Using Machine Learning to Predict and Analyze Student Grades in an Introductory Programming Course

Anonymous
Anonymous Institution
anonymous@anonymous.edu

ABSTRACT

In this paper, we investigate if machine learning can be beneficial to teachers and students alike, and the ways in which it can be used. The data was collected from students taking an introductory programming course at a large public Hispanic Serving Institution. Regression is looked at first to create a model that can predict student final grades. Using only data up to and including the midterm, the linear regression, kernel ridge regression, and Gaussian process regression were able to predict student's final grade with an average error of 6 - 7 points off the actual grade (out of 100 points). By removing assignments to represent weeks 5 - 6 in the course, the models were still able to predict student's final grade with an average error of less than 10. Classification was examined next, using a neural network to predict the letter grade a student receives. Again using only data up to and including the midterm, the initial letter grade prediction had an accuracy of 70%, but was improved to 90% when the first and second most likely predictions were taken into account. Finally, we look at how using the coefficients from the linear regression can help an instructor identify the way in which different topics affect their class. We were able to find the topics that most negatively and positively affected students' grades in the introductory programming course.

Keywords

regression, classification, Computer Science education, student performance, topic analysis

1. INTRODUCTION

Combining knowledge from both the Education field and Computer Science, this paper focuses on how we can use data analysis techniques to inform and improve teaching. Many schools may use some type of data analysis, but in a very rudimentary and, for the most part, unhelpful way. For example, data analysis may consist of getting the results of student tests and seeing what percentage of students got each problem correct. Often these results would be discussed during a short meeting, but never really end up with clear plans for change. This type of data analysis was done well after the test, and therefore was of little use as the class is already moving on to the next section.

This is where the new emerging fields of Learning Analytics and Educational Data Mining step in [1][11]. Morten Søby gives a good initial description, explaining that Learning analytics is an educational application of big data, a statistical approach that was originally leveraged by businesses to analyse commercial activities, identify spending trends, and predict consumer behaviour [9]. There is a lot of unexplored space in using these big data ideas in education, especially when it comes to predictive models. This type of analysis could be infinitely more helpful, as we could potentially have ways to preemptively identify struggling students and identify patterns in the way the class is taught. By using student grade data from an introductory programming course at a large public Hispanic Serving Institution, the goal of this paper is to explore how we can use machine learning to better analyze educational data and improve education at various levels.

Many learning analytics and educational data mining research create models for predicting student performance. These research use a variety of data sources such as summative assessments, student engagement (e.g., attendance, interaction with an LMS, activity completion), and demographic data [7][3][5]. Formative assessment data such as quizzes, lab exercises, and projects are useful for developing appropriate feedback to help students learn [10]. This work joins other research that use formative assessment to predict student performance [6][8]. In addition, it provides a deeper analysis of prediction models to identify assessments (data features) that influence student performance. Such insights can inform educators about important topics they need to consider to improve their teaching.

This research does not necessarily create a program for any school or teacher to run, but rather it investigates whether machine learning works in this context, what type of models work best, and what else we can learn from this approach. With this in mind, however, there are several objectives and goals. The main objective is to identify if there are concrete benefits to using a predictive model to analyze student grades. To accomplish this, several machine learning models are created to try to accurately predict a student's final grade. Various regression models will be used, as well as

looking at it as a classification problem using a neural network. The goal is to create models that predict the student's final grade to within 10%. With the nature of educational data it would be impossible to get a completely accurate prediction (a student may feel sick one day and do worse than normal, they may have obstacles outside of school that affects them for a time, etc.), but 10% is generally one letter grade, so predictions of this accuracy can be considered a success. We can conduct experiments to find a balance between the least amount of data features required to still get a passable prediction.

In addition to this main objective, there is still more analysis that can be done to look at other aspects of the educational field. One interesting component of these models, especially with a linear regression, is that it gives coefficients that explain how the model weights each input. Since the inputs in this case are class assessments, we can get a better understanding on how these assignments are affecting the students' final grades. By then looking into the topic for each of these assignments, we can further analyze the course itself and potentially identify ways we can change how the course is taught. The goal is to identify which topics affect student grades positively and negatively, and see if there are any commons themes among all these topics.

2. DATA

Due to the importance of the data for this project, it is worthwhile to briefly discuss what data is being used and how it is handled. As mentioned, the student grade data is from an introductory programming course at a large public Hispanic Serving Institution. Included are the grades for all quizzes, labs, projects, and tests from the Fall 2018, Spring 2019, and Fall 2019 semesters. Altogether, this comes out to 214 students in total. Table 1 provides descriptive statistics on the data collected.

Table 1: Descriptive statistics on collected data

Data	n	M	SD
Fall 2018 Students	59		
Spring 2019 Students	60		
Fall 2019 Students	95		
Total Students	214		
Quiz 1 grade	214	84.94%	0.20
Quiz 2 grade	214	83.92%	0.21
Quiz 3 grade	214	60.88%	0.29
Quiz 4 grade	214	66.58%	0.26
Quiz 5 grade	214	68.71%	0.25
Quiz 6 grade	214	68.02%	0.29
Quiz 7 grade	214	71.16%	0.31
Quiz 8 grade	214	56.07%	0.33
Lab 1 grade	214	98.13%	0.14
Lab 2 grade	214	90.95%	0.24
Lab 3 grade	214	92.48%	0.21
Lab 4 grade	214	87.95%	0.27
Lab 5 grade	214	85.80%	0.30
Lab 6 grade	214	73.70%	0.38
Lab 7 grade	214	61.91%	0.42
Project 1 grade	214	84.87%	0.29
Project 2 grade	214	68.08%	0.39
Project 3 grade	214	66.95%	0.39
Midterm	214	68.66%	0.26
Final grade	213	71.79%	0.26

Since the goal is to predict the final grade, it would not be very helpful to make a prediction when the student no longer has any time to improve their grade. Therefore, only the grades up to and including the midterm will be used. All the quizzes and labs will be used for some additional topic analysis about the class, however. Two important notes here are firstly that the percentage score for each assignment will be used. This is to normalize the data across various assignments that may have different point totals. For example, the first quiz may only be worth six points, while the fifth quiz is worth 14 points. This could incorrectly weight the fifth quiz much higher than it should be, even though the quizzes contribute equally to the students' final grade. Secondly was to define all missing assignments as a score of zero. This is reflective of how the grading works in the class, so all the missing values in the data are changed to zeros.

When using machine learning to make predictions, it is important to divide the data up into multiple train and test sets so that the model does not just learn the initial data set without being able to generalize it to new data. Especially in this case since the goal is to predict student grades for an entirely new semester. Therefore, the 214 students are divided up into a cross-validation set, development set, and test set. The test set contains 10% (21 students) of the data, and is set aside for a final test of the model. Of the data that is left, the cross-validation set contains 70% (135 students) and the development set contains 30% (58 students). The idea for these two sets is that the cross-validation contains the bulk of the data and is what the model is trained on, using the cross validation scoring method. The development set is then used to initially test the model. We can see what the predictions look like, and actually look into the development set for further analysis. We can apply feature engineering on this set to modify the features used by the models without introducing biases as the development set is not the cross-validation set used for evaluating performance. With all the data now set up and ready, we can create various models to make predictions and analyze the data.

3. REGRESSION

The first analysis to explore is regression. The nature of the data is that there are various inputs, and we are trying to predict a number for the output. A regression is a good fit for this type of data, especially because a linear regression, which is one of the simplest regressions, seems like it will be very effective. To not limit it to just one regression, six different types of regression provided by the Python-based scikit-learn library were used to determine which ones were the best fit for the data. The machine learning algorithms that were tested include kernel ridge regression ($r^2 = 0.76$), linear regression ($r^2 = 0.76$), gaussian process regression ($r^2 = 0.76$), k-nearest neighbor regression ($r^2 = 0.72$), multi-layer perceptron regression ($r^2 = 0.57$), and support vector regression ($r^2 = 0.37$). By looking at the r^2 score, which gives a measure of how well the model predicts the data, we can compare the models to see which one works best. This paper specifically will focus on the kernel ridge regression, because it performed the best and

Table 2: Five most accurate predictions using data until the midterms with Kernel Ridge Regression

Actual	Predicted	Difference
87.08	87.27	0.19
79.56	79.29	0.27
48.06	48.41	0.35
99.42	98.94	0.48
88.48	89.05	0.57

Table 3: Five least accurate predictions using data until the midterms with Kernel Ridge Regression

Actual	Predicted	Difference
58.13	75.76	17.63
32.81	50.63	17.82
34.52	52.50	17.98
65.60	47.13	18.47
30.49	11.98	18.51

the linear regression, which will be used later in the topic analysis.

3.1 Kernel Ridge Regression

The kernel ridge regression is slightly more complex than a linear regression and looks to address some of the problems that arise from ordinary least squares linear regression. Kernel ridge regression helps deal with multicollinearity which could improve the model [12]. The cross-validation set was used to train the model. The model's r^2 score using the cross-validation set was 0.76. The model was then tested using the development set to investigate the errors between the actual final grade and the model's prediction. The first experiment used the first eight quizzes, the first seven labs, the first three projects, and the midterm exam. Tables 2 and 3 show the five lowest and highest errors (out of 58 data points).

We get an average error of 6.32%, which is much less than the initial goal of 10%. Average error only tells so much, however, because if it is extremely accurate for some scores, but wildly inaccurate for others, that would not be much help. By looking into the errors from the development set, we see that it only has 14 predictions out of 58 that have a difference greater than 10, and none are over 20. More than half of the predictions have a difference that is less than 5. So far this is a huge success, as we seem to be accurately predicting student grades to within a letter grade, and many of the best predictions have errors that are less than one. The kernel ridge regression especially shines compared to the other algorithms due to a standard deviation of 5.52, which is lower than any other regression that was tried. This gives a sense that the predictions are more tightly grouped, and so we can have more trust in the predictions.

The next step is to take away assignments to see how early in the semester we can still get accurate predictions. For kernel ridge regression we can go all the way back to the sixth week of class, which includes the first six quizzes, the first five labs, and the first two projects. Take note that the midterm exam is scheduled in the ninth week of the semester. The average error is 8.06%, and the standard deviation is 6.40. Again,

Table 4: Five most accurate predictions using first six weeks of data with Kernel Ridge Regression

Actual	Predicted	Difference
8.41	8.85	0.44
93.82	93.34	0.48
79.56	80.51	0.95
88.17	87.20	0.97
41.97	40.90	1.07

Table 5: Five least accurate predictions using first six weeks of data with Kernel Ridge Regression

Actual	Predicted	Difference
34.52	52.77	18.25
12.92	-5.82	18.74
68.28	89.97	21.69
80.15	54.90	25.26
94.11	67.45	26.66

we have an average error still under 10, and the standard deviation rises slightly but stays pretty consistent. Tables 4 and 5 show the errors using the development set.

Even after going back three weeks into the semester, we only have three predictions with an error greater than 20%. 20 out of 58 students predictions had an error greater than 10%, which is only six additional students compared to when we ran the model with everything included. The average error was up by almost 2%, so the predictions are obviously less accurate, but keeping in mind the context of these predictions the difference between 4 and 6% from the final grade is irrelevant. The kernel ridge model seems to be very powerful at lowering the variance, which is an extremely important aspect of the model when coupled with keeping the error low as well.

As we use less and less data, it seems that most of the accurate predictions stay about the same, but the predictions that started off somewhat inaccurate get much worse. Most likely this is due to students who had a drastic change in approach and effort. At this point we are very early into the semester, only at week six. There are students who could score poorly on the assignments up to this point, realize they need to change, and then still end up with a good grade because there are so many assignments left. This also applies in the reverse, where a student starts extremely strong but gets complacent, and ends up with a final grade that is much lower. By looking at individual students, we can see that this is often the case. Student 178, one of the worst predictions by the model, got a 0% on the first three quizzes but then began scoring very well, and even got around a 90% on the midterm. On the other hand, Student 90 started by getting 100% on almost every assignment, but then missed a few assignments, got a 25% on the eighth quiz, and around a 70% on the midterm. By including everything up to the midterm, this effect is much less pronounced.

Due to this analysis, these results seem to still be extremely beneficial. We can see that for the average student, the prediction is accurate to within 5% of their actual grade, and it only starts deviating greatly when students have unique

Table 6: Five most accurate predictions using data until the midterms with Linear Regression

Actual	Predicted	Difference
99.42	99.14	0.28
48.06	47.76	0.30
93.82	93.50	0.32
87.08	86.72	0.36
79.56	79.10	0.46

Table 7: Five least accurate predictions using data until the midterms with Linear Regression

Actual	Predicted	Difference
58.13	75.44	17.31
32.81	50.55	17.74
34.52	53.50	18.98
65.60	46.17	19.43
30.49	10.54	19.95

circumstances, like students 178 and 90 described above. However, a case can be made that these unique circumstances can be largely disregarded when doing these predictions, since these are the cases where the student actively changed their approach and effort. This prediction in week six will still accurately predict a student's final grade if they continue on the same path. Through this analysis, we have found that by using these models we can accurately predict student grades (with an average error of under 10%), as early as the sixth week, which is only a third of the way through the semester.

3.2 Linear Regression

We also want to further explore the linear regression. Even though the kernel ridge slightly outperformed it, the linear regression is much simpler to use and understand. If its predictions have similar errors to the kernel ridge, then it can also be a powerful tool (which we will see later in the topic analysis). For the first model, the same data was used as for the kernel ridge: the first eight quizzes, the first seven labs, the first three projects, and the midterm. The linear regression also gets an initial r^2 score of 0.76 using the cross-validation set. Tables 6 and 7 show the errors on the development set.

The average error is 6.52%, and the standard deviation is 5.64. The linear regression is slightly worse than kernel ridge, but the numbers are very comparable and still give successful predictions. We still only have 14 out of 58 predictions that had an error over 10%, and none that were greater than 20%.

The linear regression was taken back to week 6, which similarly contains the first six quizzes, the first five labs, and the first two projects. The average error is 8.24% (compared to 8.06% for kernel ridge regression), and the standard deviation is 7.25 (compared to 6.40 for kernel ridge regression). Table 8 and 9 show the errors on the development set.

This is where the benefits of the kernel ridge regression become more apparent. While the linear regression has a comparable average error, the standard deviation starts to become noticeably worse. This can further be seen concretely by comparing Table 4 and 5 to Table 8 and 9. The top 5 scores have similar differences, but the bottom 5 scores show the linear regression has more variation. The worst prediction has an error of almost 6% higher than the kernel ridge regression. The increased deviation from the average error caused one of the predictions to have an error that is over 30%. These of course represent the worst predictions, and are usually not the case, but demonstrate how important a low standard deviation is for being able to trust the predictions.

Table 8: Five most accurate predictions using first six weeks of data with Linear Regression

Actual	Predicted	Difference
93.82	94.25	0.43
79.56	80.00	0.45
14.60	15.41	0.81
88.17	87.33	0.84
89.09	90.18	1.09

Table 9: Five least accurate predictions using first six weeks of data with Linear Regression

Actual	Predicted	Difference
30.49	10.76	19.73
34.52	55.25	20.73
68.28	90.81	22.53
94.11	68.47	25.64
80.15	47.86	32.29

Overall, the linear regression performed almost as well as the kernel ridge regression and made successful predictions with an average error well under 10%. This analysis gives the necessary confidence that features of the linear regression model can be used, which will be expanded upon in the topic analysis.

4. CLASSIFICATION

The first idea when trying to predict student grades was to use regression because we can use the raw scores on assignments as features, and therefore predict a numerical score for their final grade. However, the educational data actually lends itself nicely to being transformed into a classification problem. A student's final grade is naturally sorted into the five categories of A, B, C, D, and F. For the purposes of this classification, a score of 100-90 is an A, 89-80 a B, 79-70 a C, 69-60 a D, and 59-0 an F. These scales may change for different classes, but in general this will be accurate. By transforming it to a classification problem, we can reduce the complexity of the learning task and possibly generate more accurate predictions. Regression and classification algorithms can be largely different, so for the most part the classification is going to be treated as its own problem.

Using the Python TensorFlow package and after much trial and error, the best results were achieved from using a neural network prediction model with the following general features. The input layer has 128 neurons and takes in an input of 19 student grades before the midterm. Then there are a couple hidden layers with dropout layers in between, and

Table 10: Grade predictions using softmax

First Choice	Second Choice	Actual
F	D	F
A	B	A
C	B	B
A	B	A
B	A	B

finally the output layer with five neurons, one for each letter grade. The activation used for the layers is relu, while the output layer is softmax. The adam optimizer was used, Categorical Crossentropy was used for the loss (since there are more than 2 categories), and it fits the model over 30 epochs. More epochs were initially used, but this caused the model to overfit.

With the data up to and including the midterm (first eight quuizzes, first seven labs, first three projects, and the midterm), the model was trained using the cross-validation set and tested on the development set. From this we got an initial accuracy of 70.7%. This is a pretty good accuracy, but would need it to be higher to consider it beneficial to teachers and students alike.

One of the main reasons the accuracy may be lower for the classification is that there are a lot of edge cases with grades that the regressions get right but the neural network classification gets wrong. For example, if a student has a grade of 91% and the regression predicts it at an 89%, this is an extremely good prediction because it is only 2% off! However, in the neural network's case, it would predict a B when the student got an A, which is wrong. The difference between the two letter grades can be anywhere between a 1 to 10% performance difference. So, we need to look a little deeper into the predictions from the neural network.

One important feature of using the softmax activation is that it basically turns the weights of the output layer into probabilities. Using this idea, we can take the predictions and find not only what the model predicts, but also what the second highest weighted prediction was. This can hopefully get rid of a lot of the edge cases described above, because if the first prediction is wrong there is a good chance the second-best prediction can cover it. This is an acceptable line of thought, because even if the model predicts a student to have an A or B instead of just an A, that still gives extremely valuable information on how the student is on track to perform in the course. Table 10 shows a small sample of the predictions using softmax compared to the actual student grade.

As can be seen, almost all of the second-choice grades border the first-choice grades (A borders B, B borders A and C, etc.). The accuracy was then computed by checking if either the first or second most likely grade matches the actual grade, which resulted in an accuracy of 89.7%. This is extremely close to 90%, and could be considered highly acceptable. Using this classification method with a neural network, we can predict a student's grade with 70% accuracy, but predict a one-letter-grade-range of their actual grade with 90% accuracy. This small range of A to B or D to C is still very beneficial for teachers and students to know.

Now that we know classification using a neural network can make acceptable predictions, we did a similar process to the regressions by stepping back week by week to see if it was still accurate enough. When neural network was tried with the fifth week data (first five quizzes, first four labs, and first two projects) the initial accuracy based off the best prediction was down to 56.9%, and the accuracy using the second-best prediction was at 82.8%. As less data was used, the main prediction significantly decreased. It would not be considered very useful at around a 50% accuracy. However, including the second-best prediction actually kept the accuracy high, staying above 80% even as early as the fifth week.

5. TOPIC ANALYSIS

So far we have mainly looked into predicting the final grades of students, and seeing which models worked and were the best. However, it is a valid question to ask if it is really worth it over just looking at a student's current grade and using that as a predictor. One big reason to use machine learning in this aspect is because it is able to find patterns in the data and use these patterns to take a predictive approach. One positive side effect of this approach is that we can see and analyze the weights and coefficients these models give us. Often these coefficients are very complex, but in the case of the linear regression they are very straight forward. Since the linear regression was performing about as well as the others and was giving accurate predictions, we can look into this topic even more.

The idea here is that the linear regression gives a weight for each assignment. The higher the weight, the more influence it has on the final prediction. Therefore, if a particular quiz has a higher weight than other quizzes, we can infer that a student's score on this quiz has an increased effect on their final grade, even though all the quizzes are worth the same points. This is something that would be very hard to determine by just looking at a student's current grade rather than this predictive machine learning model. We can take it even one step further because we know all the topics for the assignments in the class. So, we can relate a coefficient from the linear regression to an actual topic taught in the class. By analyzing in this way, we can get a better understanding of the course itself.

First, we need to go back to the linear regression and run it again, but with a change in the data we are using. Since this analysis is not about predicting a student's grade early in the semester, the data is not limited to before the midterm. However, since the midterm, final, and projects encompass a variety of topics, these would not be helpful. Therefore, the data will be using all the quizzes (1-11) and all the labs (1-13). Running the linear regression to predict the final grade with these produces the coefficients presented in Figure 1.

The coefficients go in order of quizzes 1-11 and then labs 1-13. So for example, quiz 3 has a weight of 0.034, while quiz 5 has a weight of 0.053. The model is saying that changes in quiz 5 will affect the final prediction more than changes in quiz 3, or that it has a higher weight. So now we can look in at the actual topics for these assignments

```
Coefficients:
 [-0.00357618  0.05052817  0.03373449  0.01673683  0.05326578  0.08071988
  0.04875324  0.02614526  0.0722052   0.05541508  0.06358611 -0.02392825
  0.12198743 -0.02993337  0.07089802  0.01298907  0.02446626  0.01890762
  0.06652099  0.02301382  0.075421    0.07096309  0.0341749   0.06546691]
```

Figure 1: Coefficients for linear regression.

to get a better understanding of the topics in the course. Another piece of information that we can add on for this analysis is the average score for each assignment. This can be even more telling, because now we can find topics that have a high coefficient, which influence the final grade more so than others, but also see if the scores on those important topic's assignments are good or bad. To do this two tables were created, one sorted by average score, and one sorted by coefficient.

To better visualize the results, each assignment was given a number from 1-24 based on their position in the list. The first numbering was based on the most impactful negative assignments, or basically those assignments that have a high coefficient, but a low average score. The topic with the lowest coefficient got a 1 and the highest a 24. The topic with the highest average score got a 1 and the lowest a 24. These will be the assignments that affect a student's final grade the most, but that students also struggle with. By summing these two numbers together, we can get a list of the most impactful negative assignments. Table 11 shows the assignment, its rank in terms of average scores, its rank in terms of the linear regression coefficient, and the course topic it covers. The table is sorted by the sum of both ranks.

Using this table we can begin to make some initial observations. The first is to just look at which topics are highest on the list which are Classes, Polymorphism, and Recursion. These would be the topics that are the most likely to cause students to do poorly in the class. This can give a general overview for a teacher as to what their students are most likely to struggle on but are also extremely important to pass the course.

We can further investigate the highly ranked topics. Classes are highly ranked on the list which makes sense in this case, because most of the topics that are learned after it, such as Advanced Classes, Advanced Objects, Polymorphism, Inheritance, and Composition all rely heavily on the initial understanding of Classes. But Classes is also one of the first big new topics in the course, and so while many students already know the basics of Loops and Functions, Classes are something entirely new. This could be a reason as to why the scores are low. On the other hand, a lot of the initial topics that are taught early in the semester, such as Conditional Statements, Loops (which does not have a great average score but a very low coefficient), Variables, Expressions/Operations, seem to have a much smaller negative effect on student grades. These topics were initially introduced in a prerequisite course and reviewed in the course we are investigating. With this in mind, it might be beneficial to reduce discussions on those early topics and spend more time on Classes to ensure students have a really good

Table 11: Assessments with their average score and coefficient ranking.

Asses.	Avg.	Coeff.	Topic
Quiz 9	23	21	Classes
Lab 13	24	17	Polymorphism
Lab 8	22	18	Recursion
Quiz 6	12	23	Arrays
Quiz 11	19	16	Inheritance
Lab 11	14	20	Composition
Quiz 10	17	15	Advanced Objects
Lab 10	9	22	Advanced Classes
Lab 12	20	11	Pointers & Objects
Quiz 8	21	9	Pointers
Lab 6	20	8	Functions
Quiz 5	13	14	Functions
Lab 2	3	24	Standard Input & Expressions
Quiz 3	16	10	Conditional Statements
Lab 7	18	6	Arrays
Quiz 7	11	12	Recursion
Lab 4	4	19	Switch Statements
Quiz 2	7	13	Expressions/Operations
Quiz 4	15	5	Loops
Lab 9	8	7	Pointers
Quiz 1	6	3	Variables, Std. I/O
Lab 5	5	4	Loops
Lab 1	1	2	Tuffix Setup
Lab 3	2	1	Conditional Statements

understanding before moving into the later topics.

Another interesting feature for this particular data set is that each topic often has a quiz and lab associated with it. In some cases, like with Functions, we can see that the quiz and lab are right next to each other in the middle of the list. However, quiz 9 on Classes and lab 10 on Advanced Classes are very similar topics and have almost the same coefficient, but Classes is much further up on the list. This is due to the fact that quiz 9 averages the second worst scores, while lab 10 has the ninth best. Obviously, students seem to be doing much better on the lab portion of the topic than the quiz, and so it can be worthwhile looking into what is different about the quiz and lab, and what the lab does that seems to work better. Quizzes and labs are two very different assignments, and so they may not have a direct connection, but it would still be beneficial to look further into these ideas.

Overall, this type of analysis requires much more time, in depth thinking, and reflection than just looking at the performance of prediction models. However, they can be used to much greater effect to inform a teacher about how they teach their course. It can give a clearer idea about which topics are really affecting the course the most. If one topic is drastically impacting the course negatively, then it becomes clear that something needs to change. Whether that is how the material is actually taught, when it is taught, how long is spent on it, etc., is up for the teacher to use their expertise to decide. By potentially comparing this data between teachers, we can see if one teacher seems to be better at teaching a certain topic and try to understand why that is. There are many possibilities using this information.

6. IMPROVEMENTS AND EXTENSIONS

There can still be many improvements and extensions to what this paper has done. The data was taken from a specific class, so the concrete numbers from the analysis really only apply to that class. While it has been shown that using machine learning has real benefits for education, this research can be improved by making it more general. What this paper details only works for one specific class at a time, and that is assuming the class is taught with a similar structure and has the same assignments each time it is taught. To further expand upon this, the program built in this research cannot, for example, be reused by a high school teacher in their class. One extension to this research could be to now develop software that allows a teacher to input data and find the best models for their course.

This paper just used the numerical grade data from the class. Another huge source of data that can be expanded upon is adding in demographic data to see if that improves the predictions. A final improvement that ties into this is the amount of data. Neural networks are great at analyzing huge amounts of data, but in this case there were 213 total students that were even further divided up into smaller sets for testing. Good data can be hard to get in the educational setting, however, because as soon as assignments and structure start changing, the old data may not be useful anymore. One idea to help with this issue is to synthetically create data using approaches such as SMOTE [4] and Generative Adversarial Networks (GAN) [2]. This can create new synthetic data that is just as good as "real" data, which would greatly help with this problem.

7. SUMMARY AND CONCLUSION

This paper explored and showed the benefits of using machine learning with educational data. This was accomplished by creating prediction models, using both regression and classification, as well as looking into the individual coefficients of the model to better understand which topics positively and negatively affect that prediction of a student's final grade.

Regression models were created to predict the final grade. The three highest performing models were the linear regression, kernel ridge regression, and gaussian process regression. Using these models, we were able to get predictions with an average error of around 6 points off the actual final grade. We also found that by removing assignments to represent a time even earlier in the semester, the models still gave useful predictions as early as five weeks into the semester. Overall, the kernel ridge regression was found to be the best performing model, but linear regression was useful for later analysis.

Another way to look at the data was as a classification problem using neural networks, with letter grades instead of a flat number. The initial accuracy of the model was not the best, around 70%, but by looking at both the best and second-best predictions from the model we were able to raise it to 90%. This is an acceptable analysis, because as a teacher knowing that a student is predicted to get a B but maybe a C can be just as valuable as predicting them to get a B, but adds a lot more confidence into the prediction.

Finally, we looked further into the linear regression. This regression got very similar scores to the kernel ridge regression, but is much simpler to understand. This simplicity allowed an analysis of the coefficients of the regression to determine which assignments, and therefore topics, seemed to affect the course most negatively and positively. From a basic understanding of the course we were able to analyze several of the topics, and even suggest potential changes. This information could be further used by the teacher of the course itself.

Overall, we found that using machine learning in education can have many benefits. The models made acceptable predictions that can be used by teachers to identify how students are projected to do in the course and identify students in danger of failing. This automates a lot of this process that teachers may not have time to do, and can in fact recognize underlying patterns that give better predictions than just looking at a student's current grade. We can even use these models to look further into the structure and topics of a course, allowing us to make informed decisions on how to improve the education we are giving.

8. REFERENCES

[1] R. S. Baker and P. S. Inventado. Educational data mining and learning analytics. In *Learning analytics*, pages 61–75. Springer, 2014.

[2] K. T. Chui, R. W. Liu, M. Zhao, and P. O. De Pablos. Predicting students' performance with school and

family tutoring using generative adversarial network-based deep support vector machine. *IEEE Access*, 8:86745–86752, 2020.

[3] P. Cortez and A. M. G. Silva. Using data mining to predict secondary school student performance. 2008.

[4] H. Guo, J. Zhou, and C.-A. Wu. Imbalanced learning based on data-partition and smote. *Information*, 9(9):238, 2018.

[5] J. L. Harvey and S. A. Kumar. A practical model for educators to predict student performance in k-12 education using machine learning. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 3004–3011. IEEE, 2019.

[6] J. Nouri, M. Saqr, and U. Fors. Predicting performance of students in a flipped classroom using machine learning: towards automated data-driven formative feedback. In *10th International Conference on Education, Training and Informatics (ICETI 2019)*, volume 17, pages 17–21, 2019.

[7] C. Romero and S. Ventura. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(1):12–27, 2013.

[8] P. Shayan and M. Zaanen. Predicting student performance from their behavior in learning management systems. *International Journal of Information and Education Technology*, 9(5):337–341, 2019.

[9] M. Søby. Learning analytics. *Nordic Journal of Digital Literacy*, 9(02):89–91, 2014.

[10] D. Tempelaar, B. Rienties, J. Mittelmeier, and Q. Nguyen. Student profiling in a dispositional learning analytics application using formative assessment. *Computers in Human Behavior*, 78:408–420, 2018.

[11] O. Viberg, M. Hatakka, O. Bälter, and A. Mavroudi. The current landscape of learning analytics in higher education. *Computers in Human Behavior*, 89:98–110, 2018.

[12] V. Vovk. Kernel ridge regression. In *Empirical inference*, pages 105–116. Springer, 2013.