Investigating the Variables that Influence Online Learning

Peng Hui How, Pak Tao Lee

Stanford University
phow1@stanford.edu ptlee@stanford.edu

Abstract

Our project aims to explore a dataset that logs students' behaviors in solving mathematics exercises that are hosted on the Khan Academy [1] platform. We aim to pinpoint the significant factors (out of a pool of given factors) that suggestively predict students' engagement with solving problems and their learning gains, and to investigate the accuracy between several classical machine learning techniques in predicting students' performance.

1. Introduction

Within the last 18 months, online educational technology startups organizations such as Khan Academy and Coursera have been offering learning experiences like massive open online courses (MOOCs) to millions of users. This is just an iceberg of the huge impact of online education on the humanity. In general, online education promotes educational equality - people all around the world are now granted convenient access to high quality education. An effective model that governs the success of online learning, as measured by the learning progress of the students, is therefore in dire need. Our ultimate goal is to discover the underlying factors that affect students' absorption of knowledge, which has practical implication for provider of online education and the broader teaching community.

2. Dataset

2.1. Overview

Our current dataset consists of 5568838 tuples, each of which represents a single question attempted by a student with a unique User ID, taken as a subset of our preliminary data, which was collected from an experiment that ran on Khan Academy's platform for a month (with 200,000 students, 12 topics, 6 types of message statements). The overall dataset is organized in a hierarchical structure, in which each problem belongs to 1 of the 12 possible topics (e.g. algebra) (this is not explicitly stated in the dataset - instead, the dataset indicates the specific subtopic within the main topic, hence we need to insert an additional column for this to facilitate future analysis); and is being categorized under one of the various subtopics under a single topic; for identification, problem number, the sequence in which the problem appears (in a single topic), is also available.

[1] www.khanacademy.org

2.2. Variables

There are 13 variables in the dataset, those that match our pertinent interest to measure factors that help in online learning include the following:

1. Break variable:
   - `identity`: UserID of the student, may appear multiple times in the dataset
2. Mediation variable:
   - `alternative`: Enumerated from 0 to 5, the message type displayed on the problem attempted
   - `exercise_type`: Enumerated from 0 to 2, pre-, during, and post- close intervention
   - `exercise`: A node on the Khan Academy's knowledge map, indicating an exercise
3. Possible predictors (indepedent variables):
   - `num_coaches`: Number of instructors, constant as per User ID
   - `hints`: Number of hints employed by the student for a question, range differs for each question
   - `topic_mode`: Binary variable $\{0, 1\}$, whether or not the problem was attempted under topic-specified condition
   - `review_mode`: Binary variable $\{0, 1\}$, whether or not the problem was attempted under review condition
   - `proficiency`: Binary variable $\{0, 1\}$, depending on whether or not this problem was awarded a proficiency, indicating the problem's difficulty level
   - `time_taken`: Time taken for the students to finish a particular question
   - `problem_number`: The problem number assigned in the particular exercise
   - `problem_order`: Generated variable, the order of the problem attempted for the particular student, distinct within the same UserID

4. Performance metric (dependent variable):
   - `correct`: 0 or 1, whether or not this question was correctly answered

2.3. Analysis

Given that the 6 classes of specially designed statements displayed aside of the questions were factored into the dataset, we roughly analyzed the significance of these statements, tabulating count, mean and standard deviation gives the following:

Statement	Mean	S.D.	Frequency	L/H[2]
Control Statement	0.77	0.423	1074421	L
Science Statement	0.74	0.437	588019	L
Positive Statement	0.79	0.405	486360	H
No Header	0.79	0.404	2440579	H
Growth Mindset	0.79	0.407	491045	H
Growth Mindset + Link	0.79	0.404	488414	H
Total	0.78	0.412	5568838	

With the exception that science statement apparently gives rise to a performance slightly poorer than average, the other 4 statements inserted performed almost equally well. As the variation of mean and standard deviation across all conditions is insignificant, we have decided not to factor it into our further analysis.

3. Methods

Before plunging into the analysis, we preprocessed the dataset by doing the following:

3.1. Grouping by ID and Variable Transformation

The dataset given is grouped by problem instead of student ID, thus there might be multiple problems done by a single student across the dataset, assigning equal weight to each tuple in the dataset might result in the apparent pattern generated to be biased towards that of the students who do more problems, thus skewing the analysis result. To tackle this, we grouped the dataset by student ID, followed by taking means for all the corrseponding variables. We have also replaced `problem_order` with the new variable `num_problems`, taking the largest value of `num_problems` within each ID, indicating the number of problems done by the student.

3.2. Anomalies Detection and Deletion

At times, small amount of outliers are able to distort the entire distribution of the dataset, resulting in erroneous conclusion. We identify them by plotting Bell Curve and reading the cumulative frequency table, (for the variables that have no theoretical upper bound) delete the 1 or 2 % at the extreme right of the Bell Curve, procedurally, on each of the dependent variables. Our resulting dataset has 229865 out of the original 247486 tuples, keeping the vast majority (92.9%) of the dataset while greatly reduced the variation, as detailed below:

	Mean		Std. Dev.	
Variables	**Before**	**After**	**Before**	**After**
hints	0.4855	0.2594	1.826	0.8779
time_taken	51.69	26.41	461.2	27.45
num_coaches	0.68	0.66	0.852	0.748

As an example, the graphical illustration of the distribution of the variable `time_taken` before and after condensation is as followed:

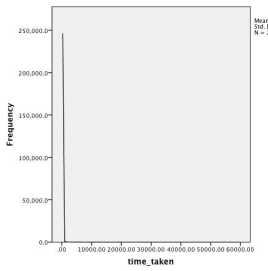

Figure 1: *Time_taken count, before condensation*

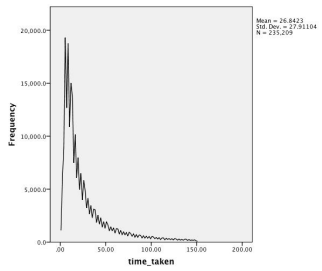

Figure 2: *Time_taken count, after condensation*

The grouping process that transforms the dataset from problem-based to student ID-based has turned some discrete variables into continuous ones, a very huge potential problem causer in future analysis. Therefore for each of the variables, we come up with a discretized copy of itself.

3.3. Variable Discretization

We discretized the continuous variables into several (3 - 5) discrete portions, to facilitate the calculation of Mutual Information (M.I.) Values and Naive Bayes.

3.4. Feature Selection

3.4.1 Mutual Information

To identify the variables that play the most significant role in predicting the students' performance measure, we calculate their mutual information (M.I.) values, where:

$$M(x_i, y) = \sum_{x_i} \sum_{y_i} p(x_i, y) \log \frac{p(x_i, y)}{p(x)i)p(y)}$$

, on the discretized variables, as detailed in the previous subsection. The result we obtained is as follows:

Variables	M.I.
review_mode	-0.0959
proficiency	-0.243
hints	-0.275
topic_mode	-0.377
num_problem	-0.607
time_taken	-0.845
num_coaches	-1.02
time_done	-1.56

In this stage, we eliminate `num_coaches` and `time_done`, two variables having the lowest M.I. values. This makes sense, especially for the latter, matching our intuition that the date when the students do the problems has no significant influence towards the performance.

3.4.2 Correlation Measure

	hints	**time_taken**	**topic_mode**	**review_mode**	**proficiency**	**num_problem**
hints	1	0.18	0.059	0.002	-0.055	-0.006
time_taken	0.18	1	-0.002	0.009	-0.062	0.016
topic_mode	0.059	-0.002	1	-0.037	**0.199**	0.056
review_mode	0.002	0.009	-0.037	1	-0.028	0.053
proficiency	-0.055	-0.062	**0.199**	-0.028	1	-0.061
num_problem	-0.006	0.016	0.056	0.053	-0.061	1

In this stage, we eliminate the variable `proficiency`.

We now proceed to our data analysis with the 5 selected features, i.e. `review_mode, hints, topic_mode, num_problem, time_taken, num_coaches`, and `time_done`, using logistic regression and Naive Bayes, these classifications method are chosen as our dependent variable, i.e. `correct`, is originally a binary variable (valued at $\{0,1\}$).

As the dataset is being grouped by student ID, the variable `correct` at each row is now the average score of the user across the dataset. In order to concretely visualize the trend across this new dataset, and to discretize this new `correct` value to its original property, i.e. a binary variable, it is necessary for us to assign a threshold between 0 to 1, and set those above it to be 1 and the rest to be 0. To do this, we create 9 new variables `correct_0.x`, $x \in [1,9]$, and perform both logistic regression (we change the regression threshold to match the corresponding threshold) and Naive Bayes on this newly structured dataset.

Remark. For the rest of the report, T.N., F.N., F.P., and T.P. represents true negative, false negative, false positive, and true positive respectively.

3.5. Logistic Regression

Since `correct` is a binary variable, it makes sense to consider logistic regression to predict our results. The model is as follows:

$$y = h_\theta(x) = \frac{1}{1 + e^{\theta^T x}}$$

Running logistic regression with different cutoff points ranging from 0.1 to 0.9 gives the following results[3]:

Threshold	T.N.	F.N.	F.P.	T.P.	Precision
0.1	2045	160	27343	200317	0.880351511
0.2	4299	513	25192	199861	0.888173493
0.3	6037	1027	23864	198937	0.891714702
0.4	8183	1887	23957	195838	0.88756879
0.5	15738	3015	30607	180505	0.853731538
0.6	18276	5955	29548	176086	0.845548474
0.7	25578	11378	29553	163356	0.821934614
0.8	38590	30474	23320	137481	0.765975681
0.9	65165	164582	1	117	0.284001479

Table 1: *Results of Logistic Regression*

3.6. Naive Bayes

Another method that can be applied on a classification problem is the NaÃŕve Bayes[4] method. We will predict correct using these three variables, with a model similar to the email spam classifier problem we saw in lecture, where for $j \in \{0,1\}$, $p(y = j \mid x)$ is:

$$\frac{(\prod_{i=1}^{n} p(x_i \mid y = j))p(y = 1)}{(\prod_{i=1}^{n} p(x_i \mid y = j))p(y = 1) + (\prod_{i=1}^{n} p(x_i \mid y = j))p(y = j)}$$

Running Naive Bayes with different cutoff points ranging from 0.1 to 0.9 gives the following results:

[3] Precision = (T.P.+F.N.)/(The size of the dataset)

[4] In our case, $n = 5$. We are also assuming that the features we used are conditionally independent given y.

Threshold	T.N.	F.N.	F.P.	T.P.	Precision
0.1	1379	130	47698	180658	0.791930046
0.2	2142	260	48600	178863	0.787440454
0.3	3001	555	50572	175737	0.777578144
0.4	4391	957	66202	158315	0.707832858
0.5	7765	1438	68003	152659	0.697905292
0.6	8137	3452	76201	142075	0.653479216
0.7	13683	6326	79745	130111	0.62555848
0.8	19423	17000	64708	128734	0.644539186
0.9	37196	190763	30	1876	0.169978031

Table 2: *Results of Naive Bayes*

4. Results

4.1. ROC Curves - Threshold Selection

In order to maxize the true positive rate (TPR) and minimize the false positive rate (FPR), we plot the receiver operating characteristic (ROC) curve, a plot of TPR against FPR, at different cutpoints of our binary classification tests.

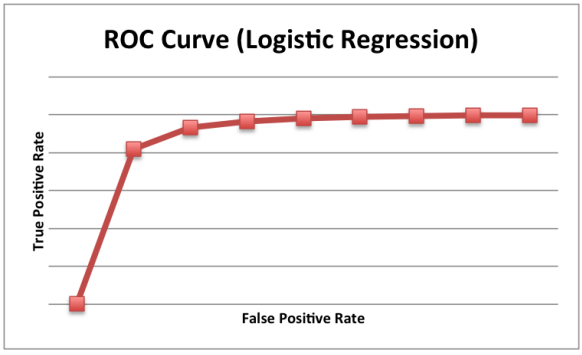

Figure 3: *ROC of the class of logistic regressions.*

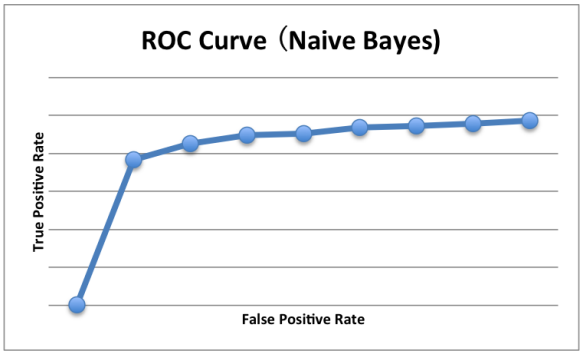

Figure 4: *ROC of the class of naive bayes.*

Interpolating from the ROC curves, we observe that for both analysis, threshold = 0.8 gives rise to the largest gradient on the ROC curves.

Using threshold = 0.8 as suggested, referring to Table 1 and Table 2, the precision of logistic regression and naive bayes are 76.6% and 64.5% respectively. Thus logistic regression performs notably better than naive bayes.

4.2. What Logistic Regression Suggests

	Beta[5]	S.E.[6]
hints	-4.233	0.048
time_taken	-0.017	0
num_problems	0	0
topic_mode	0.019	0.02
review_mode	0.017	0.049
constant	1.950	0.008

From the table, we interpret that logistic regression with 0.8 threshold suggests that
$\log \frac{p}{1-p} = 1.950 - 4.233 * hints - 0.017 * time\_taken$
$+0 * num\_problems + 0.019 * topic\_mode + 0.017 * review\_mode$
, where $p = p(correct = 1)$.

4.3. Interpretation of the results

In addition of these coefficients, we can also interpret in a very rough sense that the students tend to perform better when:

1. The topic is specified in the question (variable: `topic_mode`), or
2. When they are doing review (variable: `review_mode`);

while performing poorly when:

1. Many hints were requested (variable: `hints`), or
2. Longer time is taken (variable: `time_taken`).

Moreover, the total number of problems (variable: `num_problems`) has very insignifant influence on the student's results.

5. Evaluation

5.1. Cross Evaluation on Logistic Regression

Performing 2-fold cross evaluation on logistic regression with 0.8 threshold, we obtain accuracy of 76.6% and 75.4% respectively on the training set and the test set respectively.

5.2. Alternatives to Logistic Regression

As shown in the previous subsection, despite the discovered fact that logistic regression performs remarkably better than its quick and dirty counterpart, i.e. Naive Bayes, it might not be the most accurate model to predict the dependent variable, as a quarter of the predicted value is false. An explanation for this might be that the data might not be structured linearly - to tackle this, we might want to try support vector machine (SVM) with nonlinear kernels; another explanation might be that students perform differently across different exercises, and this is not factored into our analysis - to tackle this, we can make use of the hierarchical property of the exercises (exercise $\rightarrow$ exercise_type $\rightarrow$ problem_number of mixed model analysis, or include the exercise properties as predictor features in our regression.

5.3. The Flaws of Naive Bayes

In the context of our experiment, we assume that the 5 selected variables are conditionally independent given the student's performance. This is a rather hasty assumption. As an example , `hints` and `time_taken` might be correlated, given that it takes time for the students to read the hints' content. This results in the low accuracy rate for Naive Bayes analysis.

5.4. Intrinsic Skewness within the Dataset

There are less than 10% of the students who did more than 10 questions in the given dataset, 30% of which did only 1 question, giving a rather skewed dataset.

5.5. Conclusion

The variables that are the most useful in predicting students' performance are `review_mode`, `hints`, `topic_mode`, `num_problem`, `time_taken`, `num_coaches`, and `time_done`. Between logistic regression and Naive Bayes, logistic regression with threshold 0.8 is the best model. Therefore, we conclude that we would use logistic regression with threshold 0.8 on `review_mode`, `hints`, `topic_mode`, `num_problem`, `time_taken`, `num_coaches`, and `time_done` to predict future students' performance given the sets of features in the dataset.

6. Future Work and Vision

To perform more accurate analysis, there are several methods pertaining to our dataset that we can possibly employ, as follows:

1. Incorporate the significance of problem order into the analysis, there might be underlying patterns that plays important role in data prediction
2. Use mixed effect model to take care of the hierarchical structure of the dataset, after giving appropriate treatment to the raw data
3. Use pricipal component analysis for more accurate feature selection
4. Locally weighted logistic regression instead of the usual logistic regression
5. Use support vector machine - this is good since kernel can be selected precisely, as we face underfitting problem with current set of methods

Our work in this project will also be presented to the Graduate School of Education for the EDUC 407X Seminar in the winter quarter.

7. Acknowledgement

We would like to express our utmost gratitude towards Prof Andrew Ng and all the teaching assistants of CS 229 who provided valuable feedback to the project along the way. We would also like to Dr. Joseph Jay Williams from the Graduate School of Education for his continuous high-level guidance on the project.

Note: One of our members, Peng Hui How is enrolled in the EDUC 407X seminar this quarter and is using this work for the class.

References

[Mansinghka] Mansinghka, V.K., Shafto, P., Jonas, E., Petschulat, C., Gasner, M., Tenenbaum, J.B. (accepted pending revision). CrossCat: A fully Bayesian nonparametric method for analyzing heterogeneous, high dimensional data. Journal of Machine Learning Research study.

[IDRE] Institute for Digital Research and Education, UCLA. Annotated SPSS Output - SPSS `http://www.ats.ucla.edu/stat/spss/output/logistic.htm`

[ICS] Donald Bren School of Information and Computer Sciences, UCI. Decision Theory, Naive Bayes, ROC Curve `http://www.ics.uci.edu/~welling/teaching/ICS273Awinter11/DecTh273Awinter11.pdf`