Predicting future efficency during an adaptive test by previous actions

Yanlin Li (1003770305)

Abstract

This project is an exploration of how the factor efficiency in using time during a test can be predicted by actions when doing previous sections. Using a specific test as an example, we extracted action and time information from the original data and fitted them into a logistic ridge regression model. An accuracy of 71.98% was achieved at the end. The model and the approach can be used in designing an adaptive test.

Keywords: Adaptive testing, Efficiency, Action, Test, Logistic Ridge Regression

Introduction

Recently, the idea of adaptive testing is becoming increasingly popular. The idea is that the system can study the students' actions or outcomes when doing previous sections of a test. (Adaptive Testing FAQ) By modeling and analyzing the performance, the test can automatically adjust the difficulty or time allowed for the next sections so that the students can better handle their test questions. The score is based on both accuracy and the chosen difficulty. This method can give a more accurate assessment of the students' academic ability because the effort used are the same for each student.

Tests such as GRE (Graduate Record Examinations) and GMAT (Graduate Management Admission Test) are now using this technique. However, they still have some limitations. First, they use only the accuracy in the previous sections to adjust the difficulty in the following sections, which may result in a much too simple model. This model can hardly reflect the students' levels of academic ability. Second, the difficulty of next session can easily reflect the accuracy or score they already had and influence their emotions during tests, which may cause bias in academic ability evaluation. Consequently, adding more variables in the model is essential.

Actually, whether a student uses his or her time efficiently during the test is also an important factor in academic ability. Efficient usage of time of problem-based tests is defined as:

1. being able to complete all problems within the time limit, and
2. being able to allocate a reasonable amount of time to solve each problem.

This definition is specified on the NAEP Data Mining Competition 2019 (2019) website. Adding this variable can solve the above problem and can better homogenize students' effort used in the test. Thus in this project, we will take a math test as an example and explore how students' actions when doing previous sections of problems can affect their efficiency of using time in the next section.

The dataset we will explore is provided by the Educational Testing Service and was accessed from NAEP Data Mining Competition 2019 (2019). This dataset includes a compilation of actions students made during a test in the 2016-2017 academic year. During the test, students worked on "blocks" (same as "sections") of test math problems in a limited time of 30 minutes per block. Given information about the students' action when doing block A (the first section), the goal is to predict whether they spent time efficiently in block B (the second section). This prediction can be used in evaluating students' efficiency in using time during the next section.

The significance of this project is that when implementing same adaptive tests, the trained model can classify students into different levels of time allocation ability and distribute their questions. In a broader context, this example will serve as a proof that previous actions can be used to predict efficiency and can be considered as a factor in other problem-based adaptive tests. For simplicity, in this project, we will use only two levels of efficiency, which is efficient and not efficient. Similar and maybe more complex models can be used for other adaptive tests in practice.

In the next following sections, the dataset and model will be introduced and analyzed and predictions will be made for assessment. In the Methodology section (Section 2), the data will be introduced and the model used in prediction will be described. Results of the prediction and model details will come in the Results section (Section 3). Inferences of the data, some conclusions, weaknesses and next step can be found in the Conclusion section (Section 4).

Methodology

Data

General introduction

The data is provided by Educational Testing Service and was collected during a test in the 2016-2017 academic year. The data was collected when students were doing math problems in test. Because this was a computerized exam, their actions when doing the problems could be recorded by the system and was compiled into a single "action data" (it will be introduced below). The time when they were doing each actions were also recorded and included in this data. Altogether, 1232 observations were documented, which includes action information for 1232 students.

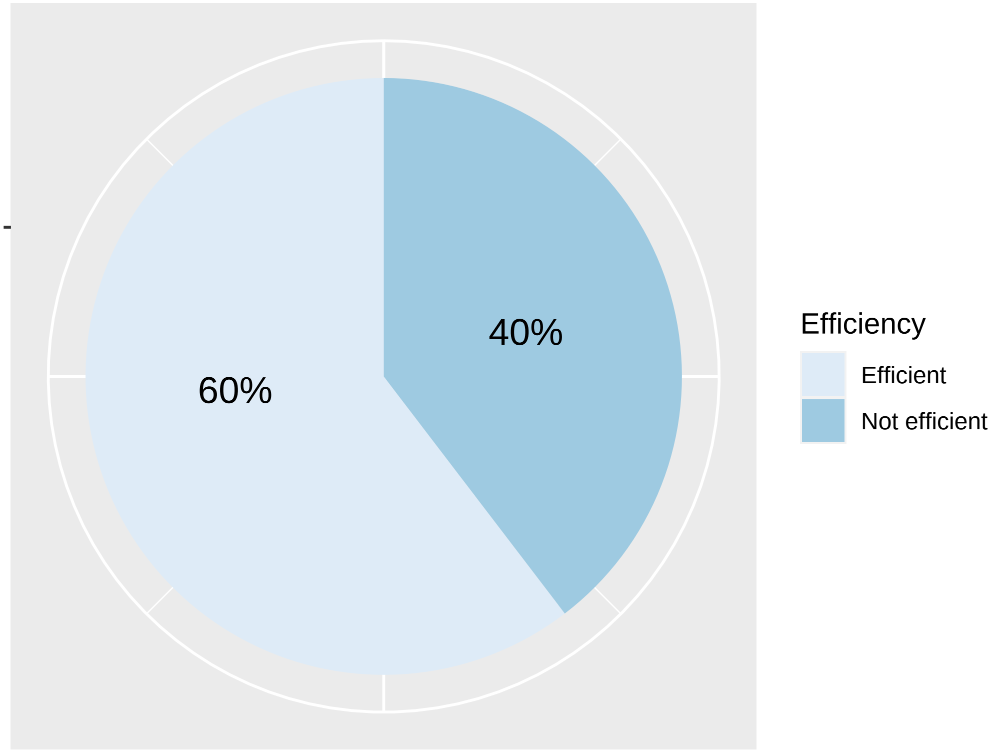
The sample in this case is all the students who took the test during 2016-17 academic year. Sampling methods of this data should be considered in two different contexts. In a narrow context, which means that our goal is to develop an efficiency prediction model for this specific test, the sampling method is cluster sampling. A cluster is all the students who took this test in a specific academic year. A census was done for a randomly chosen academic year, which was 2016-17. The target population is all the students who have taken this test. The sampling frame is a list of students who have taken the test in different academic years. In a broader context, this project is designed for exploring the general impact of previous actions on future efficiency, so that efficiency in allocating time can be modeled into all adaptive testing. In this case, the sampling method is non-probability sampling (convenience sampling), which may cause bias (Go for "Discussion - Weaknesses and Next steps" section for more detail). The target population is all the students who have taken an adaptive test.

Target variable and action data

It was mentioned in the introduction session that the students worked on "blocks" ("sections") of test math problems, referred to Block A and B. The goal is to use the action data in Block A to predict whether the student used time efficiently in Block B. The targeted variable (or: "response variable") is a binary indicator of whether or not the student spent their time in Block B efficiently. According to the definition mentioned in the NAEP Data Mining Competition 2019 (2019) website, the targeted variable will be calculated by two parts:

1. A binary variable indicating the whether the student completed all the problems in Block B;
2. Binary variables for each problem indicating whether or not they had allocated a reasonable amount of time for them. The term "reasonable amount of time" means the minimum possible time needed to solve each problem. The threshold was chosen based on the distribution of the total amount of time students spent on each problem in the dataset, with a 5th percentile cut-off.

The algorithm was not explicitly provided by the NAEP Data Mining Competition 2019 (2019) website and all the targeted variable data were already pre-processed for use. Here is a graph (Graph 1) of the distribution of the targeted variable.



Graph 1 : Proportion of efficiency

The action data is a list of data frame containing some details about students' action during the test. The first 4 rows of the first student's action data looks like this (Table 1):

Table 1: First student's action data

STUDENTID	Block	AccessionNumber	ItemType	Observable	ExtendedInfo	EventTime	Time
2.333e+09	A	VH356862	Directions	Enter Item		2017-02-10 14:44:54.167	0.000
2.333e+09	A	VH356862	Directions	Next		2017-02-10 14:45:01.263	7.096
2.333e+09	A	VH356862	Directions	Exit Item		2017-02-10 14:45:01.280	7.113
2.333e+09	A	VH098810	MCSS	Enter Item		2017-02-10 14:45:01.437	7.270

The definitions of each columns are:

1. STUDENTID: A unique identifier for each student which (to the best of our ability) cannot be traced back to individual students Block: The block that the action happened in. Logged actions are only provided for Block A.
2. AccessionNumber: A unique identification of a problem/item.
3. ItemType: The type of the item, including:
   - MCSS: Multiple-choice single-selection question.
   - GridMS: Grid-style multiple-selection question. Multiple MCSS presented in a grid. All MCSS share the same set of choices, differing only the questions
   - MatchMS: Match multiple-selection question. Students drag different values to multiple pre-defined blanks/locations to complete expressions, equations, etc.
   - ZonesMS: Multiple selection of pre-defined zones. Students click all the zones(images) that answer the questions.

- FillInBlank: Single fill in the blank question.
- MultipleFillInBlank: Multiple fill in the blank question.
- CompositeCR: Composite constructed response question. Students fill in one or more blanks, usually open ended. This could be paired with other minor questions e.g. drop down
- BQMCSS: Background/Survey MCSS question.

4. Observable: The type of the action the student took.
5. ExtendedInfo: Additional information on the student action (e.g. which choice the student clicked)
6. EventTime: The timestamp of when the action was taken.
7. Time: The length of time since the start of the first action.

Data Preprocessing and feature identification

Some pre-processing steps are necessary to make our data clean. In order to evaluate the model that we will use, the data is separated into training set and test set (1000:232). The response variable is regulated into 1 (Efficient) and 0 (Not efficient) for the convenience of regression.

Since the action data is a large list of data frame, it cannot be fitted into a model easily. Thus, we need to dig into the raw data and identify some important features from the actions in Block A that can affect the efficiency of block B. In other words, going beyond the original format and creating a clean data that is more friendly to modeling. **The details of feature identification here can be seen in Appendix 1: Action data feature identification.**

Here is the first 6 rows and 5 columns of the final data that will be used to build the model (Table 2):

Table 2: Cleaned data

	numact	VH356862.Directions	VH098810.MCSS	VH098519.MCSS	VH098808.MCSS
2333000033	0.69	0.32	0.35	0.29	0.74
2333000882	0.69	0.36	0.25	0.36	0.19
2333000955	0.69	0.46	0.26	0.36	0.51
2333001552	0.69	0.87	0.28	0.21	0.46
2333001935	0.69	0.63	0.27	0.43	0.54
2333001986	0.02	0.46	0.38	0.94	0.33

There are 25 features in total. The first feature ("numact") indicates the normalized value of total actions a student has done. The other column names ("VH356862 Directions", "VH098810 MCSS", . . .) are all the accession numbers (i.e. all action names), together with their item types. Their values are the normalized total time used on that action.

Now, both targeted variable and cleaned action data are ready for modeling.

Model

Model selection

The target variable is binary so this should be a classification problem. A logistic regression can be suitable in this case. However, with a total of 25 features and only 1000 training data, a simple logistic model may result in overfitting (which means that the model performs well on training data but poor on testing data). Consequently, Logistic Lasso or logistic ridge regression can also be candidates of models.

Introduction to these three candidates of models can be seen in Appendix 3: Model introduction. The evaluation methods used in model selection is in Appendix 2: Model performance

evaluation. Make sure to read them through.

Here is a table (Table 3) about the performances of the three models:

Table 3: Performances of the models

	Aggregate Score	Adjusted Kappa	Adjusted AUC	Accuracy
Logistic Regression	0.5184	0.2537	0.2647	0.6164
Ridge	0.8157	0.3798	0.4359	0.7241
LASSO	0.7326	0.3440	0.3886	0.7069

From the table (Table 3), it is clear that ridge regression dominates all the other models and performs the best in prediction. We will choose ridge regression to be our model.

Model introduction and discussion

Logistic ridge regression is a kind of penalized logistic regression model. A penalty term is introduced to limit the size of coefficients. This process can efficiently avoid overfitting and increase the accuracy for prediction.

The model is the same as logistic regression:

$$ln(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_{26}$$

where

1. p_i: The probability that the ith student can use time efficiently in Block B
2. $\beta_1, ..., \beta_n$: coefficients of the model
3. β_0: intercept of the model
4. $x_1, ..., x_{26}$: features in action data for the ith student. They are: numac, VH356862, VH098810, VH098519, VH098808, VH139047, VH098759, VH098740, VH134366, VH098753, VH134387, VH098783, BlockRev, VH098812, VH139196, VH134373, VH098839, VH098597, VH098556, VH098522, VH098779, VH098834, EOSTimeLft, SecTimeOut, HELPMAT8. (Item types are not included for simplicity.)

The log-odds transform $ln(\frac{p_i}{1-p_i})$ on the left hand side can deal with binary variables and use the probability to do prediction.

The model can be calculated by a minimization problem:

$$min_{\beta_0,\beta} \sum_{i=1}^{m} [y_i ln(p_i) + (1-y_i) ln(1-p_i)] + \lambda \sum_{j=1}^{n} \beta_j^2$$

where

1. λ is a constant penalty term (should be found by cross validation, which is a method of separating train data and use a part of it to test different λ's and find the best one).
2. y_i: the real value of whether the ith student can use time efficiently in Block B. (values can be 0 or 1)

The package and function `glmnet` will be used to run the regression. The best λ is chosen by cross validation, which can also be done by the `glmnet` package. The coefficients for the best model can be calculated using the best λ. Plug in these coefficients (rounded to 2 decimal spaces) from the output and we can get the final model equation (Item types are not presented for simplicity):

$$ln(\frac{p_i}{1-p_i}) = -5.03 + 0.35numac + 1.08VH356862 + 0.05VH098810 + 0.66VH098519 + 0.06VH098808$$

$$+0.40VH139047 + 0.23VH098759 + 0.13VH098740 + 0.28VH134366 + 0.38VH098753 + 0.28VH13437$$

$$+0.47VH098783 - 0.52BlockRev + 0.38VH098812 + 0.13VH139196 + 0.78VH134373$$
$$+0.38VH098839 + 0.05VH098597 - 0.04VH098556 + 0.31VH098522 + 0.60VH098779$$
$$+0.83VH098834 + 1.20EOSTimeLft + 3.41SecTimeOut - 0.44HELPMAT8$$

The optimal λ is 0.06.

Results

The prediction by the logistic ridge regression model achieves an accuracy of 71.98%. The Aggregate score is 0.7945. These statistics means that when we are using this model for a prediction of a student's future efficiency during an adaptive test, the result will be correct in about 71.98% of the cases.

About the details of the model, here is a table (Table 4) of all the coefficients for different actions, together with their item types.

Table 4: Coefficients of logistic ridge regression model

actions	itemtypes	coefficients
(Intercept)	(Intercept)	-4.48
numact	numact	0.30
VH356862	Directions	0.98
VH098810	MCSS	0.05
VH098519	MCSS	0.58
VH098808	MCSS	0.06
VH139047	MatchMS	0.34
VH098759	MCSS	0.22
VH098740	MCSS	0.13
VH134366	MultipleFillInBlank	0.24
VH098753	MCSS	0.34
VH134387	FillInBlank	0.26
VH098783	MCSS	0.43
BlockRev	BlockReview	-0.52
VH098812	MCSS	0.36
VH139196	CompositeCR	0.14
VH134373	FillInBlank	0.71
VH098839	MCSS	0.36
VH098597	MCSS	0.07
VH098556	MCSS	0.00
VH098522	MCSS	0.30
VH098779	MCSS	0.56
VH098834	MCSS	0.75
EOSTimeLft	TimeLeftMessage	1.06
SecTimeOut	TimeOutMessage	2.96
HELPMAT8	Help	-0.39

This table below (Table 5) classifies the actions by range of their coefficients. Accession number of a certain action will be classified into a certain range if and only if this action has a coefficient within that range.

Table 5: Actions with coefficient ranges

range	actions
coef < -1	(Intercept)
-1 < coef < 0	BlockRev,VH098556,HELPMAT8
0 < coef < 0.1	VH098810,VH098808,VH098597
0.1 < coef < 0.3	VH098759,VH098740,VH134366,VH134387,VH139196,VH098522
0.3 < coef < 0.5	numact,VH139047,VH098753,VH098783,VH098812,VH098839
0.5 < coef < 1	VH356862,VH098519,VH134373,VH098779,VH098834
coef > 1	EOSTimeLft,SecTimeOut

Because the accession numbers can be too abstract to understand, here is a table (Table 6) for the classification of item types with respect to different range of coefficients. A certain item type will be classified into a certain range if and only if at least one action of this type has a coefficient in that range.

Table 6: Itemtypes with coefficient ranges

range	itemtypes
coef < -1	(Intercept)
-1 < coef < 0	BlockReview,MCSS,Help
0 < coef < 0.1	MCSS
0.1 < coef < 0.3	MCSS,MultipleFillInBlank,FillInBlank,CompositeCR
0.3 < coef < 0.5	numact,MatchMS,MCSS
0.5 < coef < 1	Directions,MCSS,FillInBlank
coef > 1	TimeLeftMessage,TimeOutMessage

where:

- MCSS: Multiple-choice single-selection question.
- MatchMS: Match multiple-selection question. Students drag different values to multiple pre-defined blanks/locations to complete expressions, equations, etc.
- FillInBlank: Single fill in the blank question.
- MultipleFillInBlank: Multiple fill in the blank question.
- CompositeCR: Composite constructed response question. Students fill in one or more blanks, usually open ended. This could be paired with other minor questions e.g. drop down

Here is a table (Table 7) about the coefficient information for different item types.

Table 7: Distribution of coefficients for different item types

item types	number of actions	minimum	maximum	sd	mean
(Intercept)	1	-4.48	-4.48	NA	-4.48
BlockReview	1	-0.52	-0.52	NA	-0.52
CompositeCR	1	0.14	0.14	NA	0.14
Directions	1	0.98	0.98	NA	0.98
FillInBlank	2	0.26	0.71	0.32	0.48
Help	1	-0.39	-0.39	NA	-0.39
MatchMS	1	0.34	0.34	NA	0.34
MCSS	14	0.00	0.75	0.23	0.30
MultipleFillInBlank	1	0.24	0.24	NA	0.24

item types	number of actions	minimum	maximum	sd	mean
numact	1	0.30	0.30	NA	0.30
TimeLeftMessage	1	1.06	1.06	NA	1.06
TimeOutMessage	1	2.96	2.96	NA	2.96

The columns:

1. "number of actions": number of actions of this type.
2. "minimum" and "maximum": the smallest and largest coefficients for the type.
3. "sd": standard deviation of the coefficients if there are more than one action of this type; NA otherwise.
4. "mean": the mean of all coefficients for this type.

Discussion

Summary

In the above sections, we contributed to establish an efficiency prediction model for a certain adaptive test. The data is a compilation of actions students made during testing in the 2016-2017 academic year. The goal is to predict the relationship between actions taken in Block A (the first section) and efficiency usage of time in Block B. An introduction to the topic was made first in the Introduction section. Then, in the Methodology section, the data including action data and targeted variable was presented and cleaned. From the messy data, useful features were identified, which are the total number of actions done and the amount of time spent on each action. Then logistic ridge regression was chosen from the three candidates of models because of its dominance in performance. The cleaned data was fitted into the model and the final model was also specified. In the Result section, the performance of the model (accuracy: 71.98%, aggregate score: 0.7945) was given and some tables about the coefficients were presented.

Conclusions

The data we used, which was provided by Educational Testing Service allows us to make some inference about efficiency prediction in adaptive tests. Since for most existing adaptive tests, efficiency usage of time has not been modeled into the system yet, this project can provide a brief picture about this procedure.

With a high accuracy of 71.98% and an aggregate score of 0.7945, the model performs well in predicting students' future efficiency of using time when doing this specific test. More broadly speaking, a similar model can also work well for math tests consist of multiple choice, fill in blanks, composite response and background questions. Consequently, efficiency prediction is possible for this kind of tests and the model can be applied to adaptive testing systems.

To dig deeper into the model, we would also like to explore the meanings of the coefficients in that model. Their sizes and signs can reflect how this particular action may contribute to the future efficiency. To make the inference more general for application, we will use the term item types, instead of action names (accession numbers), for analysis. The item types can be separated into two categories, problems and non-problem actions.

We first explore actions for solving test problems. For multiple choice (single or multiple selection) questions, the range and standard deviation of coefficients are quite large. Most of the coefficients are positive. So spending more time on these questions can somehow result in an increase in future efficiency, but the influence is quite different for each questions. Because no sample question was provided, no further analysis can be made in this case. Text questions such as fill in blanks and composite constructed response has a small and positive impact on future efficiency. Spending a reasonable amount of time (and not too much) is appreciated.

Besides, non-problem actions also have great influence in future efficiency. Surprisingly, the total number of actions taken has only moderate impact on future efficiency. So whether a student has done all the actions cannot make a huge difference in future efficiency, although the more the better. Some non-problem actions such as block review and help have negative impact in future efficiency. Maybe because doing those actions indicates that the student is not familiar with the test (especially when clicking help). Other non-problem actions, including directions, time left and time out have strong positive impact on future efficiency. Doing these actions may indicates that a student has extra time after finishing all the required problems.

This model is accurate enough to be applied to similar adaptive tests with same types of questions and non-problem actions. The prediction can serve as a part of the student academic ability measurement. Students with a predicted high future efficiency of using time will do a more difficult test with less allowed time. At the same time, the range of scores will also be higher.

Weaknesses

Here is a list of several weaknesses of this project:

1. The model we used may be too simple. The efficiency prediction is a convoluted problem and a single logistic ridge regression cannot handle the functional terms in the model. The accuracy of prediction needs improvement.

2. There are some unused information in the original data, such as some details of actions. If we make good use of these information, the prediction can be more accurate.

3. In a broader context, we want to convince that previous actions can be used to predict future efficiency of using time, so that the efficiency can become a factor in all adaptive tests. However, this project is only a convenience sample of all the adaptive tests. This non-probability sampling technique can cause bias in the results, because the situation can be different for other types of tests such as writing essays and language exams. Consequently, this project can only serve as an example of modeling actions to predict efficiency.

Next steps

Here are the analysis that we can do in the future:

1. Improve the prediction accuracy. We can try other models such as Generalized Additive model to model functional terms. Maybe some other machine learning models such as neural network are also useful.

2. Dig deeper into the data provided. Identify more patterns from the unused data and add them into the model.

3. Include students' data collected from other types of adaptive tests, such as language exams, physics tests and essay writing. Use multi-stage cluster sampling. That is, randomly pick a test from each category of tests and randomly choose one academic year. Census can be done for all chosen tests in that academic year. Same analysis as in this project can be done for all the samples. If all the predictions provide good results, we can say that it is possible for all adaptive tests to add efficiency level adjustment.

Reference

1. NAEP Data Mining Competition 2019. (2019). Retrieved December 17, 2020, from https://sites.google.com/view/dataminingcompetition2019/

2. Adaptive Testing FAQ. (n.d.). Retrieved December 17, 2020, from https://www.testpartnership.com/adaptive.html

3. Uoftlibraries. (2020, December 09). University of Toronto Libraries. Retrieved December 17, 2020, from https://login.library.utoronto.ca/

4. Prabhakaran, S. (2017). Welcome to r-statistics.co. Retrieved December 17, 2020, from http://r-statistics.co/

5. Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33(1), 1-22. URL: http://www.jstatsoft.org/v33/i01/.
6. Yihui Xie (2020). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.30.
7. Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez and Markus Müller (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. BMC Bioinformatics, 12, p. 77. DOI: 10.1186/1471-2105-12-77 http://www.biomedcentral.com/1471-2105/12/77/
8. David Meyer, Achim Zeileis, and Kurt Hornik (2020). vcd: Visualizing Categorical Data. R package version 1.4-8.
9. Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686
10. Kun Ren (2016). rlist: A Toolbox for Non-Tabular Data Manipulation. R package version 0.4.6.1. https://CRAN.R-project.org/package=rlist
11. Douglas Bates and Martin Maechler (2019). Matrix: Sparse and Dense Matrix Classes and Methods. R package version 1.2-18. https://CRAN.R-project.org/package=Matrix
12. David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2020). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-4. https://CRAN.R-project.org/package=e1071
13. Hadley Wickham and Dana Seidel (2020). scales: Scale Functions for Visualization. R package version 1.1.1. https://CRAN.R-project.org/package=scales
14. Narkhede, S. (2019, May 26). Understanding AUC - ROC Curve. Retrieved December 19, 2020, from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
15. Narkhede, S. (2019, August 29). Understanding Confusion Matrix. Retrieved December 19, 2020, from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62
16. McHugh, M. (2012). Interrater reliability: The kappa statistic. Retrieved December 19, 2020, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3900052/
17. Martin, W. (2019, November 07). Cohen's Kappa in R: Best Reference. Retrieved December 19, 2020, from https://www.datanovia.com/en/lessons/cohens-kappa-in-r-for-two-categorical-variables/
18. Max Kuhn (2020). caret: Classification and Regression Training. R package version 6.0-86. https://CRAN.R-project.org/package=caret
19. Bhattacharyya, S. (2020, September 28). Ridge and Lasso Regression: L1 and L2 Regularization. Retrieved December 19, 2020, from https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b
20. Wikipedia. (2020, December 12). Tikhonov regularization. Retrieved December 19, 2020, from https://en.wikipedia.org/wiki/Tikhonov_regularization
21. Grolemund, H. (2017). R for Data Science. Retrieved December 20, 2020, from https://r4ds.had.co.nz/index.html

Appendix

1. Action data feature identification

There are no existed model that can automatically extract useful information from data of this format, so we should think about the link intuitively. In most cases, a student will keep their problem solving behavior when doing different sections. We can extract data relevant to efficiency in Block A, so the definition of efficiency (in “Introduction”section) can be useful here. Two assumptions deductible from the definition are made for feature selection:

1. All the accession numbers I found in training set indicates all the actions made in the block.
2. There are only two things that may affact the efficiency:
    - The percentage of all actions he did (different from the target variable, I used all actions instead of all problems, because actions such as going forward and back, ending the section can also reflect something about efficiency.);
    - The total time a student used in each action, compared with other students.

From the assumption, what will affect the results are total time a student took to make each action and the total actions a student made. In that case, all discovered accession numbers in the training set will become feature names of the data. The values will be the total time a student spent on each action. Besides, the total problem a student did is also a feature. The time limit of 30 minutes is also added into consideration. If any operations were done out of that limit, this action will not be considered. Finally, each feature will be normalized using normal CDF (the value equals to the percentile of normal distribution), so the number will be between 0 and 1.

2. Model performance evaluation

We usually measure the model performance by accuracy of classification. However, this measure can be misleading when two categories are not similar in sizes. In this dataset, 60% of the students are efficient in using time. This can result in a 60% classification accuracy even if we use a models that just predict everyone as efficient. Although the accuracy indicates that this works fairly well, this model is actually meaningless. So in addition to accuracy, we need to identify a new method of model performance evaluation. Here is the definition:

Aggregate score:

- $Adjusted\ AUC = 0$, if $AUC < 0$; $2(AUC - 0.5)$, otherwise
- $Adjusted\ Kappa = 0$, if $kappa < 0$; $kappa$, otherwise
- $Aggregate\ Score = Adjusted\ AUC + Adjusted\ Kappa$

There are two terms that we need further explanation:

1. AUC (Area under ROC curve):

    ROC is a curve with FPR (False positive rate) on the x-axis and TPR (True positive rate) on the y-axis.

    $$TPR = \frac{True\ positive}{True\ positive + False\ negative}$$

    $$FPR = \frac{False\ positive}{True\ negative + False\ positive}$$

    with positive class as the one with less cases. AUC is the area under the ROC curve. (Narkhede, 2019)

2. Kappa:

    Kappa can be used to test interrater reliability. Rater reliability represents the extent to which the data collected in the study are correct representations of the variables measured. (McHugh, 2012) This statistic has range -1 to 1. The formula is:

    $$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)}$$

    where $Pr(a)$ represents the actual observed agreement, and $Pr(e)$ represents chance agreement.

3. Model introduction

Logistic regression, logistic ridge regression and logistic lasso all have the same final format. For a regression with n features and m data points (in this data, $n = 25,\ m = 1000$) $x_1, ..., x_n$ to model, the model is

$$ln(\frac{p}{1-p}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

where

1. p: a vector of probability of events. $p = \begin{pmatrix} p_1 & \cdots & p_m \end{pmatrix}$
2. $\beta_1, ..., \beta_n$: coefficients of the model
3. β_0: intercept of the model
4. $x_1, ..., x_n$: vectors of feature values. $x_i = \begin{pmatrix} x_{1i} & \cdots & x_{ni} \end{pmatrix}^T$

Here are a few matrix representations of the terms before. Let:

1. $X = \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix}$ be the matrix of data.
2. $\beta = \begin{pmatrix} \beta_1 & \cdots & \beta_n \end{pmatrix}^T$
3. $y = \begin{pmatrix} y_1 & \cdots & y_m \end{pmatrix}^T$ is the observed value of the targeted variable.

The above equation can be rewritted as

$$ln(\frac{p}{1-p}) = \beta_0 + X\beta$$

We use optimization to get the values of $\beta_0, \beta_1, ..., \beta_n$. For three different models, this optimization problem is different:

1. Logistic regression:

$$min_{\beta_0,\beta} \sum_{i=1}^{m} [y_i ln(p_i) + (1 - y_i) ln(1 - p_i)]$$

2. Logistic ridge regression:

$$min_{\beta_0,\beta} \sum_{i=1}^{m} [y_i ln(p_i) + (1 - y_i) ln(1 - p_i)] + \lambda \sum_{j=1}^{n} \beta_j^2$$

   where λ is a constant penalty term.

3. Logistic lasso:

$$min_{\beta_0,\beta} \sum_{i=1}^{m} [y_i ln(p_i) + (1 - y_i) ln(1 - p_i)] + \lambda \sum_{j=1}^{n} |\beta_j|$$

   where λ is a constant penalty term.

By doing the minimization problems, we can get the optimal values of the coefficients and the final model.