Analysis of Conditions for Reliable Predictions by Moodle Machine Learning Models

https://doi.org/10.3991/ijet.v16i06.18347

László Bognár
University of Dunaújváros, Dunaújváros, Hungary

Tibor Fauszt (✉)
Budapest Business School University of Applied Sciences, Budapest, Hungary
fauszt.tibor@uni-bge.hu

Gábor Zsolt Nagy
Eduroll Consulting, Budapest, Hungary

Abstract—In this paper the issue of bias-variance trade-off in building and operating Moodle Machine Learning (ML) models are discussed to avoid traps of getting unreliable predictions. Moodle is one of the world's most popular open-source Learning Management System (LMS) with millions of users. Although since Moodle 3.4 release it is possible to create ML models within the LMS system very few studies have been published so far about the conditions of its proper application. Using these models as black boxes hold serious risks to get unreliable predictions and false alarms. From a comprehensive study of differently built machine learning models elaborated at the University of Dunaújváros in Hungary, one specific issue is addressed here, namely the influence of the size and the row-column ratio of the predictor matrix on the goodness of the predictions. In the so-called Time Splitting Method in Moodle Learning Analytics the effect of varying numbers of time splits and of predictors has also been studied to see their influence on the bias and the variance of the models. An Applied Statistics course is used to demonstrate the consequences of the different model set up.

Keywords—Machine learning; online learning; student success; Moodle

1 Introduction

According to the classical definition Learning Analytics (LA) is "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs" [22]. Within this rather wide scope of LA one specific objective at a university can be to optimize the teaching-learning process of a university course to predict students' performance or reduce the risk of students not achieving the minimum grade to pass the course. As the online education is getting to be widespread student retention becomes a big challenge that universities must face [5], [9], [10]. Therefore, the Learning Management Systems (LMS) have been continuously revised and different tools are developed to identify students at risk of dropping out [2], [15].

Recently Machine Learning (ML) technics have been widely used to predict students' success (or failure) at fulfilling the course requirements and to help those students who seem to be at risk of dropping out. There are multiple machine learning models used in education [3], [8], [17], [20]. Đambić et al. made a model with 5 independent variables with that purpose of identifying students who needs additional attention. The variables are based on the students' performance at the first period of the semester. The F1 Score of their model was 77% which value is in an acceptable range although their model was highly biased [4]. Lykourentzou et al. proposed a dropout prediction method for e-learning courses and used three different machine learning methods with time-invariant (e.g., age, gender, etc.) and time-varying (e.g., grades, attendance) predictors in their study. The model was split into 7-time segments. The time-invariant student data were found to be less accurate predictors of a student's decision to drop out compared to time-varying data and the models had significantly better values of accuracy as the course progressed [12]. Erkan proposed a method for accurate prediction of at-risk students in an online course [6]. In his study three machine learning algorithms were used separately for the classification of students according their levels of risk. The model contained only time-varying predictors in 3-time segments. He found that the number of time-varying predictors has an impact on overall model accuracy. As the courses progressed the models had better accuracy. Moseley and Mead analyzed 3978 records on 528 nursing university students split into training set and validation set [16]. Machine learning algorithm with rule induction method was developed to predict dropouts in nursing courses. The model was able to predict the dropping out with 84% accuracy and 70% of the identified students dropped out in fact. Kotsiantis et al. compared six different ML algorithms in two experiments and found the Naïve Bayes algorithm the most appropriate to be used for the prediction of students' dropout [11]. Their study was among the first to develop a machine learning model to identify students who were likely to drop out. The attributes used in the model were students' registry data and attributes from tutors' recorded data. Tan end Shao developed and compared three prediction models based on different ML algorithms [21]. They found all algorithm to be appropriate for student dropout prediction but they found Decision Tree algorithm presented a better performance. Ram et al. considered 21 variables (sense of belonging, scholastic context, financial status, prior academic details, and family background) which were related to different aspects of student's demographic data in their model [19]. They found that first term GPA is the most important predictor for first-year dropout rates. Furthermore, metrics were defined to infer students' social integration from smart card transactions. These new features were effective in significantly improving precision and recall rates in identifying drop-out students. Chai and Gibson evaluated different models for predicting which first year students are most at-risk of leaving at their first semester of study in the period of pre-enrolment, enrolment, in-semester, and end-of-semester [1]. A dataset of 23,291 students were analyzed and the used machine learning technics were logistic regression, decision trees and random forests. The model achieved the best performance with logistic regression with 67% precision and 29%

recall. Mogus et al. analyzed Moodle LMS logs and students' feedbacks for that purpose to measure students' effectiveness in learning [14]. In their analysis Statistica 8 and Weka software were used for data preprocessing, classification, regression, clustering, association rules and visualization. Evangelista also used Weka software in the model based on Moodle log data [7]. The predictors related to student's study behavior were course viewing time, resource views, quiz taken, replies in discussions, and views at weekends. It was found that predictor attributes such as activities completed, course views and assignment passed are the ones which are strongly correlated to students' performance. Młynarska et al. found that the timing of activities is an important factor in terms of grade [13]. Nurbiha et al. found that attracting the students'attention on the very first page that they visited is a very important factor to encourage them to stay in Massive Open Online Courses [18].

In this paper the issue of bias-variance trade-off in building and operating Moodle Machine Learning (ML) models are discussed to avoid traps of getting unreliable predictions. Moodle is one of the world's most popular open-source learning platforms with millions of users and widely used in online education. Starting with Moodle 3.4 release ML Analytics are integrated part of the system. Using the tool, scheduled analysis can be executed based on ML models. During the course, those students can be identified automatically who are lagging and are at risk of dropping out. The creators of Moodle provided a great tool for course builders with this new feature. However, very few studies have been published so far about the conditions of its proper application [15].

To develop a Learning Management System (LMS) with integrated ML models it requires cooperation of specialists in different fields like course architects, course teachers, statisticians, IT specialists, LMS administrators, etc. Later in the operational phase when the system is used on a daily basis to set up different ML models for different courses usually only some course teachers and LMS administrators are involved. Frequently the ML models are used as ready-made tools by some course teachers.

Even in case of a well-developed LMS system the continual supervision of the applicability of the machine learning models should be inevitable. Using these models as black boxes hold serious risks to get unreliable predictions and false alarms (false no alarms). Unreliable predictions may arise either from the inappropriately determined models or from the differences of the circumstances in the model building and the prediction phases. Here the conditions for reliable predictions by Moodle Machine Learning models are analyzed.

2 Machine Learning Models in Education

Predictive modelling allows us to model the relation between an educational target (also referred as to dependent variable) and a set of predictors (also referred to as independent variables or features) related to the learners and their learning activities in a given learning contexts. For example, in a binary model for a given student the value of the target can be 0 if the course is passed and 1 if the course is failed. The values of the predictors for a given student are either determined by some known features of the student (e.g.: gender, age, grades in former courses, etc.) or extracted and computed from activity logs stored in an LMS throughout a course. In case of online university courses when students' learning activities are recorded in great details the predictors from activity logs are widely used.

Predictive modelling can be implemented using Supervised Learning (SL) algorithms which search a modelling function that maps the predictors-target relation. In the model building phase, the algorithms use a previous realization of the course (run in one of the former semesters) to determine this function and then in the operational phase, in a later course, it can be used for prediction.

Mathematically it can be written as a hypothesis in the general form

$$Y = f\left(X_1, \dots, X_j, \dots, X_p\right) + \varepsilon \tag{1}$$

where *Y* is the target, *f* is some unknown function of the predictors $X_1, \dots, X_p$ (usually fixed function with unknown parameters) and *ε* is a random error term, which is independent of the predictors and has mean zero.

The SL algorithms try to get the "best" $\hat{f}$ estimate for the unknown *f* function using the existing data of the predictors and the target. The "best" $\hat{f}$ is defined as the function which minimizes the so-called *cost function Q* which is usually the squared expected difference between the actual (*Y*) and the predicted ($\hat{Y}$) values of the target.

$$Q = E\left[(Y - \hat{Y})^2\right] = \text{minimum.} \tag{2}$$

In practice we use a certain subset of the existing data called *Training set* and get a certain $\hat{f}$. If we use a different *Training set*, we are very likely to get a different $\hat{f}$. As we keep changing *Training sets*, we get different outputs for $\hat{f}$. The amount by which it varies as we change the *Training sets* is called *Variance*.

To estimate the true *f* with different methods like linear or logistic regression we frequently use some simple function between the predictors and the target. For most real-life scenarios, however, the true relationship is more complicated. Simplifying assumptions give *Bias* to a model. The more erroneous the assumptions with respect to the true relationship the higher the *Bias*, and vice-versa.

Generally, a model will have some error when tested on some test data (called *Validation set* in what follows). It can be shown mathematically that both *Bias* and *Variance* can only add to a model's error. We want a low error, so we need to keep both *Bias* and *Variance* at their minimum. However, that is not quite possible. There is a trade-off between *Bias* and *Variance*.

Table 1. The X matrix of predictors and the Y target vector in a Time Splitting model

Sample	Student	Predictors										Time Split	TargetY
		Time Split Indicators					*Core Predictors*						
		S_1	…	S_k	…	S_c	X_1	…	X_j	…	X_p		
		X											
1	1	1		0		0						1	0
…	…	…		…		…							…
…	n	1		0		0							1
…	1	0		0		0						…	0
…	…	…		…		…							…
…	n	0		0		0							1
(k-1)n+1	1	0		1		0						k	0
…	…	…		…		…							…
Sample i	→	0		1		0							1
…	1	0		0		0						…	0
…	…	…		…		…							…
…	n	0		0		0							1
…	1	0		0		1						c	0
…	…	…		…		…							…
m = cn	n	0		0		1							1

If one specific course from its starting date to its ending date is used to build the model (this time interval is used as a whole to collect the data) the predictor X_j takes one specific value for each student. For the whole set of enrolled students, it can be represented by a vector $\boldsymbol{X_j}$ with dimension of *n* where *n* is the number of students and the *i-th* element of this vector refers to the *i-th* student.

There are situations where the predictor X_j takes different values in different learning contexts for the same student. Here the different learning contexts may refer for example to different courses or to different time segments of the same course. In these situations, the vector $\boldsymbol{X_j}$ has dimension of *m* which is not equal to the number of students' *n* anymore since for one student more than one values can be assigned to the X_j predictor relating to the different learning contexts. Now the *i-th* element of this vector refers to a certain student in a certain learning context. In this case usually additional predictors are given to the model to indicate the context itself. If *c* different contexts are given for each student, then $m = nc$.

A widely used version of the aforementioned situation is the so-called *Time Splitting method* when in case of one specific course one semester is divided into *c* time subintervals *(Time Splits)* and the values of the predictors are recorded in each *Split*. Here one specific *Split* defines the specific learning context. In this method *c* additional predictors $S_1,\ldots,S_c$, the so-called *Time Split Indicators*, are given to the model indicating the actual *Time Split* when the data are recorded. The original $X_1,\ldots,X_p$ predictors can be called as *Core Predictors*. In this model the values of the target are repeated in each *Split*. Hence the set of the $\boldsymbol{X_j}$ predictor vectors form the $\boldsymbol{X}$ matrix of predictors with dimension of $(m\ x\ r)$, *where* $r = c+p$. The $\boldsymbol{Y}$ target vector has the dimension of *m*. One row of this $\boldsymbol{X}$ matrix together with the corresponding value of $\boldsymbol{Y}$ is called a *Sample*. In Table 1 the layout of $\boldsymbol{X}$ and $\boldsymbol{Y}$ is depicted.

3 Circumstances Influencing the Goodness of the Model and the Reliability of the Predictions

Many factors and circumstances have influences on the “goodness” of the model and the reliability of the predictions. In general, the most influential circumstances are *a. type of course investigated (online, on-site, full-time, part-time, subject of the course, course activities, course length, etc.); b. The minimum grade to pass the course c. type of predictors (related to known features or to social or cognitive activities, etc.); d. type of Time Splitting Scheme; e. type of SL algorithms used; f. metrics used to measure the “goodness” of the model.*

All of these may have consequences for the details, for the properties of the mathematical model, namely for the:

- number of the predictors r, the Samples m and hence the size m x r and the m/r row-column ratio of the predictor matrix ***X*** and the size m of the target vector ***Y***,
- sparsity and skewness of the predictor matrix ***X***,
- sparsity and skewness of the target vector ***Y***,

  and therefore, for the:
- Bias and Variance of the model.

4 The Courses and the Predictors of the Present Study

The course of the *Applied Statistics* at the University of Dunaújváros in Hungary has been used to build the ML models in the spring semester of the academic year 2019/20. More precisely the data for both versions of the course one for the full-time students and the other for the correspondence students were used. 57 full-time students and 94 correspondence students were involved in the study.

The course was delivered through the Moodle LMS and developed in a way to be able to serve even fully online students. It means that students’ learning activities were pursued mainly within the LMS framework and their activities could be recorded in detail. The subject material could be attained by using different type of learning resources and activities: *Lecture videos, Minitab videos (videos for problem solving with a statistical software), PDF lecture notes, Books of solved exercises, Quizzes for Self-testing.*

Due to the scientific nature of the course the predictors refer mainly to cognitive activities and their values were computed from the recorded course logs like number of quiz attempts, max grades at quizzes, number of clicks to view videos or to learn from lecture notes or from books of exercises.

The learning material was partitioned into 7 chapters and altogether 57 core predictors have been defined. In the different ML models, some of them or all these predictors have been selected.

In this paper only the course for full-time students is discussed. Throughout the course the students wrote four midterm-tests, 25 points of each, and the sum of the points they earned had to reach the minimum points of 70 to get a grade to pass the course. This is an important difference from the correspondence course where the students wrote only one final test to get their grades. An obvious consequence of the continuous testing of full-time students is that the students are encouraged to learn uniformly throughout the course hence the matrix ***X*** is not sparse with many empty rows in the early *Time Splits*. In the case of the correspondence course, the situation is basically different where in the early *Time Splits* rare students' activity can be recorded and the predictors have zero values in many samples and in this sense the matrix ***X*** is skewed and sparse.

Actually 9 students failed out of the 57 hence the ***Y*** vector was rather skewed. The main purpose of the prediction was to correctly classify those students who are at risk not fulfilling the minimal course requirement.

One specialty of the Moodle Learning Analytics used in our study must be mentioned since it may have influence on the results. It automatically appends additional predictors to the *Core Predictors*, namely the averages of the individual predictor values in each *Split*, hence it doubles the number of the original *Core Predictors*. To differentiate these averaged value predictors from the original ones in what follows two groups of the *Core Predictors* will be distinguished: The *Learning Predictors* (LP, directly referring to some features of the students or to their learning activities) and the *Average Predictors*.

5 Performance Metrics

In the comprehensive study mentioned above two different SL algorithms, Logistic Regression (LR) and a two-layer-feed-forward Neural Network (NN) were used to train and evaluate the models (MATLAB 2008, release 2018b). Here only the results of the LR analyses are discussed where 5-fold cross-validation technics were applied.

To quantify the models' performance different widely accepted metrics computed from the confusion matrix were determined. In the present context the elements of the confusion matrix:

- *Number of true positives (TP): when for a sample the prediction is 1 (the student fails to pass the course) when the actual value of the target is 1 (failed),*
- *Number of false positives (FP): when for a sample the prediction is 1 (the student fails to pass the course) when the actual value of the target is 0 (passed),*
- *Number of true negatives (TN): when for a sample the prediction is 0 (the student passes the course) when the actual value of the target is 0 (passed),*
- *Number of false negatives (FN): when for a sample the prediction is 0 (the student passes the course) when the actual value of the target is 1 (failed).*

For all the models the following metrics have been evaluated: *F1 score (F1), normalized Matthiew Correlation Coefficient (nMCC), Precision, Recall, Error rate, Accuracy (Acc)*.

F1 [0-1] and *nMCC* [0-1] are widely used metrics to get general overview for the goodness of a model. *F1* is better to use if the frequency of the classes in the target *Y* is highly imbalanced [the number of failed students (*Y* = *1*) is much less than those who passed the course (*Y = 0*)] and the focus is on the reliable prediction for those being at risk to fail. *F1* is defined as *F1 = 2*Precision*Recall/(Precision + Recall)*.

In this case the *Precision* and the *Recall* are the metrics showing more easily interpretable model performance for predicting failure for students at risk. According to the definition *Precision = TP/(TP + FP)* showing *the fraction of correctly classified failed students of all students that the model predicted as failed*, and *Recall = TP/(TP + FN) showing the fraction of correctly classified failed students of all students that actually failed*.

*nMCC is defined as nMCC = (MCC + 1)/2 where MCC = (TP*TN – FP*FN)/Sqrt[(TP +FP)*(TP + FN)/(TN + FP)*(TN + FN)]. nMCC* may be considered better describing the overall goodness of the model when the misclassification of both the *failed* and the *passed* cases are considered equally important. The values of *F1* and *nMCC* are very close when the data in ***Y*** are balanced.

These *F1* and *nMCC* metrics have been evaluated in the *Training (Tr)* and the *Validation (V)* phases too. The *F1* and the *nMCC* values in the *Validation* phase *[F1(V) or nMCC(V)]* are directly related to the *Biases* of the models (*not the same but strongly related to it*). The differences of these metrics' values between the *Training* and the *Validation phases [F1(Tr) – F1(V) or nMCC(Tr) – nMCC(V)]* are related to the *Variances* of the models (*not the same but strongly related to it*).

Another way of studying the *Bias* and the *Variance* of a model is to sketch its *Learning Curve*. This curve shows the changing tendency of the *Error rate = [(FP + FN)/m]* when continuously increasing number of *Samples* are used to train the model and an independent *Validation* set is used for validation after each training.

Plotting the *Error* rate against the *Sample* size for both the *Training* and the *Validation sample* set the *Learning Curves* can be depicted. From these *Learning Curves* not only the actual model *Bias* and *Variance* can be seen but the way of possible model improvement can be deducted (more samples or more/less indicators would be better to use).

Accuracy is defined as *Accuracy = (TP + TN)/m.*

6 Findings

In Table 2 and in Table 3 the main characteristics of the investigated models and the metrics related to the models' performance are summarized. The same full-time Applied Statistics course was used to build all models. In Table 2 the models contain both the *Learning Predictors* and the *Average Predictors*, but in the models in Table 3 only the *Learning Predictors* form the *Core Predictors*.

Table 2. a. Models with Learning and Average Predictors

Model	# of Split	# of Samples m	# of Core Predictors p	Ratio m/p
Model 1	10	570	114	5.0
Model 2	10	570	56	10.2
Model 3	10	570	28	20.4
Model 4	4	228	114	2.0
Model 5	4	228	56	4.1
Model 6	4	228	28	8.1
Model 7	1	57	114	0.5
Model 8	1	57	56	1.0
Model 9	1	57	28	2.0

b. Models with Learning and Average Predictors and their performance metrics

Model	Training			ValidationBias			Variance	
	Acc	*F1*	*nMCC*	*Acc*	*F1*	*nMCC*	*F1(Tr)- F1(V)*	*nMCC(Tr)- nMCC(V)*
Model 1	0.94	0.81	0.88	0.86	0.61	0.76	0.20	0.12
Model 2	0.95	0.82	0.90	0.87	0.60	0.76	0.22	0.14
Model 3	0.88	0.52	0.73	0.84	0.39	0.66	0.13	0.07
Model 4	0.97	0.88	0.93	0.81	0.52	0.71	0.36	0.22
Model 5	0.96	0.85	0.92	0.82	0.44	0.67	0.41	0.25
Model 6	0.89	0.56	0.76	0.84	0.42	0.67	0.14	0.09
Model 7	1.00	1.00	1.00	0.47	0.29	0.54	0.71	0.46
Model 8	0.98	0.95	0.97	0.82	0.44	0.67	0.51	0.30
Model 9	0.95	0.80	0.90	0.75	0.22	0.53	0.58	0.37

Table 3. a. Models only with Learning predictors

Model	# of Split	# of Samples m	# of Core Predictors p	Ratio m/p
Model 10	10	570	57	10.0
Model 11	10	570	28	20.4
Model 12	10	570	14	40.7
Model 13	4	228	57	4.0
Model 14	4	228	28	8.1
Model 15	4	228	14	16.3
Model 16	1	57	57	1.0
Model 17	1	57	28	2.0
Model 18	1	57	14	4.1

b. Models only with Learning Predictors and their performance metrics

Model	Training			ValidationBias			Variance	
	Acc	*F1*	*nMCC*	*Acc*	*F1*	*nMCC*	*F1(Tr)-F1(V)*	*nMCC(Tr)-nMCC(V)*
Model 10	0.94	0.81	0.88	0.91	0.71	0.83	0.10	0.05
Model 11	0.95	0.82	0.90	0.89	0.60	0.77	0.22	0.13
Model 12	0.88	0.52	0.73	0.85	0.40	0.67	0.12	0.06
Model 13	0.97	0.88	0.93	0.77	0.47	0.68	0.41	0.25
Model 14	0.96	0.85	0.91	0.83	0.53	0.72	0.32	0.19
Model 15	0.89	0.56	0.76	0.84	0.37	0.64	0.19	0.12
Model 16	1.00	1.00	1.00	0.60	0.28	0.52	0.72	0.48
Model 17	0.98	0.95	0.97	0.74	0.21	0.52	0.74	0.45
Model 18	0.95	0.80	0.89	0.77	0.32	0.59	0.48	0.30

Studying the values in these tables and constructing figures to see the tendencies better some general findings can be summarized:

At any model with a given number of *Core Predictors* the models are getting better according to their *Variances* as the number of *Splits* and hence the *Ratios* are increasing (Figure 1). *Model 10* has the smallest *Variance* where it has only *Learning Predictors* and the number of *Ratio* is 10 *(*570 *Samples*, 10 *Splits*).

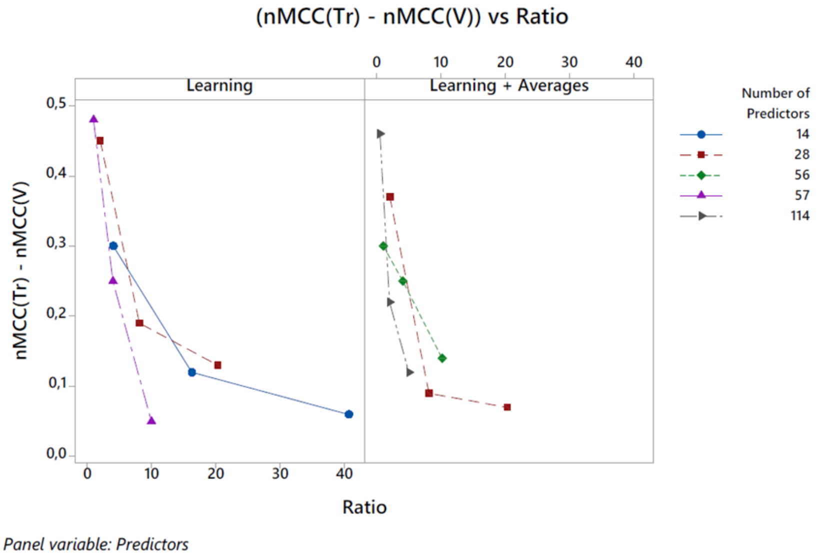

Fig. 1. *Variance*. The *nMCC(Tr) – nMCC(V)* metrics against the *Ratio* values

Comparing the magnitude of the *Variances* of the models with different number of *Core Predictors* the picture is not so clear. The number of *Core Predictors* and the number of *Split*s together determine the goodness of the model with respect to their

Variances. This can be seen better in Figure 2 where the *nMCC(Tr) – nMCC(V)* values are depicted against the number of *Splits*. When the number of *Splits* is 1 the large number of *Core Predictors* causes overfitting and the model *Variances* are too high. As the number of *Splits* is on the increase usually the models with larger number of *Core Predictors* outperform the other models reaching lower *Variances*.

In cases when the models have only *Learning Predictors,* at any model with a given number of predictors the models' goodness according to their *nMCC (V)* values are tendentiously increasing (the *Biases* are decreasing) as the number of *Splits* and hence the *Ratios* are increasing (Figure 3). The best *nMCC (V)* value is reached at *Model 10* with 57 *Learning Predictors* and 10 *Splits*. Usually *nMCC(V)* is larger if the number of *Learning Predictors* is larger, however it is important to point out that the phenomenon of overfitting appears here also at small *Split (Ratio)* values. In the models with *Learning* and *Average Predictors* the tendencies are similar however the magnitude of the differences between the models with different number of predictors are not so convincing.

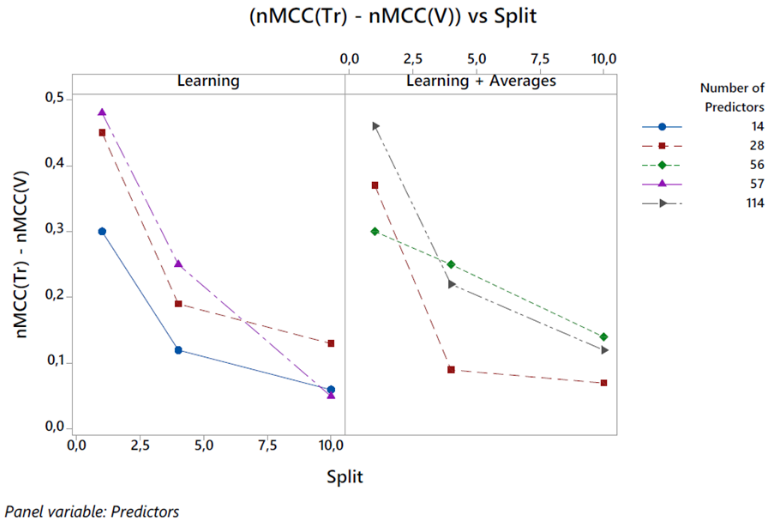

Fig. 2. *Variance.* The *nMCC(Tr) – nMCC(V)* metrics against the number of *Splits*

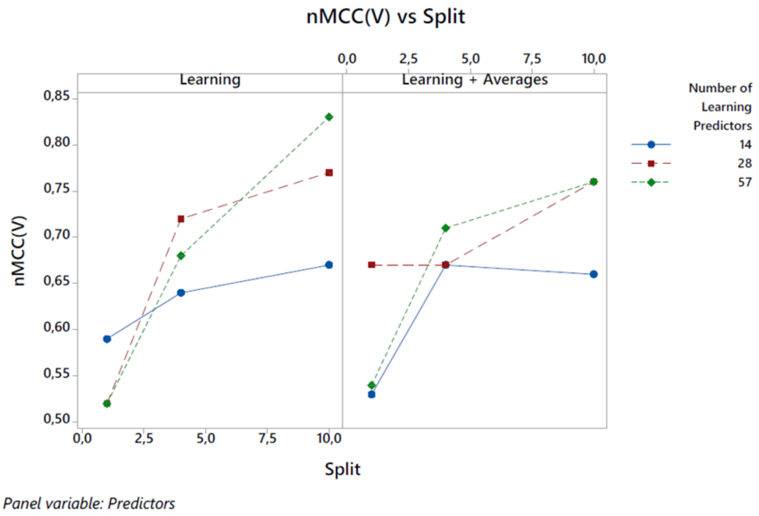

Fig. 3. *Bias.* The *nMCC(V)* metrics against the number of *Splits*

According to the findings above among the 19 models investigated the best model with minimum *Variance* and *Bias* is *Model 10* with 57 *Core Predictors* (*Learning Predictors* only) where the number of *Splits* is 10 hence the number of *Samples* is 570 and the *Ratio* is 10.

In Figure 4 its *Learning Curve* is shown and in Figure 5 the *Learning Curve* of the *Model 14* can be seen as an example for comparison.

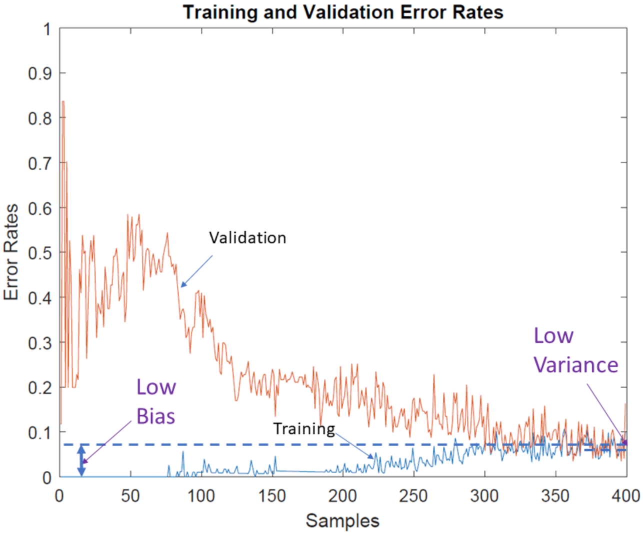

Fig. 4. Bias and Variance. The Learning Curve for the Model 10

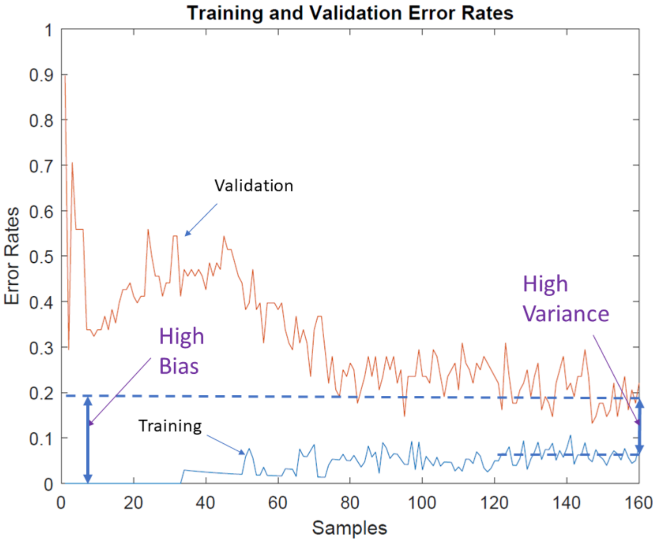

Fig. 5. Bias and Variance. The Learning Curve for the Model 14

The closeness of the *Training* and the *Validation* curves in Figure 4 indicates low *Variance* and the low level of the *Validation Error rate* suggests low *Bias*. The curves also suggest that collecting more *Samples* would not improve this model since both the *Training* and the *Validation Error Rate* curves reached their "horizontal imaginary asymptote". To further reduce *Bias* more predictors should be added but in this case the number of *Samples* also must be increased to avoid overfitting.

The goodness of the model can be characterized with the different performance metrics. The *Validation Accuracy* = 0.91 is high however it can be misleading. If the focus is on the reliable prediction for students at risk the *F1(V)* = 0.71 is better to use. More interpretable metrics are the *Precision(V)* = 0.69 and mainly the *Recall(V)* = 0.73. This *Recall(V)* value shows that of all students that actually failed 73% is classified correctly.

The role of the *Average Predictors* in the models is rather ambiguous. The effect of their presence is illustrated in Figure 6 and in Figure 7. In the models with 57 *Learning Predictors* and 570 *Samples* both the *Variance* and the *Bias* are worse when the *Average Predictors* are added to the *Learning Predictors*. The models become overfitted. In cases with fewer number of *Learning Predictors* the presence of the *Average Predictors* may have advantageous effects on model *Bias* or *Variance*.

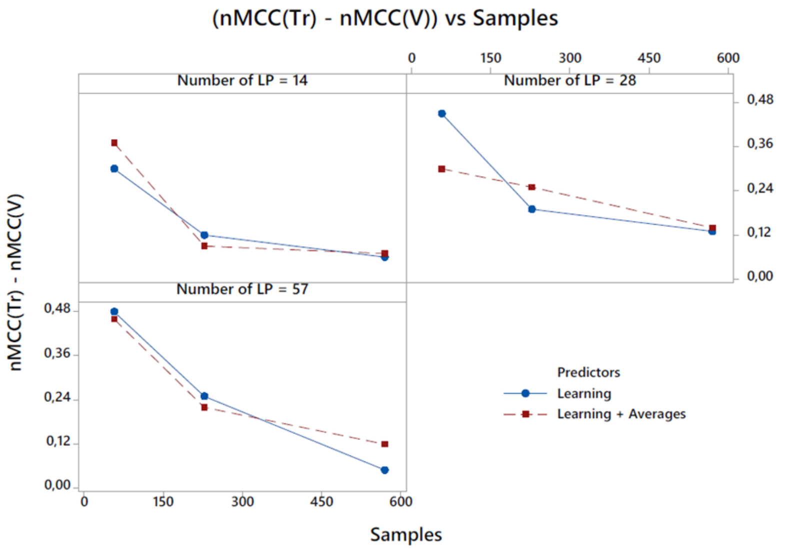

Fig. 6. The influence of the Average Predictors on the (nMCC (Tr) – nMCC(V)) metrics

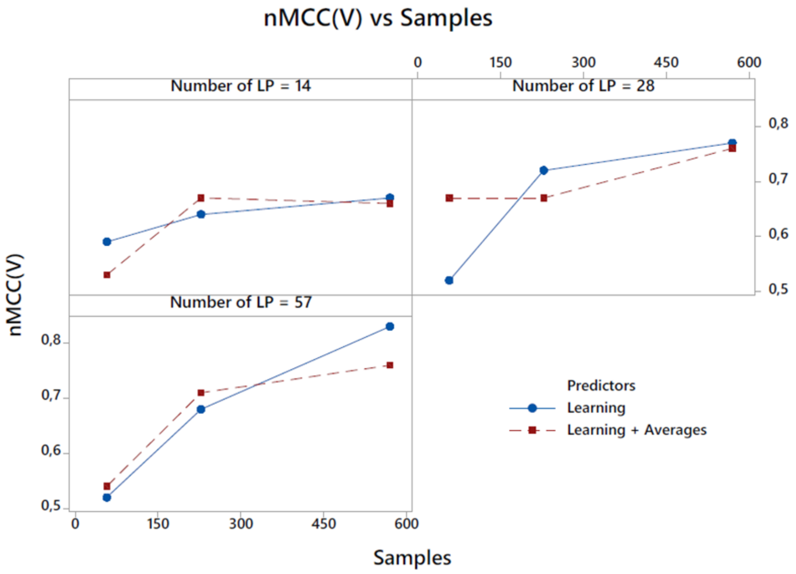

Fig. 7. The influence of the Average Predictors on the nMCC(V) metrics

7 Conclusion

In this paper some important aspects of building and operating educational machine learning models have been discussed. The influence of the size and the row-column ratio of the predictor matrix on the goodness of predictions has been shown. In the so-called *Time Splitting method* of Moodle Learning Analytics, the effect of varying numbers of time splits and predictors has been studied to see their influence on the bias and the variance of the models. As an example, the results of the analysis of an Applied Statistics course for full-time students were shown.

8 References

[1] K. E. Chai, D. Gibson, "Predicting the Risk of Attrition for Undergraduate Students with Time Based Modelling," In proc. *12th International Conference on Cognition and Exploratory Learning in Digital Age*. (CELDA) 2015, pp. 109-116.

[2] H. Coates, R. James, G. Baldwin, "A Critical Examination of the Effects of Learning Management Systems on University Teaching and Learning", *Tertiary Education and Management*. 2005, vol. 11, pp. 19–36. https://doi.org/10.1080/13583883.2005.9967137

[3] D. R. Cox, "The Regression Analysis of Binary Sequences," *Journal of the Royal Statistical Society*. Series B (Methodological). 1958, vol. 20, issue 2, pp. 215–242. https://doi.org/101111/j.2517-6161.1958.tb00292.x

[4] G. Đambić, M. Krajcar, D. Bele, "Machine learning model for early detection of higher education students that need additional attention in introductory programming courses," In proc. *International Journal of Digital Technology & Economy*. 2016, vol. 1, no. 1, pp. 1-11.

[5] D. Delen, "A comparative analysis of machine learning techniques for student retention management," *Decision Support Systems*. 2010, vol. 49, issue 4, pp. 498-506. https://doi.org/10.1016/j.dss.2010.06.003

[6] E. Erkan, "Identifying At-Risk Students Using Machine Learning Techniques: A Case Study with IS 100," *International Journal of Machine Learning and Computing*. 2012, vol. 2, no. 4, pp. 476 – 480. https://doi.org/10.7763/ijmlc.2012.v2.171

[7] E. Evangelista, "Development of Machine Learning Models using Study Behavior Predictors of Students' Academic Performance Through Moodle". *International Journal of Innovative Technology and Exploring Engineering*. (IJITEE), 2019, vol.8, issue-6S3.

[8] M. H. Hassoun, *Fundamentals of Artifical Neural Networks. MIT Press*, 1995, Cambridge, MA, USA.

[9] J. W. Jia, M. Mareboyana, "Predictive models for undergraduate student retention using machine learning algorithms," *Transactions on Engineering Technologies*. 2014, Springer, Dordrecht. pp. 315-329. https://doi.org/10.1007/978-94-017-9115-1_24

[10] J. S. Kamaghe, E. T. Luhanga, M. Kisangiri, "The Challenges of Adopting M-Learning Assistive Technologies for Visually Impaired Learners in Higher Learning Institution in Tanzania," *International Journal of Emerging Technologies in Learning*. 2020, vol. 15. no 1. https://doi.org/10.3991/ijet.v15i01.11453

[11] S. Kotsiantis, C. Pierrakeas, P. Pintelas, "Preventing student dropout in distance learning systems using machine learning techniques," In proc. AI Techniques in Web-Based Educational Systems at Seventh International Conference on Knowledge-Based Intelligent Information & Engineering Systems, 2003 September, Oxford, UK, pp. 3-5. https://doi.org/10.1007/978-3-540-45226-3_37

[12] I. Lykourentzou, I. Giannoukos, V. Nikolopoulos, G. Mpardis, V. Loumos, "Dropout prediction in e-learning courses through the combination of machine learning techniques," *Computers & Education*, 2009, vol. 53, no. 3, pp. 950-965. https://doi.org/10.1016/j.compedu.2009.05.010

[13] E. Młynarska, D. Greene, P. Cunningham, “Indicators of Good Student Performance in Moodle Activity Data,” 2016, In arXiv preprint arXiv:1601. 02975

[14] A. M. Mogus, I. Djurdjevic, N. Suvak, “The impact of student activity in a virtual learning environment on their final mark,” *Active Learning in Higher Education*. 2012, vol. 13, no. 3, pp. 177–189. https://doi.org/10.1177/1469787412452985

[15] D. Monllaó, D. Q. Huynh, M. Reynolds, M. Dougiamas, D. Wiese, “A Supervised Learning framework for Learning Management Systems,” In proc. First International Conference on Data Science, E-Learning and Information Systems, ACM, 2018, October, pp. 1-8. https://doi.org/10.1145/3279996.3280014

[16] L. G. Moseley, D. M. Mead, “Predicting who will drop out of nursing courses: a machine learning exercise,” *Nurse Education Today*, 2008, May, vol. 28, no. 4, pp. 469-475. https://doi.org/10.1016/j.nedt.2007.07.012

[17] S. K. Murthy, “Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey,” *Data Mining and Knowledge Discovery*, 1998, vol. 2, pp. 345–389.

[18] A. S. Nurbiha, A. Zaleha, “Using Learning Analytics to Improve MOOC Instructional Design,” *International Journal of Emerging Technologies in Learning*. 2019, vol. 14. no. 24.

[19] S. Ram, Y. Wang, F. Currim, S. Currim, “Using big data for predicting freshmen retention,” In proc. International Conference on Information Systems: Exploring the Information Frontier, ICIS, 2015, ISBN: 978-0-9966831-1-1.

[20] I. Steinwart, A. Christman, “Support Vector Machines,” Springer Information Science and Statistics. 2008, LLC. ISBN: 978-0-387-77241-7.

[21] M. Tan, P. Shao, “Prediction of Student Dropout in E-Learning Program Through the Use of Machine Learning Method,” International Journal of Emerging Technologies in Learning. 2015, vol. 10. no 1. https://doi.org/10.3991/ijet.v10i1.4189

[22] Society for Learning Analytics Research. SOLAR. Online: http://www.solaresearch.org/mission/about

9 Authors

László Bognár is the professor of Applied Statistics at the University of Dunaújváros, Táncsics M. u. 1. Dunaújváros, Hungary H-2400, www.uniduna.hu. He also works for APAVE Hungary and holds trainings in different fields of industrial statistics, www.apave.hu. bognar.laszlo@uniduna.hu

Tibor Fauszt is the associate professor of Information Technology at the Budapest Business School University of Applied Sciences. Buzogány u. 10-12. Budapest, Hungary, H-1149. https://en.uni-bge.hu/. He is a member of the Informatics Sciences Committee of the Hungarian Rectors' Conference. http://www.mrk.hu/bizottsagok/informatikai-tudomanyok-bizottsaga/. fauszt.tibor@uni-bge.hu

Gábor Zsolt Nagy is an IT professional and an e-learning consultant at Eduroll Consulting, Zsókavár u. 18, Budapest, Hungary, H-1157. He is the CEO of Eduroll Consulting and he has been embracing digital technologies into organizations and companies to transform their learning programs and strategy by introducing digital learning techniques. He also worked for the European Commission and the Budapest University of Technology and Economics. eduroll.eu. gnagy@eduroll.eu

Article submitted 2020-09-09. Resubmitted 2020-11-02. Final acceptance 2020-11-05. Final version published as submitted by the authors.