- knn(df_train, xnew2, df_train$Confident, k=kk[i]);
  temperror2<- mean(ypred2.train != df_train$Confident)
  cverror2 <- cbind(cverror2,temperror2)
  #Test error
  xnew3 <- df_test;
  rbind(TrainErr, cverror2)
  ypred3.test <- knn(df_test, xnew3, df_test$Confident, k=kk[i]);
  temperror3 <- mean(ypred3.test != df_test$Confident)
  cverror3 <- cbind(cverror3,temperror3)
}
TrainErr3 <- c(TrainErr3, cverror2)
TestErr3 <- c(TestErr3, cverror3)

### Cross-Validation
##instructor quality
n1 = dim(df_train)[1]; # training set sample size
n2= dim(df_train)[1]; # testing set sample size
n = dim(df)[1]; ## the total sample size
set.seed(7406)
B= 100; ### number of loops
TEALL = NULL ### Final TE values
for (b in 1:B){}
  ## randomly select n1 observations as a new training subset in each loop
  flag <- sort(sample(1:n, n1));
  df_traintemp <- df[flag,]; ## temp training set for CV
  df_testtemp <- df[-flag,]; ## temp testing set for CV
  ##knn
  kk <- cbind(1,3,5,7,9,11,13,15)
  cverror3 <- NULL
  for (i in 1:length(kk)){
    xnew3 <- df_testtemp[,c(1:8,17)];
    ypred3.test <- knn(df_traintemp[,c(1:8,17)], xnew3, df_traintemp$Confident, k=kk[i]);
    temperror<- mean(ypred3.test != df_testtemp$Confident)
    cverror3 <- cbind(cverror3,temperror)
  }

  ### Method 1: LDA
  mod1 <- lda(df_traintemp[,c(1:8)], df_traintemp$Confident);
  pred1test <- predict(mod1,df_testtemp[,c(1:8)])$class;
  t1 <- mean(pred1test != df_testtemp$Confident)

  ## Method 2: QDA
  mod2 <- qda(df_traintemp[,c(1:8)], df_traintemp$Confident)
  pred2test <- predict(mod2,df_testtemp[,c(1:8)])$class
  t2 <-  mean(pred2test != df_testtemp$Confident)

  ## Method 3: Naive Bayes
  mod3 <- naiveBayes(df_traintemp[,c(1:8)], df_traintemp[,17])
  pred3test <- predict(mod3,df_testtemp[,c(1:8)])
  t3 <- mean(pred3test != df_testtemp$Confident)
```

```
  ### Method 4: (multinomial) logisitic regression)
  mod4 <- multinom(Confident ~., data=df_traintemp[,c(1:8,17)], trace = F)
  pred4test <- predict(mod4,df_testtemp[,c(1:8)])
  t4 <- mean(pred4test != df_testtemp$Confident)

  TEALL = rbind(TEALL, cbind(t1, t2, t3, t4, cverror3));

}

colnames(TEALL) <- c("LDA", "QDA","Bayes", "LogReg", "KNN1", "KNN3", "KNN5", "KNN7", "KNN9", "KNN11", "KNN13", "KNN15")
a <- apply(TEALL, 2, mean);
b <- apply(TEALL, 2, var);
a <- round(a, 4)
b <- round(b, 4)
n <- c("LDA", "QDA","Bayes", "LogReg", "KNN1", "KNN3", "KNN5", "KNN7", "KNN9", "KNN11", "KNN13", "KNN15")
c <- cbind(n,a,b)
c <- data.frame(c, row.names = NULL) %>%
  transform(c, a = as.numeric(a)) %>%
  transform(c, b = as.numeric(b)) %>%
  mutate(across(where(is.numeric), ~ round(., 4)))
c_ins <- c[, 1:3]

T1=t.test(TEALL[,6],TEALL[,1],paired=T)
T2=t.test(TEALL[,6],TEALL[,2],paired=T)
T3=t.test(TEALL[,6],TEALL[,3],paired=T)
T4=t.test(TEALL[,6],TEALL[,4],paired=T)
T5=t.test(TEALL[,6],TEALL[,5],paired=T)
T6=t.test(TEALL[,6],TEALL[,7],paired=T)
T7=t.test(TEALL[,6],TEALL[,8],paired=T)
T8=t.test(TEALL[,6],TEALL[,9],paired=T)
T9=t.test(TEALL[,6],TEALL[,10],paired=T)
T10=t.test(TEALL[,6],TEALL[,11],paired=T)
T11=t.test(TEALL[,6],TEALL[,12],paired=T)

W1=wilcox.test(TEALL[,6],TEALL[,1],paired=T)
W2=wilcox.test(TEALL[,6],TEALL[,2],paired=T)
W3=wilcox.test(TEALL[,6],TEALL[,3],paired=T)
W4=wilcox.test(TEALL[,6],TEALL[,4],paired=T)
W5=wilcox.test(TEALL[,6],TEALL[,5],paired=T)
W6=wilcox.test(TEALL[,6],TEALL[,7],paired=T)
W7=wilcox.test(TEALL[,6],TEALL[,8],paired=T)
W8=wilcox.test(TEALL[,6],TEALL[,9],paired=T)
W9=wilcox.test(TEALL[,6],TEALL[,10],paired=T)
W10=wilcox.test(TEALL[,6],TEALL[,11],paired=T)
W11=wilcox.test(TEALL[,6],TEALL[,12],paired=T)

t <- c(T1$p.value,T2$p.value,T3$p.value,T4$p.value,T5$p.value,T6$p.value, T7$p.value, T8$p.value, T9$p.value, T10$p.value, T
11$p.value)
w <- c(W1$p.value,W2$p.value,W3$p.value,W4$p.value,W5$p.value,W6$p.value, W7$p.value, W8$p.value, W9$p.value, W10$p.value, W
11$p.value)
n <- c("1. LDA", "2. QDA","3. Bayes", "4. LogReg", "5. KNN1", "7. KNN5", "8. KNN7", "9. KNN9", "10. KNN11", "11. KNN13", "1
2. KNN15")

a <- data.frame(cbind(as.numeric(t),as.numeric(w)))

format(round(a, 4), nsmall=4)

b <- cbind(n, a)

### Cross-Validation
##pedagogical practices
n1 = dim(df_train)[1]; # training set sample size
n2= dim(df_train)[1]; # testing set sample size
n = dim(df)[1]; ## the total sample size
set.seed(7406)
B= 100; ### number of loops
TEALL2 = NULL ### Final TE values
```

```
for (b in 1:B){
  ### randomly select n1 observations as a new training subset in each loop
  flag <- sort(sample(1:n, n1));
  df_traintemp <- df[flag,]; ## temp training set for CV
  df_testtemp <- df[-flag,]; ## temp testing set for CV
  ##knn
  kk <- cbind(1,3,5,7,9,11,13,15)
  cverror3 <- NULL
  for (i in 1:length(kk)){
    xnew3 <- df_testtemp[,c(9:17)];
    ypred3.test <- knn(df_traintemp[,c(9:17)], xnew3, df_traintemp$Confident, k=kk[i]);
    temperror<- mean(ypred3.test != df_testtemp$Confident)
    cverror3 <- cbind(cverror3,temperror)
  }

  ### Method 1: LDA
  mod1 <- lda(df_traintemp[,c(9:16)], df_traintemp$Confident);
  pred1test <- predict(mod1,df_testtemp[,c(9:16)])$class;
  t1 <- mean(pred1test != df_testtemp$Confident)

  ## Method 2: QDA
  mod2 <- qda(df_traintemp[,c(9:16)], df_traintemp$Confident)
  pred2test <- predict(mod2,df_testtemp[,c(9:16)])$class
  t2 <-  mean(pred2test != df_testtemp$Confident)

  ## Method 3: Naive Bayes
  mod3 <- naiveBayes(df_traintemp[,c(9:16)], df_traintemp[,17])
  pred3test <- predict(mod3,df_testtemp[,c(9:16)])
  t3 <- mean(pred3test != df_testtemp$Confident)

  ### Method 4: (multinomial) logisitic regression)
  mod4 <- multinom(Confident ~., data=df_traintemp[,c(9:17)], trace = F)
  pred4test <- predict(mod4,df_testtemp[,c(9:16)])
  t4 <- mean(pred4test != df_testtemp$Confident)

  TEALL2 = rbind(TEALL2, cbind(t1, t2, t3, t4, cverror3));

}

colnames(TEALL2) <- c("LDA", "QDA","Bayes", "LogReg", "KNN1", "KNN3", "KNN5", "KNN7", "KNN9", "KNN11", "KNN13", "KNN15")
a <- apply(TEALL2, 2, mean);
b <- apply(TEALL2, 2, var);
a <- round(a, 4)
b <- round(b, 4)
n <- c("LDA", "QDA","Bayes", "LogReg", "KNN1", "KNN3", "KNN5", "KNN7", "KNN9", "KNN11", "KNN13", "KNN15")
c <- cbind(n,a,b)
c <- data.frame(c, row.names = NULL) %>%
  transform(c, a = as.numeric(a)) %>%
  transform(c, b = as.numeric(b)) %>%
  mutate(across(where(is.numeric), ~ round(., 4)))
c_ped <- c[, 1:3]

T1=t.test(TEALL2[,6],TEALL2[,1],paired=T)
T2=t.test(TEALL2[,6],TEALL2[,2],paired=T)
T3=t.test(TEALL2[,6],TEALL2[,3],paired=T)
T4=t.test(TEALL2[,6],TEALL2[,4],paired=T)
T5=t.test(TEALL2[,6],TEALL2[,5],paired=T)
T6=t.test(TEALL2[,6],TEALL2[,7],paired=T)
T7=t.test(TEALL2[,6],TEALL2[,8],paired=T)
T8=t.test(TEALL2[,6],TEALL2[,9],paired=T)
T9=t.test(TEALL2[,6],TEALL2[,10],paired=T)
T10=t.test(TEALL2[,6],TEALL2[,11],paired=T)
T11=t.test(TEALL2[,6],TEALL2[,12],paired=T)

W1=wilcox.test(TEALL[,6],TEALL[,1],paired=T)
W2=wilcox.test(TEALL[,6],TEALL[,2],paired=T)
W3=wilcox.test(TEALL[,6],TEALL[,3],paired=T)
```

```
W4=wilcox.test(TEALL[,6],TEALL[,4],paired=T)
W5=wilcox.test(TEALL[,6],TEALL[,5],paired=T)
W6=wilcox.test(TEALL[,6],TEALL[,7],paired=T)
W7=wilcox.test(TEALL[,6],TEALL[,8],paired=T)
W8=wilcox.test(TEALL[,6],TEALL[,9],paired=T)
W9=wilcox.test(TEALL[,6],TEALL[,10],paired=T)
W10=wilcox.test(TEALL[,6],TEALL[,11],paired=T)
W11=wilcox.test(TEALL[,6],TEALL[,12],paired=T)

t <- c(T1$p.value,T2$p.value,T3$p.value,T4$p.value,T5$p.value,T6$p.value, T7$p.value, T8$p.value, T9$p.value, T10$p.value, T
11$p.value)
w <- c(W1$p.value,W2$p.value,W3$p.value,W4$p.value,W5$p.value,W6$p.value, W7$p.value, W8$p.value, W9$p.value, W10$p.value, W
11$p.value)
n <- c("1. LDA", "2. QDA","3. Bayes", "4. LogReg", "6. KNN3", "7. KNN5", "8. KNN7", "9. KNN9", "10. KNN11", "11. KNN13", "1
2. KNN15")

a <- data.frame(cbind(as.numeric(t),as.numeric(w)))

format(round(a, 4), nsmall=4)

b <- cbind(n, a)

### Cross-Validation
##combined
n1 = dim(df_train)[1]; # training set sample size
n2= dim(df_train)[1]; # testing set sample size
n = dim(df)[1]; ## the total sample size
set.seed(7406)
B= 100; ### number of loops
TEALL3 = NULL ### Final TE values
for (b in 1:B){
  ### randomly select n1 observations as a new training subset in each loop
  flag <- sort(sample(1:n, n1));
  df_traintemp <- df[flag,]; ## temp training set for CV
  df_testtemp <- df[-flag,]; ## temp testing set for CV
  ##knn
  kk <- cbind(1,3,5,7,9,11,13,15)
  cverror3 <- NULL
  for (i in 1:length(kk)){
    xnew3 <- df_testtemp;
    ypred3.test <- knn(df_traintemp, xnew3, df_traintemp$Confident, k=kk[i]);
    temperror<- mean(ypred3.test != df_testtemp$Confident)
    cverror3 <- cbind(cverror3,temperror)
  }

  ### Method 1: LDA
  mod1 <- lda(df_traintemp[,c(1:16)], df_traintemp$Confident);
  pred1test <- predict(mod1,df_testtemp[,c(1:16)])$class;
  t1 <- mean(pred1test != df_testtemp$Confident)

  ## Method 2: QDA
  mod2 <- qda(df_traintemp[,c(1:16)], df_traintemp$Confident)
  pred2test <- predict(mod2,df_testtemp[,c(1:16)])$class
  t2 <-  mean(pred2test != df_testtemp$Confident)

  ## Method 3: Naive Bayes
  mod3 <- naiveBayes(df_traintemp[,c(1:16)], df_traintemp[,17])
  pred3test <- predict(mod3,df_testtemp[,c(1:16)])
  t3 <- mean(pred3test != df_testtemp$Confident)

  ### Method 4: (multinomial) logisitic regression)
  mod4 <- multinom(Confident ~., data=df_traintemp[,c(1:17)], trace = F)
  pred4test <- predict(mod4,df_testtemp[,c(1:16)])
  t4 <- mean(pred4test != df_testtemp$Confident)

  TEALL3 = rbind(TEALL3, cbind(t1, t2, t3, t4, cverror3));
```

```
}

colnames(TEALL3) <- c("LDA", "QDA","Bayes", "LogReg", "KNN1", "KNN3", "KNN5", "KNN7", "KNN9", "KNN11", "KNN13", "KNN15")
a <- apply(TEALL3, 2, mean);
b <- apply(TEALL3, 2, var);
a <- round(a, 4)
b <- round(b, 4)
n <- c("LDA", "QDA","Bayes", "LogReg", "KNN1", "KNN3", "KNN5", "KNN7", "KNN9", "KNN11", "KNN13", "KNN15")
c <- cbind(n,a,b)
c <- data.frame(c, row.names = NULL) %>%
  transform(c, a = as.numeric(a)) %>%
  transform(c, b = as.numeric(b)) %>%
  mutate(across(where(is.numeric), ~ round(., 4)))
c_both <- c[, 1:3]

T1=t.test(TEALL3[,12],TEALL3[,1],paired=T)
T2=t.test(TEALL3[,12],TEALL3[,2],paired=T)
T3=t.test(TEALL3[,12],TEALL3[,3],paired=T)
T4=t.test(TEALL3[,12],TEALL3[,4],paired=T)
T5=t.test(TEALL3[,12],TEALL3[,5],paired=T)
T6=t.test(TEALL3[,12],TEALL3[,6],paired=T)
T7=t.test(TEALL3[,12],TEALL3[,7],paired=T)
T8=t.test(TEALL3[,12],TEALL3[,8],paired=T)
T9=t.test(TEALL3[,12],TEALL3[,9],paired=T)
T10=t.test(TEALL3[,12],TEALL3[,10],paired=T)
T11=t.test(TEALL3[,12],TEALL3[,11],paired=T)

W1=wilcox.test(TEALL3[,12],TEALL3[,1],paired=T)
W2=wilcox.test(TEALL3[,12],TEALL3[,2],paired=T)
W3=wilcox.test(TEALL3[,12],TEALL3[,3],paired=T)
W4=wilcox.test(TEALL3[,12],TEALL3[,4],paired=T)
W5=wilcox.test(TEALL3[,12],TEALL3[,5],paired=T)
W6=wilcox.test(TEALL3[,12],TEALL3[,6],paired=T)
W7=wilcox.test(TEALL3[,12],TEALL3[,7],paired=T)
W8=wilcox.test(TEALL3[,12],TEALL3[,8],paired=T)
W9=wilcox.test(TEALL3[,12],TEALL3[,9],paired=T)
W10=wilcox.test(TEALL3[,12],TEALL3[,10],paired=T)
W11=wilcox.test(TEALL3[,12],TEALL3[,11],paired=T)

t <- c(T1$p.value,T2$p.value,T3$p.value,T4$p.value,T5$p.value,T6$p.value, T7$p.value, T8$p.value, T9$p.value, T10$p.value, T
11$p.value)
w <- c(W1$p.value,W2$p.value,W3$p.value,W4$p.value,W5$p.value,W6$p.value, W7$p.value, W8$p.value, W9$p.value, W10$p.value, W
11$p.value)
n <- c("1. LDA", "2. QDA","3. Bayes", "4. LogReg","5. KNN1", "6. KNN3", "7. KNN5", "8. KNN7", "9. KNN9", "10. KNN11", "11. K
NN13")

a <- data.frame(cbind(as.numeric(t),as.numeric(w)))

format(round(a, 4), nsmall=4)

b <- cbind(n, a)

### compare final selected models

T1=t.test(TEALL[,6],TEALL2[,5],paired=T)
T2=t.test(TEALL[,6],TEALL3[,12],paired=T)

t <- c(T1$p.value,T2$p.value)
w <- c(W1$p.value,W2$p.value)
n <- c("Pedagogical", "Combined")

a <- data.frame(cbind(as.numeric(t),as.numeric(w)))

format(round(a, 4), nsmall=4)

b <- cbind(n, a)
```