Yang, J. C. et al. (Eds.) (2018). Proceedings of the 26th International Conference on Computers in Education. Philippines: Asia-Pacific Society for Computers in Education

# Predicting Student Test Performance based on Time Series Data of eBook Reader Behavior Using the Cluster-Distance Space Transformation

**Alexander ASKINADZE[a*], Matthias LIEBECK[a] & Stefan CONRAD[a]**
[a]*Heinrich Heine University Düsseldorf, Germany*
*askinadze@cs.uni-duesseldorf.de

**Abstract:** This paper describes our participation in the task of predicting student performance at the learning analytics workshop which is hosted at the ICCE2018 conference. The task provides two datasets consisting of student time series click data behavior from an eBook reader. The goal is to predict the score and to predict whether a student passes the course or not. We transformed the time series data of student eBook actions in different features for the regression and the classification task. Among many feature subsets examined, feature subsets that have emerged through t-test, f-regression, and random forest regression have delivered comparatively better results. After an extensive feature engineering, we tried a new approach, based on k-Means, which transforms the selected features into the cluster-distance space. We evaluated the original and resulting features with different classifiers and regressors. For both datasets and both problems (regression and binary classification), the feature sets created with the cluster-distance space transformation have delivered better results.

**Keywords:** educational data mining, learning analytics, student score prediction, deep feature synthesis, regression, machine learning, time series

## 1. Introduction

In recent years, the number of digital learning opportunities has been steadily increasing. Lectures can be viewed as interactive videos at any time, and learning can be promoted with quiz apps. Many different digital devices, such as augmented reality glasses, eBooks and many more, can be used to support the learning process.

A survey conducted by Weisberg (2011) showed that 83% of students would use a digital textbook on a tablet as the primary or secondary textbook and 91% would use a digital textbook on an e-reader as the primary or secondary textbook.

The use of digital devices and distributed learning environments such as eBooks results in usage data that can be tracked. The research area Learning Analytics (LA) deals with the "collection, analysis, and reporting of data about learners and their contexts" (Romero & Ventura, 2013), while the research area Educational Data Mining (EDM) "focuses on the development of methods for exploring the unique types of data that come from an educational context" (Romero et al., 2010).

Using EDM and LA techniques, the usage data of the students on eBooks could be used to create different data mining models that could predict student performance. Shahiri and Husain (2015) pointed out in their review of student performance prediction that the information provided from the data mining models can be important for educators to monitor the performance prediction and to optimize the teaching approach.

To track the data from the distributed learning environments such as eBooks, the xAPI (Experience API) specification (https://github.com/adlnet/xAPI-Spec) can be used, which describes, among other things, the format in which student interactions can be sent. The collected

data can be used to visualize students' usage patterns or to train data mining models (Figure 1) that can, for example, predict student performance or early dropout.

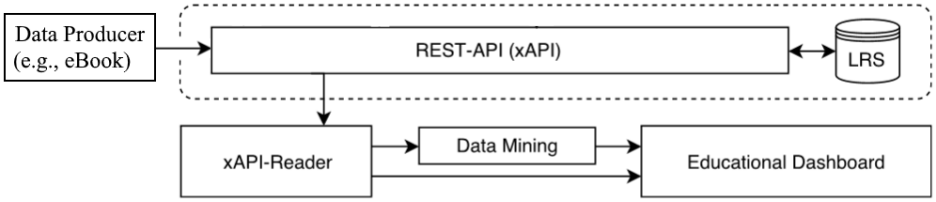


*Figure 1.* Using data mining on xAPI data (Askinadze & Conrad, 2017)

In order to make predictions about eBook reader clickstream usage data, approaches to predict student performance on time series data are particularly important. Research has already been conducted on making predictions based on student time-series data, for example, at the Workshop on Scientific Findings from the ASSISTments Longitudinal Data Mining Competition in the context of the Educational Data Mining Conference 2018, where various approaches to predict whether a student will pursue a career in STEM fields based on time series data were presented (Yeung, Lin, Yang & Yeung, 2018; Makhlouf & Mine, 2018).

In this paper, we focus on the shared task of predicting student performance in the learning analytics workshop which is hosted at the ICCE 2018 conference. The given dataset consists of eBook usage data (Ogata et al., 2015; Flanagan & Ogata, 2017) that were collected by xAPI statements from two different courses. For each student in the courses, there are time series data of his reading behavior and a final score. The goals are to predict the score and to predict whether a student passes the course or not. The task organizers asked the tasks participants to report two report evaluation metrics: root-mean-squared error (RMSE) as a metric for regression and area under the curve (AUC) as a classification metric. Since we optimize our hyperparameter search for a given metric, we decided to solve two machine learning problems independently and optimize for their respective metric: (1) binary classification task of predicting whether a student fails or passes a test; (2) regression task of predicting a student's final test score.

Predicting student performance based on reading behavior has been previously researched from Chau, Li and Lin (2017). They also focused on the prediction of a student's final grade. In contrast to our work, they pursued the binary classification problem of predicting whether a student will perform above average instead of passing or failing the course. Three fundamental differences from our work are: (1) their time series data was based on reading behavior that was tracked every 10 seconds instead of xAPI events; (2) they had access to the underlying eBook texts and were able to extract the number of read words per second; and (3) they were able to include the performance on multiple-choice questions from their eBook reader system into their machine learning features.

## 2. Dataset

The educational data mining shared task of predicting student performance comprises logged user behavior of an eBook system called BookRoll. In BookRoll, eBooks are displayed page by page. Users are able to perform multiple actions, called click data, including scrolling to the next page, scrolling to the previous page, jumping to a specific page, adding and removing bookmarks, adding and removing markers on text passages (with the two options of marking as "important" and marking as "not understood"), adding and deleting memos, clicking links, using keyword searches, and jumping to search results. BookRoll's user interface is visualized in Figure 2.


*Figure 2.* BookRoll as eBook system (Flanagan & Ogata, 2017)

The challenge comprises BookRoll's click data from two courses and the final test score of each student. Both courses differ regarding how many lectures the students had: The first course only had one intensive lecture, whereas the second course consisted of three lectures. The goal of the shared task is to predict the students' final test scores based on their eBook reader behavior as a regression task.

Each course covered three different eBooks. As a result, we cannot directly compare the students in both courses and we cannot infer insights from different course lengths since the books and, therefore, the finals tests were different. However, we can benchmark our machine learning approaches independently and aim to find a combination of a feature set and a classifier that works well for both datasets.

The first course was attended by 53 students and BookRoll's log contains 28826 points of click data. In the second course, 36929 click data points were logged from 55 students. For a machine learning problem, both datasets contain a very low number of points for training and evaluating a classifier. The threshold of passing or failing an exam is at 60 points. By looking at the violin plots of both courses in Figure 3, we can see that distribution is skewed toward students with a high number of points, which will aggravate our data sparsity problem.

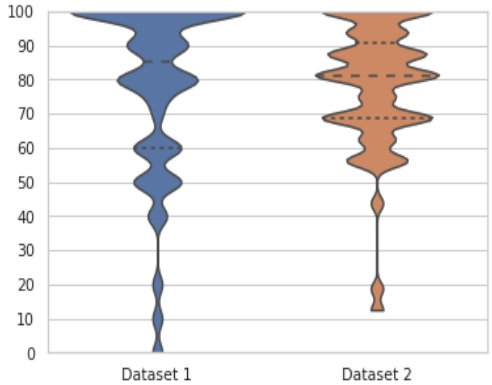


*Figure 3.* Violin plots of the students' scores

## *3.* Feature Engineering and Data Representations

In this chapter, we describe our feature engineering, discuss the feature distributions and relations to students' scores, and explain how we selected certain features and how we reduced the dimensionality of the feature vectors.

### *3.1 Feature Engineering*

Due to the small number of students, we decided to use classical machine learning methods and to transform the click data into a single vector for each student instead of sequentially passing click data into a recurrent neural network. In addition, the number of click data per student fluctuated greatly.

### *3.1.1 Deep Feature Synthesis*

In the next subsection, we show how we create a single vector for each student from time series data of his or her actions. Among other things, we were inspired by the Deep Feature Synthesis (DFS) approach (Kanter & Veeramachaneni, 2015), whose idea we will briefly describe.

DFS provides an approach to automatic feature engineering on relational datasets, that is, datasets that consist of multiple tables that are related to each other. This also applies to our dataset, as we have a table of students and their final grade and a table of their time series click data. Mathematical operations, which return a single value from a list of numeric values, are applied to numerical attributes within the time series data. Such operations include, for example, minimum, maximum, average, standard deviation and sum.

We have only partially used DFS for our feature vector by using the idea of applying the above operations to the time series data to get a single vector per student. Additionally, we also manually created features which we deemed useful. The details of our feature creation are outlined below.

### *3.1.2 Feature Creation*

We started to create a single feature vector for each student by grouping his or her click data. First, we one-hot-encoded the columns "action," "operation name," "marker color," device code," and "contentsid," which resulted in more columns since each possible value of a column is now represented by a binary column. For instance, instead of having a single column "action" that can contain one of the possible actions, e.g., *read or exited*, we now separate columns for each action in which only one value is 1 and all other values are 0. This results in columns for xAPI statements (launched, read, bookmarked, highlighted, noted, searched, exited), operation names (open, next page, previous page, page jump, search, search jump, add bookmark, delete bookmark, add marker, delete marker, add memo, change memo, delete memo, link clicked, close), device codes (pc, mobile, and tablet), marker usage (marked as important or as not understood), and columns for each of the three books. Then, for each student, we aggregated over numerical columns by calculating their sum, their mean and their standard deviation. For columns containing strings, e.g., "marker text" and "memo text," we reported their mean and standard deviation regarding character lengths.

Additionally, we derived multiple features that we thought could positively impact the classifiers:

- Jump distance: For the operation name "page jump," we inferred how many pages a student jumped by comparing the page number the student jumped from to the page number he or she jumped to. Our motivation for this feature was that the jump distance could be related to the learning behavior and the memory performance, which might impact the final test score. For this feature, we calculated the mean and the standard deviation.
- Unique page numbers: We noticed that some students did not read all the pages of a book and that students sometimes read specific pages multiple times. Therefore, we decided to track the number of unique read pages per book.
- Repetitive page reads: Also, we calculated how often each page was read per book and report the minimum (basically how often a book was read completely), mean (how often a page was read on average), and maximum amount (the number of times the most viewed page was read) per book.
- Different days: Based on the timestamp of the click data, we inferred on how many different days BookRoll was used by the student.
- Session time: We were also interested in how long the students used BookRoll. We grouped the xAPI statements per day and calculated an average session time and the maximum session time in seconds.
- Mean read time: To consider if a student is just skimming through the books or taking his or her time reading the pages, we calculated the read time between consecutive xAPI read statements. Then, we reported how many seconds a student needed to read a page. We

included this feature as a global feature, as well as individually, for each of the three books.

In the following, the vectorial representation with all features is denoted as $X_{All}$. The subset containing our derived features is called $X_{own}$.

After creating all of our features, we used the student's scores and their median to divide all students into two balanced groups representing the "lower half" or "upper half" of the class. Since the median is used as a threshold, each half consists of approximately 27 students, depending on the dataset. Then, we created boxplots for both groups and for each feature in order to visually estimate whether the distribution of the features might be useful for our machine learning approach. Figure 4 shows interesting feature distributions for the lower and the upper half of the students for the second dataset. The first subplot shows that students from the upper half performed more read operations on average than students from the lower half. Similarly, the number of click data for book 3 is also higher on average. There are also differences between the groups regarding the number of "previous page" clicks and the usage of memos.

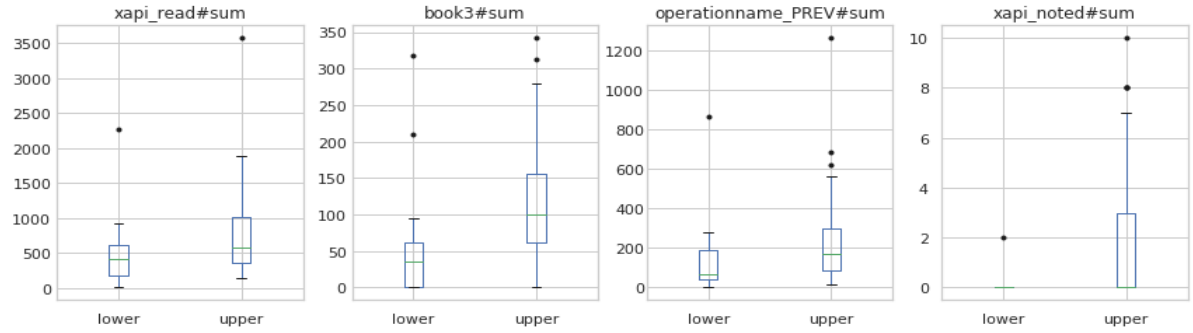


*Figure 4.* Boxplots of the distribution of four features for the second dataset

Afterwards, we calculated correlations between all of our features and the test score. Since we observed low correlation scores for many of our features, we can omit features and only focus on meaningful ones. Therefore, we will use feature selection techniques which are listed in the next subchapter.

## 3.2 Feature Selection and Dimensionality Reduction

We evaluated multiple feature selection techniques to select good features:

### *3.2.1 T-tests*

We used t-tests on the distributions for the lower and upper half of the students for each feature. Our null hypothesis is that the mean of the distribution of both groups is the same. The subset $X_{ttest}$ only contains features for which we were able to reject the null hypothesis with a significance level of $\alpha = 0.1$ for the first dataset and $\alpha = 0.05$ for the second dataset. The most important ten features are:

**Dataset 1**

1. operationname_PREV#sum
2. operationname_ADD_MARKER#mean
3. pageno#pages_same_max_book2
4. operationname_ADD_MARKER#std
5. book2#sum
6. marker_not_understood#std
7. xapi_read#sum
8. xapi_highlighted#std

9. pageno#pages_same_mean_book2
10. marker_not_understood#mean

**Dataset 2**

1. xapi_noted#std
2. operationname_ADD_MEMO#std
3. xapi_noted#sum
4. pageno#pages_same_mean_book3
5. operationname_ADD_MEMO#sum
6. pageno#pages_unique_book3
7. book3#sum
8. xapi_noted#mean
9. operationname_ADD_MEMO#mean
10. operationname_PREV#mean

Further information is available in our GitHub repository: https://github.com/askinadze/la-icce2018

### *3.2.2 Random Forest Regressor*

We trained a random forest regressor on the whole dataset. Then, we selected the 15 most important features and denoted them with $X_{RF}$. Because of the small number of data points in the datasets, we only used ten trees. If one selects a higher number of trees, the most important 15 features hardly change.

### *3.2.3 F-regression*

We performed univariate linear regression tests on the features and the score to select the $k$ most meaningful features as feature set $X_{fregression}$. Additionally, we combined the ten best features from $X_{ttest}$, $X_{RF}$, and $X_{fregression}$ into the feature set $X_{Best}$.

### *3.2.4 K-Means Transformation*

After looking at feature selection methods, we still noticed that the data points cannot be separated easily, e.g., with a hyperplane, in the feature space. Therefore, we decided to further reduce the dimensionality of our data points to reduce sparsity problems by projecting our data points onto the centroid space obtained by a k-Means clustering and representing each data point in the cluster-distance space, as proposed by Jeon (2001). As a result, each data point is a k-dimensional vector, where the i-th dimension represents the distance of the original data point to the i-th cluster centroid.

## 4. Evaluation

After describing our features, we can now report how they perform on the challenge's datasets. For each combination of feature set and classifier, we perform 10 times 3-fold cross-validations with different seeds. This allows us to report a mean and a standard deviation which allows us to judge the stability of our approaches. Since we consider the two different problems of regression (predicting the score) and binary classification (predicting whether a student passes or fails), different classifiers/regressors and different feature sets are used. The following classifiers and their respective regression versions were used:

- SVM with RBF kernel: For each run, a grid search on the $\gamma$and the $C$ parameters is performed
- Random forest classifier: For each run, a grid search on a different number of trees, different number of maximum depths, and different number of maximum features is performed
- KNN classifier: For each run, a grid search on a different number of nearest neighbors,

different number of weights (uniform, distance) and different types of Minkowski distance is performed

The following feature sets that are described in the feature engineering chapter are used:

- $X_{All}$ and its scaled version $X_{All-scaled}$
- $X_{Own}$ and its scaled version $X_{Own-scaled}$
- $X_{Best}$ and its scaled version $X_{Best-scaled}$
- Various subsets of $X_{fregression-scaled}$ are tested containing the 3,4,5, ... best selected features
- $X_{ttest}$ and its scaled version $X_{ttest-scaled}$
- $X_{RF}$ (15 best features found by the random forest regressor) and its scaled version $X_{RF-scaled}$
- K-Means transformed versions of all features set described above

For each evaluation, the three best combinations of classifiers/regressors and feature sets are presented in the next subchapters. For example the notation "SVR + X_fregression + kmeans" means that the feature set $X_{fregression-scaled}$ (with an optimized number of best features) is transformed into the cluster-distance space (with an optimized number of clusters found by k-Means) and then used for regression by an SVR .

### *4.1.1 Baseline for regression*

In order to gauge the effectiveness of our machine learning approaches, we decided to compare our results with two baselines which are shown in Figure 5: (1) predicting the mean of the gold labels; (2) average of ten random predictions.

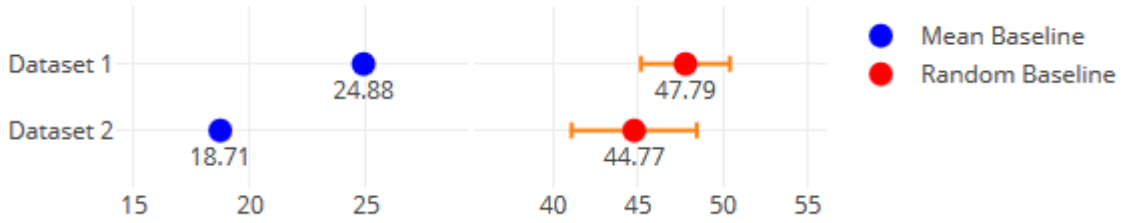


*Figure 5. RMSE baselines for both datasets*

### *4.1.2 Regression Results*

#### *4.1.2.1. Dataset 1*

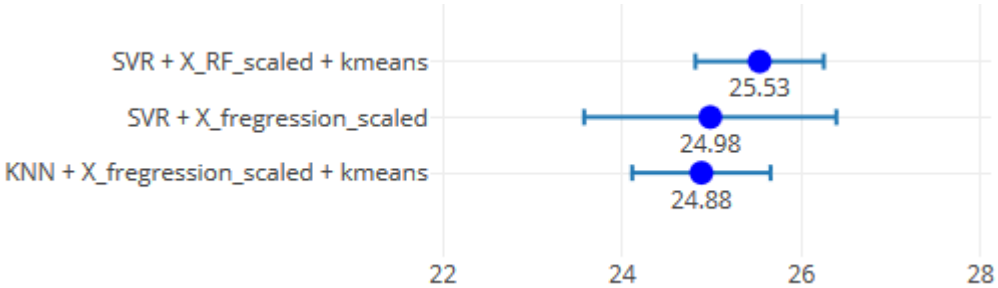


*Figure 6.* Best three regression results (RMSE) for dataset 1

The best result was achieved by the KNN regressor on the $X_{fregression-scaled+kmeans}$ feature set with an averaged RMSE score of 24.88 ($\mp$0.77), which is equal to our mean baseline.

#### *4.1.2.2. Dataset 2*

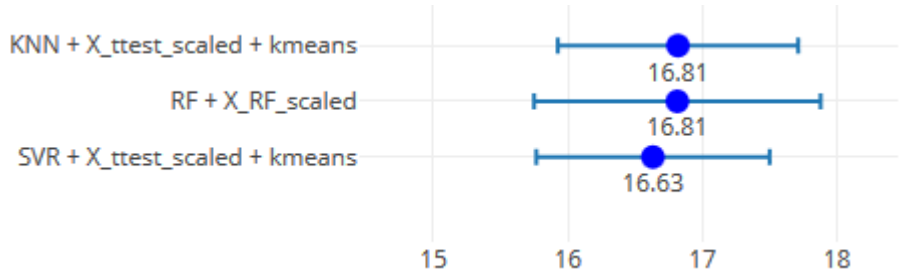


*Figure 7.* Regression results for dataset 2

For the second dataset, all three regressors were able to beat the mean baseline (see Figure 7). The best result was achieved by the random forest regressor on the $X_{fregression-scaled+kmeans}$ feature set with an averaged RMSE score of 16.63 ($\mp$0.87).

### *4.2 Binary Classification*

We now outline our results for the binary classification task of predicting whether a student will fail or pass the test. The gold labels for the binary classification task were derived by determining whether the gold score is at least as high as the passing score of 60. For both datasets, this resulted in skewed labels distributions, see Figure 3.

In Figure 8.1, we used the t-SNE method to visualize the two classes "passed" and "failed" based on the $X_{ttest}$ feature set. It can be seen that the classes are not simply separable from each other. By using our k-Means clustering approach, some elements of the "failed" class appear to be further away from the other elements. Therefore, we believe that using the cluster-distance can be advantageous. The visualization for the k-Means transformation of the feature set $X_{ttest}$ with 2 clusters is shown in Figure 8.2.

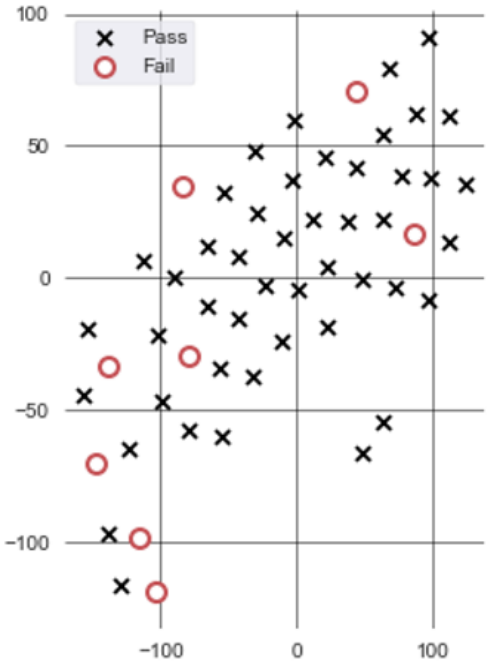


*Figure 8.1.* T-SNE data visualization

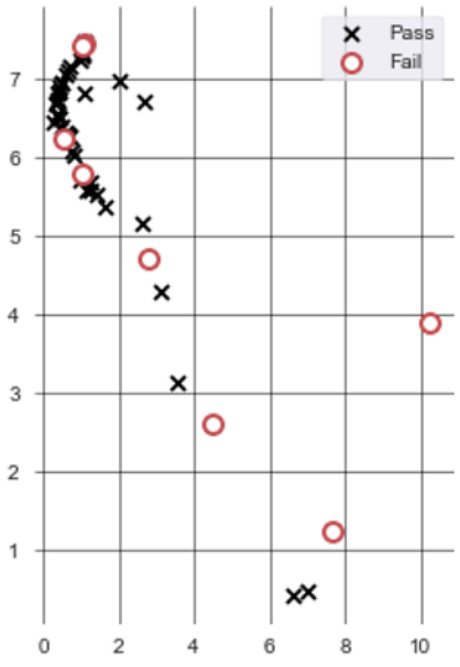


*Figure 8.2.* K-Means 2 clusters

*Figure 8.* 2D-visualization of $X_{ttest}$ for the binary classification problem

### 4.2.1 *Binary Classification Results*

#### 4.2.1.1. *Dataset 1*

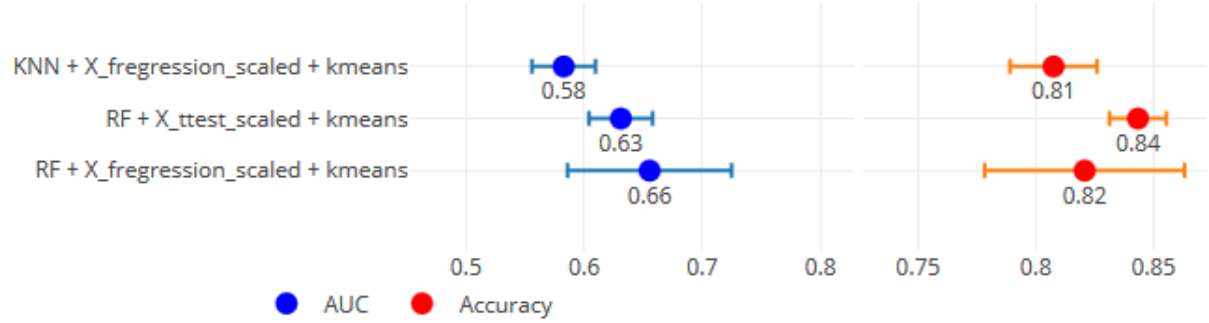


*Figure 9.* Best 3 AUC results for dataset 1

The best result was achieved by the random forest classifier on the $X_{fregression-scaled+kmeans}$ feature set with an averaged AUC score of 0.66 (∓0.07) (see Figure 9). The corresponding accuracy result for that setting is 82%.

#### 4.2.1.2. *Dataset 2*

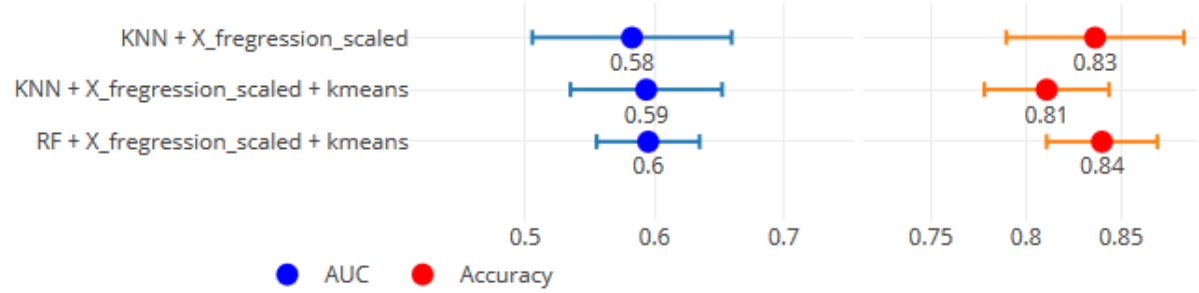


*Figure 10.* Best 3 AUC results for dataset 2

For the second dataset, the best result (see Figure 10) was also achieved by the random forest classifier on the $X_{fregression-scaled+kmeans}$ feature set with an averaged AUC score of 0.6 (∓0.04). The corresponding accuracy result for that setting is 84%.

### 4.3 *Further Evaluation Notes*

Additionally, we experimented with several oversampling techniques in order to balance the class distribution. Unfortunately, our results did not improve for either of the datasets.

## 5. Conclusion

We transformed time series data of student eBook actions in different features for the regression and the classification task. After an extensive feature engineering, we tried a new approach, based on k-Means, which transforms the selected features into the cluster-distance space. We evaluated the original and resulting features with different classifiers and regressors.

As our results show, it was difficult to solve the regression task for the first dataset (24.88 ∓0.77) for which we were not able to beat the mean baseline. We were only able to achieve better RMSE scores (16.63 ∓0.87) on the second dataset. We believe this behavior to result from the different score distributions of the dataset, as visualized in the violin plots in Figure 3. Regarding AUC, we achieved higher scores on the first dataset (0.66 ∓0.07) than on the second dataset (0.6 ∓0.04).

For both existing datasets and both problems (regression and binary classification), the feature sets created with the cluster-distance space transformation delivered better results than other models.

Therefore, we believe that the k-Means based approach can be helpful in cases where the data is difficult to separate. We will further investigate this approach in future work on other datasets.

## Acknowledgements

This work was partially funded by the PhD program *Online Participation*, supported by the North Rhine-Westphalian funding scheme *Fortschrittskollegs*.

## References

Askinadze, A., & Conrad, S. (2017, June). A web Service Architecture for Tracking and Analyzing Data from Distributed E-Learning Environments. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2017 IEEE 26th International Conference on* (pp. 208-213). IEEE.

Flanagan, B. & Ogata, H. (2017). Integration of Learning Analytics Research and Production Systems While Protecting Privacy. In *International Conference on Computers in Education (ICCE2017)* (pp. 333-338)

Jeon, M. (2001). *Centroid-based Dimension Reduction Methods for Classification of High Dimensional Text Data* (Doctoral dissertation).

Kanter, J. M., & Veeramachaneni, K. (2015, October). Deep feature synthesis: Towards automating data science endeavors. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on* (pp. 1-10) IEEE

Makhlouf, J., & Mine, T. (2018) Predicting if students will pursue a STEM career using School-Aggregated Data from their usage of an Intelligent Tutoring System. *Proceedings of the 11th International Conference on Educational Data Mining*, 533-536.

Ogata, H., Yin, C., Oi, M., Okubo, F., Shimada, A., Kojima, K., & Yamada, M. (2015). E-Book-based learning analytics in university education. In *International Conference on Computer in Education (ICCE 2015)* (pp. 401-406)

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

Romero, C., Ventura, S., Pechenizkiy, M., & Baker, R. S. (Eds.). (2010). *Handbook of educational data mining*. CRC press.

Shahiri, A. M., & Husain, W. (2015). A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, *72*, 414-422.

Weisberg, M. (2011). Student attitudes and behaviors towards digital textbooks. *Publishing Research Quarterly*, *27*(2), 188-196.

Yeung, C. K., Lin, Z., Yang, K., & Yeung, D. Y. (2018). Incorporating Features Learned by an Enhanced Deep Knowledge Tracing Model for STEM/Non-STEM Job Prediction. *arXiv preprint arXiv:1806.03256*.

Chau, H., Li, A., & Lin, Y. R. (2017). Predicting Students Performance Based on Their Reading Behaviors. *2017 International Conference on Social Computing, Behavioral-Cultural Modeling & Prediction and Behavior Representation in Modeling and Simulation*