COMP6237: Data Mining group Coursework - STEM or NOT STEM

Anton Okhotnikov
ao2u17@soton.ac.uk

Fangfei Liu
fl1m17@soton.ac.uk

Lakshmi Mukkawar
lpm1n17@soton.ac.uk

Qi Zhang
qz5n17@soton.ac.uk

Richa Ranjan
rr2n17@soton.ac.uk

Yeldos Balgabekov
yb2n17@soton.ac.uk

Yifang Zhou
yz5f17@soton.ac.uk

ABSTRACT

With the increasing demands for skilled workers in the Science, Technology, Engineering and Mathematics (STEM) fields, more and more students are getting themselves enrolled into STEM fields in their Under-graduation courses. STEM plays an important role in the development of modern society. Whether students choose this major or not, can be understood by monitoring their motivations and interests while they are at middle-school level. The aim of this project is to build a predictive analysis model that would predict if the students will choose STEM as their Under-graduation Major or not.

1 INTRODUCTION

The task chosen here is an *ASSISTment* Data Mining competition. ASSISTments is a free web-based mathematics tutoring system for middle-school mathematics, provided by **Worcester Polytechnic Institute** (WPI). By examining the middle-school data, we are predicting whether a student will pursue a career in **Science Technology Engineering and Mathematics** (STEM) or not.

The ASSISTments system has a straightforward design. Students use this system for mathematics classes. For solving each question, a student performs one or more actions. If a student solves it correctly, the system goes for the next question. When the student solves it incorrectly, the system *scaffolds* the problem in different parts and helps the student to solve sub-problems before solving the original problem. This has been recorded in the action log files which also includes students' behaviour while solving each question. Observing the actions of the students like time is taken to solve a particular problem, the number of correct solutions in a particular skill, etc. determine their inclination towards going forward with STEM major.

2 EXPLORING THE DATA

The data set was obtained from the *ASSISTments Datamining Competition 2017.* It contained data from over 360 colleges about students who used ASSISTments during their middle-school mathematics classes. The outcome is predicted from a particular studentfis interaction with the system. The data includes 10 action log files which are series of actions performed by students. It contains 76 different features related to the type of problem, correctly and incorrectly solved questions, student skills and behaviour etc. The Training dataset has columns like unique student id, whether student entered in STEM or not, average correctness, etc., with 500 observations. There are about 900,000+ lines of pupil actions in this data set. This contains the information about actions of individual students. There are 72 features with both continuous and categorical variables. In addition to these features, we engineered some more (discussed in detail in further sections). On investigation, we found some issues with the data, like **missing values**, **unbalanced data** and **miss-clicks**. Handling these was our next step.

3 PRE-PROCESSING

The first step towards pre-processing was to make the data consistent with training models. To handle missing values, we tried different approaches like taking extreme values, replacing with average values (or 0s in some cases). For example, we replaced the missing values with extreme values (like -999) for decision tree-based models as this is one of the commonly used approaches for handling noisy data while building decision trees [2]. Additionally, we tried replacing the missing values with the average values or 0s to train on models like Logistic Regression and SVM. To sum up, this was an empirical process and we handled the missing data depending upon the type of models used for training.

Another issue that we faced was unbalanced data. Out of the 500 observations spread across the action logs, only 164 corresponded to the ”is STEM” label, and the rest belonged to the ”non-STEM” category. To handle this, we performed *down-sampling* on the data. We connected the 164 observations to another 164 ”non-STEM” labelled data set and used this for further calculations. However, we trained our models using both down-sampled sets, as well as the unbalanced set. The observations of both approaches have been discussed in further sections.

Another issue with the data was *misclick*, which could have a negative effect on predictions. Any action for which ”timeTaken” is smaller than 2 seconds, was considered as a misclick action. The threshold of 2 seconds was inspired by some known features, for instance, time spent on help was under 2 seconds. There are 177430 misclick actions in total and the only solution was to remove all misclicks.

4 FEATURE ENGINEERING

The most crucial step in any learning process is the creation of features. While there were 72 features available in the data, we used the following methods to design additional features.

4.1 Association Analysis on Skills

As part of the Feature Engineering process, the top five skills were presented as a vital feature. Association Analysis, an unsupervised learning technique, that looks for hidden patterns in the data, was used for identifying patterns in the skills. The methodology used is described below:

- For each student, a threshold value(5 in this case) was decided, in order to consider only that many numbers of skills. A sparse matrix was created to identify the skills which belong to the top five categories.
- For the top five skills for each student, the respective *accuracies* were calculated, using the given *correct* and *total* actions values.
- For applying the Associativity rule, the data was required to be transferred from frame format into a single-row transactional format. This is done to club together the top five skills for a particular student.
- Using the `association_rules` package from the library `mlxtend` [1], the confidence levels for a student choosing STEM or NON-STEM was calculated. The `mlxtend` library provides the implementation of the *Apriori* algorithm. The confidence level was calculated based on the `min_threshold` value chosen (60% in this case). The probability of a student who possesses a particular skill (specified by `antecedants`) is given by the confidence value, which determines how likely a student is, to choose STEM or Non-STEM.

With the sparse matrix created above, the percentage of non-empty cells can be found, which gives the density of the number of skills possessed by a student, that is also one among the top five skills.

4.2 Bag Of Words

Another feature engineering technique used was the *Bag-of-Words* (BOW) model. This is commonly used in classification problems where the frequency of occurrence of each word is used as a feature for training a classifier. Each unique word will, therefore, correspond to a numerical feature. For this, the columns ”sy assistments usage”, ”skill” and ”problem type” fields were converted into categorical values, and these categories were further used as separate features. The *K-Means Clustering* method was used to group similar actions into words. These words were later used as features to determine the behaviour of each student. The hypothesis behind using this approach is that similar actions when grouped together to form clusters, serve as ”bags of words”, which can further be treated as a feature to determine a particular student’s choice of skills. Based on that selection, further deductions can be made if the student chooses STEM or NON-STEM fields.

4.3 Skills as Features

After applying the *Pearson correlation* and Random Forest importance, it was found that correctness was the most informative feature. Then, we tried to decompose the accuracy for each skill and see if we can use it for making predictions. The box plot below shows the accuracies based on one such skill.

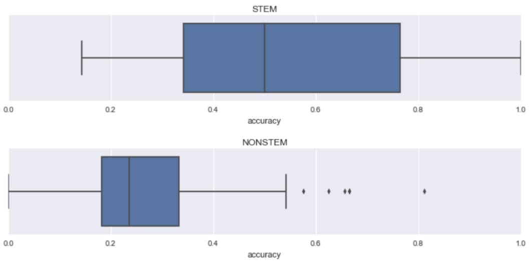


Figure 1: Box-plot for an example skill

This seemed to be a good prediction factor initially. However, the results obtained were significantly low, even for the simple, linear models. One important insight that we gained from this observation was that there were some students who scored well in a particular skill problem, but only with one or two attempts, while other students tried those problems for more than 5 times. After applying some filters, the accuracy in skills was not too informative.

4.4 General Student Behaviour

According to the previous section, accuracy on single skills has no significant impact on the results. Thus, we tried to classify the skills by difficulty.

[1] https://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/

We created new features with the help of existing features available in training set. Some of them are, Number of questions per session for each student, number of times a student used the system, total number of questions attempted by each student in the months of April and May (considered as summer months, i.e. before exam), and accuracy for the number of questions solved in these months.

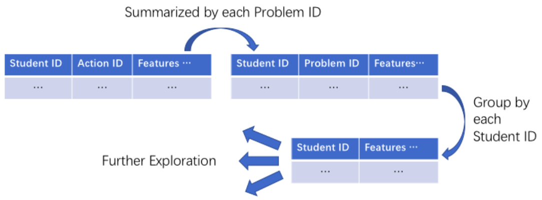
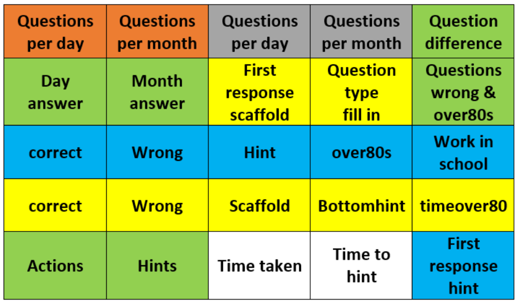


Figure 2: Training set creation process

According to **Fig.2**, there are two feature engineering processes. We transform the original action log to a table that summarizes actions by each question per student. We filter the important features which may affect the performance.

Step 1: For each question solved by every student, we used statistical methods to connect features, for instance, averaging, counting and ratio. The reason why we chose to take these two steps is that the original action log is too disperse for training and it is ineffective. **Fig.3** is the table of new features.



Questions per day	Questions per month	Questions per day	Questions per month	Question difference
Day answer	Month answer	First response scaffold	Question type fill in	Questions wrong & over80s
correct	Wrong	Hint	over80s	Work in school
correct	Wrong	Scaffold	Bottomhint	timeover80
Actions	Hints	Time taken	Time to hint	First response hint

Figure 3: 25 new features : orange (average), green (number), gray (standard deviation), blue (rate), yellow (true or false)

Step 2: We used the same method as step 1 to summarize 25 features regarding student information with different objects. Given that the performance of different types of questions may vary, we created new features based on the type of questions. We classified the type of skills into three classes based on the difficulty of the questions and created new features based on these three classes. The degree of difficulty is defined from three aspects, average correct rate, the correct rate of students who are accepted by 'STEM' and the correct rate of students who are not accepted by 'STEM'. Finally, we created 100 features $(25features \times (3classes + 1general))$. Including the 9 features described in [3], we have 109 features.

4.5 The relationship between school and answering frequency

Except for students' factors, we considered looking into some external factors might be helpful. The school influence is a good example. Schools may have different arrangements for taking tests. For instance, the average number of questions answered per day and the number of days that students were solving questions were additional features.

5 FEATURE SELECTION

There are some obvious problems which would result in a negative effect on the prediction. First, there are too many features in the data and even new features were created. Second, the training set has only 514 observations which include 164 students who are labelled as 'is STEM' and 350 students who are labelled as 'non-STEM'. To avoid the bias, the training data set must be balanced, and the balanced training set has 328 studentsfi data. Compared with the huge number of features, the student's data is too less to make a prediction. Third, based on the previous two aspects, the model has high chances of over-fitting. Thus, it is necessary to do feature selection before building the model.

Correlation Analysis. The correlation coefficients for each relationship type are respectively 0.0-0.3, 0.3-0.6, 0.6-0.9 and 0.9-1.0. Besides, the correlation coefficient can also be interpreted as positive or negative. Therefore, the relationships between all features should be interpreted. We calculated the Pearson correlation coefficient between all features and the target feature, 'is STEM', and filtered features that have correlation coefficient more than 0.3.

This method was not suitable for categorical data because it was not sufficient to identify the correlation between features and labels. For calculating the correlation between features, we considered correlation levels. As shown in **Fig.4**, we could use one feature to represent all strongly correlated features like, 'aveknow' and 'avecorrect', which specifies average knowledge level, and average correctness level.

Variance. Our approach was to calculate the variance of each feature and select the feature whose variance is larger than the threshold. For instance, if the variance of one feature is close to 0, then students cannot be differentiated using this feature. This feature may be considered inefficient as this method can only be applied to normalized data. The variance does not guarantee that the feature is important. It could simply be a large magnitude.

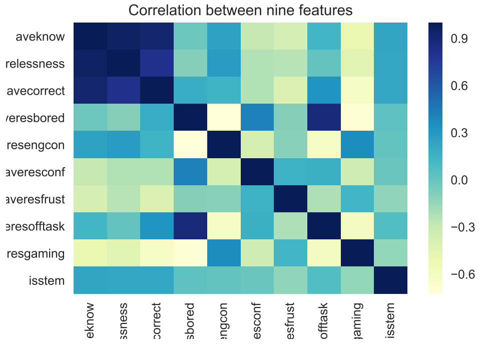


Figure 4: Correlation coefficient between 9 features

Maximal information coefficient (MIC). This method is different from correlation analysis as information entropy can only be applied to categorical data. The data contained both categorical and continuous data. The **Maximal Information Coefficient** (MIC) is the variants of information entropy and it is a better option for calculating correlation between categorical data and continuous data.

$$MIC(x, y) = max\{I(x, y)/log_2 min\{n_x, n_y\}\}$$

where $I(x, y)$ is the mutual information between data x and y. n_x, n_y are the number of bins into which x and y are partitioned, respectively.

We used 9 features, as shown in the figure. The features in the middle, such as 'averesbored', has a lower MIC. Meanwhile, other features have a higher MIC, which means they are more important. The phenomenon can be partly proved by Pearson Coefficient as well, though all correlations are weak. The difference of MIC between 9 features is not obvious, which means MIC might not be the best method.

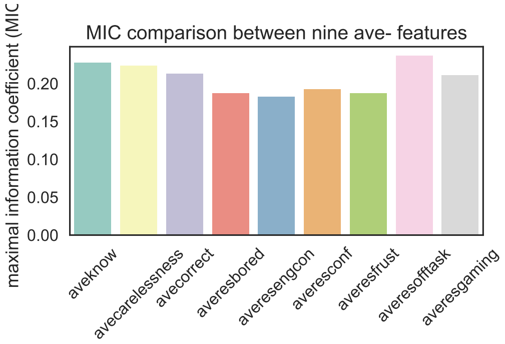


Figure 5: Maximal information coefficient (MIC) comparison between 9 features

Polynomial regression. Another method we used for creating features is higher order polynomial functions. Combination of existing features can often produce new features for classification. For example, we have features X, Y in the dataset and we create new features by taking squares of the X or Y and adding it to the dataset which can increase our classification accuracy.

Results. After removing the features which have high Pearson correlation, features with MIC more than 0.26 are:

The final selected 8 features are: average knowledge, average results of the task, number of actions, hint rate, hint rate and time to hint for middle difficulty skills, number of actions and first respond hint rate for easy difficulty skills.

6 MODELS USED

Once the features were ready, we had a range of classifiers we could use to train. Using cross-validation (80% train and 20% test), we checked the accuracy of classifiers.

6.1 Logistic Regression

Logistic Regression was our first choice of model, as is the case with any categorical data involved. We used the logistic regression libraries from `sklearn` to train our model and we applied *l1 regularization* in the classifier function which does feature selection.

6.2 SVM

Support Vector Machines (SVM) can be used for both classification and regression analysis. We used packages from `sklearn` for this classifier and calculated cross validation score on different feature sets.

6.3 Random Forest

Random forest is an ensemble learning method for classification and regression. This classifier works by averaging the results from shallow decision trees to get good accuracy (strong classifier). We changed quite a few parameters of the random forest classifier. We tuned parameters like 'max-depth' (maximum depth of the tree), 'n-estimators' (number of decision trees) and 'maximum-features' (number of features to consider when looking into the best split).

6.4 XGBoost

XGBoost (Extreme Gradient Boosting algorithm) is also a decision tree-based approach which ensembles simple decision trees and adds new trees depending on the error in the earlier ensemble.

7 RESULTS

The results of various classifiers are summarized in AUC terms in this section.

Training on the extended dataset and new features. First, we performed all classifications on the extended dataset in which 9 features were 'ave' features from the original dataset and other 100 features were created by us. We then had few experiments with reduced feature space. We applied PCA and performed classification on the first 45 principal components (Table 1). Later, we tried to use only 9 original features. We also tried to reduce the feature space by giving a threshold for MIC value of feature such that this value should be more than 0.2, 0.24 and 0.26 respectively. We also tried to explore a non-linear dependency between our features and target predictions. We extended our reduced datasets (with MIC > 0.26 for each feature and for 9 original features) by appending square of the current features.

The results are shown in **Table 1**. AUC score was the performance metric used. It was observed that normalization of features did not give any improvements in the results when compared to the non-normalized data.

Feature sets	RF	XGBoost	SVM	**LR**
109 features	0.537	0.553	0.5	0.63
Selected by PCA	0.531	0.569	0.5	0.6445
9 features	0.531	0.5593	0.456	0.6603
MIC > 0.2	0.5373	0.5562	0.6155	0.651
MIC > 0.24	0.553	0.584	0.628	0.6763
MIC > 0.26	0.63	0.597	0.61	**0.6816**
MIC polynomial	0.59	0.603	0.625	0.675
9 polynomial	0.516	0.55	0.547	0.667

Table 1: Model AUCs on different feature sets

The models provided some interesting observations. First, the ensemble methods (Random Forest and XGBoost) did not give a good AUC score for most of the datasets compared to other classifiers. Secondly, reducing feature space by making MIC threshold no less than 0.26 gave us the best AUC score with the Logistic Regression. This meant that some of our engineered features are redundant. Adding polynomial features did not give us a significant increase in performance (sometimes even decreasing). So we concluded that there is no square relationship between constructed features and final prediction.

Bag-Of-Words. With different Bag-Of-Words sizes, models like Random Forest(RF), XG-Boost, Support Vector Machines(SVM) and Logistic Regression(LR) were used for training. The AUCs obtained are summarized in the **Table 2** below:

BOW	RF	**XGBoost**	SVM	LR
BOW 1000	0.6502	0.6507	0.6354	0.6634
BOW 500	0.6611	**0.6894**	0.6022	0.6290
BOW 200	0.6263	0.6698	0.6352	0.5750
BOW 50	0.6033	0.6427	0.6065	0.5888

Table 2: Model AUCs on BOW

BOW 500 provides the best AUC when trained using the XG-Boost model. This means that BOW 500 has the most optimal vocabulary size when compared to other BOW sizes. The results show that AUCs obtained using BOW features are approximately equal to AUCs from other feature engineering techniques.

Thus it can be concluded that grouping similar actions were a different approach and a slightly new way of understanding the features.

Training on Association Rules. Similar classification models were applied on the dataset containing association rules (described in section 4.1). The AUC using this approach is low for each classifier (average AUC around 0.35 - 0.5). This shows that it is not possible to distinguish STEM and NON-STEM students by their accuracy in the most advanced skills for each.

Response frequencies between schools. As shown in **Table 3**, the third school has the highest rate of 'is STEM'. However, the students from this school did the least questions and spent the least number of days on solving them. Meanwhile, students in the first school did most questions but least of them are choosing STEM. This is further explained in the next section.

School ID	1	2	3	4
average questions	31.32	31.04	**27.18**	28.57
number of day	7.44	9.24	**5.83**	7.70
number of ISSTEM	8	86	44	26
number of people	52	242	99	116
rate of ISSTEM	15.4%	35.5%	**44.4%**	22.4%

Table 3: Response frequencies between schools

8 DISCUSSION

Polynomial problem. The end behaviour of polynomial function[1] may lead to the difference in results when using polynomial expansion and the result when not using it. The distribution of the 9 'ave' features might have non-linear correlation that makes the performance of polynomial expansion better. However, the distribution of all new features has linear correlation that makes the polynomial expansion worse. In **Fig.6**, linear and polynomial both work in the training set. However, validation data locates at the tail, that AUC would decrease, for instance, data 'A'.

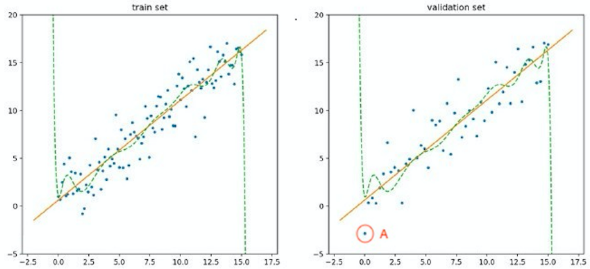


Figure 6: The end behavior of polynomials

Bias of the datasets. There are fewer questions answered in the third school, compared to the other three schools. However, the proportion of the third school students choosing STEM is the highest. There may be two possibilities:

- Bias in the student sampling: Most of the students sampled are excellent in the third school. The performances of other schools' students are evenly distributed.
- Bias in the school sampling: The third school may be more concerned with the education for mathematics. Besides, the third school might be an elite school.

The efficiency of new features. The performance of all new features is better than the 9 'ave' features. The 'ave' features are extracted by summarizing the student effects, disengaged behaviour and performance. Our idea is that this summary method may limit other possibilities for student behaviour. The new features are not sufficient to discover more aspects of student behavioural characteristics. External factors such as the difficulty of the questions are also considered.

Alternative Features. Other than the features discussed above, the Bag-of-Words and Association Analysis were used as alternative approaches for feature engineering. From the results obtained above(sections 7.2) it can be derived that if there are a significant number of similar actions, BOW can provide good AUC scores. On the other hand, the association rules provided a lower AUC score for the same data(section 7.3). This suggests that the associativity rules, when applied to skills, can be a good approach, but not as good as other feature engineering techniques.

9 CONCLUSION

While there can be several factors that determine a student's choice of the STEM fields, making definite conclusions about these factors is a difficult task. For example, if we look at some of the techniques applied above, such as the Bag-Of-Words, the results might show an AUC similar to other approaches, but it was a new, more out-of-the-box approach for feature engineering. This being an unconventional method, proved to be quite insightful for grouping similar actions.

Additionally, models like SVM, Logistic Regression, XG-Boost and Random Forest provided an average AUC of about 0.55-0.65. This may not be a very good score, but different combinations of features and models provide different results, which can be used for drawing conclusions in various scenarios related to the student or the test methods.

10 CHALLENGES AND LIMITATIONS

The challenges faced in this task are mentioned below:

- One of the main challenges we faced was the data being unbalanced. Out of the 500 observations spread across the action logs, only 164 corresponded to the "is STEM" label. The data, as we see, was biased towards the NON-STEM choices. We down-sampled the data for our training.
- In terms of data, we did not have a direct mapping between students and actions. So, to construct the training data, we had to map all the students to their respective actions manually.
- Initially, there were about 72 original features available as part of the action logs. While this was a considerable number, most of these features seemed to be unrelated and weak. We could only use 9 of these features and we had to perform feature engineering to come up with 100 additional features!

11 FUTURE ENHANCEMENTS

To solve this classification problem, an interesting approach would be to use Neural Network-based training. Since the original data set is a sequence of actions there is a good potential for Recurrent Neural Networks[2] to handle this type of data and to achieve a good AUC score. While the AUC score improvement cannot be guaranteed, better results can be achieved in terms of speed, or even to gain some interesting insights.

REFERENCES

[1] Maurice George Kendall and others. 1946. The advanced theory of statistics. *The advanced theory of statistics* 2nd Ed (1946).

[2] John Mingers. 1989. An empirical comparison of selection measures for decision-tree induction. *Machine learning* 3, 4 (1989), 319–342.

[3] Maria Ofelia Pedro, Ryan Baker, Alex Bowers, and Neil Heffernan. 2013. Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. In *Educational Data Mining 2013*.

[2] https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5