

Article

Academic Success Assessment through Version Control Systems

Ángel Manuel Guerrero-Higueras ^{*}, Camino Fernández Llamas , Lidia Sánchez González ,
Alexis Gutierrez Fernández , Gonzalo Esteban Costales  and Miguel Ángel Conde González 

Robotics group, dept. of Mechanical, Computer Science, and Aerospace Engineering, University of León,
Campus de Vegazana s/n, 24071 León, Spain; camino.fernandez@unileon.es (C.F.L.);
lidia.sanchez@unileon.es (L.S.G.); alexis.gutierrez@unileon.es (A.G.F.); gestc@unileon.es (G.E.C.);
mcong@unileon.es (M.Á.C.G.)

* Correspondence: am.guerrero@unileon.es

Received: 16 January 2020; Accepted: 18 February 2020; Published: 21 February 2020



Abstract: Version control systems' usage is a highly demanded skill in information and communication technology professionals. Thus, their usage should be encouraged by educational institutions. This work demonstrates that it is possible to assess if a student can pass a computer science-related subject by monitoring its interaction with a version control system. This paper proposes a methodology that compares the performance of several machine learning models so as to select the appropriate predicting model for the assessment of the students' achievements. To fit predicting models, three subjects of the Degree in Computer Science at the University of León are considered to obtain the dataset: computer organization, computer programming, and operating systems extension. The common aspect of these subjects is their assignments, which are based on developing one or several programs with programming languages such as C or Java. To monitor the practical assignments and individual performance, a Git repository is employed allowing students to store source code, documentation, and supporting control versions. According to the presented experience, there is a huge correlation between the level of interaction for each student and the achieved grades.

Keywords: computer programming; version control system; machine learning; learning analytics

1. Introduction

Learning Analytics (LA) is defined in [1] as the measurement, collection, analysis, and reporting of data about learners and their contexts in order to understand and optimize the learning process and its environments. The use of LA provides many advantages such as carrying out data mining in education and also helping lecturers to optimize their teaching as they obtain current and actual feedback of each student's learning process. Therefore, by analyzing data from both Student Institutional Systems (SISs) or Learning Management Systems (LMSs), information about the students' interaction is gathered. Such data allow lecturers to identify patterns that point out if a resource is adequate or if the learning outcomes can be achieved.

According to recent literature, LA research is widely applied to estimate academic results. In particular, in [2], the students' probability to pass or fail a course was determined. By employing predictive analytics, those students whose chances to pass a subject were lower could be identified, so an intervention could be adapted in order to mitigate the academic failure. In other words, the impact on education was truly positive. However, these methods depend on the field where they are applied, the sample features, or the data source [3], as well as what has an influence on the obtained accuracy. In existing methodologies that use learning algorithms, the quality data are an essential requirement;

such a drawback avoids detecting the influence of different features on academic success, as well as designing a ready-to-use tool that can be applied in general.

On the contrary, the Module Evaluator (MoEv) tool presented in [4,5] handled these problems by evaluating different parametric and non-parametric learning algorithms, automatically identifying the more accurate one for a specific problem. MoEv follows the methodology proposed in [6], but it appends a crucial step so as to generalize models: the feature extraction is done automatically by choosing the most discriminant variables. In order to design a general tool that makes it possible to analyze any data source, the first stage is to identify the most valuable information for the model. Usually, predictive models on education consider data sources such as SISs [7], the trace data recorded by LMSs and other learning environments [8], or hybrid data sources formed by a combination of them [9].

On the other hand, the use of a Version Control System (VCS) is an indispensable skill for Information and Communications Technology (ICT) professionals and demanded by companies [10]. Thus, it is widely used by lecturers during practical assignments so as to provide a meaningful learning experience.

In order to facilitate software development, there are specific tools that manage the code changing named version control [11]. In this field, a version means how the product is at a certain time. Moreover, these tools usually provide version management since it is a hard task if it is done manually. Not only a VCS has to provide storage for the source code, images, or documentation, but also they can be editable, keeping a record of every performed action so as to revert changes at any time in case this is needed. This last feature is very important since it makes it possible to return an element to its previous state, which is quite common in programming. Some examples of popular VCSs are CVS, Subversion [12], or Git [13].

The authors aim to estimate students' academic achievements by supervising their daily performance with VCSs. In order to do so, there are two issues with which to deal. The first one is if academic success is related to the considered variables that are computed from the students' interaction with VCSs. The second is if it is possible to design a model that analyses those features in order to estimate the students' success automatically.

Preliminary results of this approach were presented in [4,5,14]. A proof of concept about using the features extracted from the interaction with VCSs to predict academic success was presented in [14]. To build the prediction models, a single dataset gathered data from the Operating Systems Extension Subject of the degree in Computer Science at the University of León. Specifically, the dataset contained data from the 2016-2017 course. In [4,5], in order to build more accurate predictions models, the authors used two different datasets also gathering data from the Operating Systems Extension Subject. The first dataset, which contained data from the 2016-2017 course, was used to fit and test the models. The second one, which contained data from the 2017-2018 course, was used to evaluate that such models ensured an optimal generalization. This subject was selected because students are required to complete a practical assignment, which implies developing software solutions so as to pass the course. In the required practical assignments of this subject, a VCS was employed to store both source code and documentation. Regarding the first issue described before, both works demonstrated that some variables related to students' interaction with VCSs were discriminant. Relating to the second one mentioned earlier, both works presented models that allowed the prediction of students' success by analyzing such variables. The main difference between [4] and [5] had to do with the contents of the datasets. In [4], the dataset contained not only variables related to the interaction with VCSs, but also some additional variables, which increased model accuracy. In [5], only variables related to the interaction with VCSs were used. The accuracy of the models described in [4] was higher, but the authors considered the models presented in [5] more interesting since they may be used to build a general and ready-to-use tool.

To ensure an optimal generalization of prediction models, the use of a much bigger dataset was proposed in [15]. A new dataset was built not only gathering data from the Operating Systems Extension subject, but also from Computer Programming I and Computer Organization subjects. These subjects also belong to the degree in Computer Science at the University of León. As in

Operating Systems Extension, students had to develop software solutions using a VCS to store both source code and documentation.

Besides the data related to the interaction with VCSs, the work in [15] proposed including some additional variables in the dataset extracted from the source code. These were the total number of source code lines of the software solution, the number of used subroutines, or the number of variables or objects, among others. The authors considered that these variables may be used to analyze the quality of the source code and, thus, study the evolution of students' programming skills.

In this work, the authors built new prediction models by using the dataset presented in [15]. The new models presented a higher accuracy than the ones described in [5] and a similar accuracy to the models in [4]. It is important to point out that the optimized models proposed here, contrary to the models presented in [4], ensured an optimal generalization since they just used variables related to the interaction with VCSs and their contents.

The organization of this paper is as follows: Section 2 includes a description of the empirical evaluation of the considered classification algorithms, the experience details, the materials, and methods used; Section 3 shows the obtained results; the discussion of the results is presented in Section 4; Section 5 gathers the conclusions and future work.

2. Materials and Methods

This section includes a description of all the elements used to design and evaluate models to estimate students' academic success through their interaction with a VCSs. First, a dataset was built by gathering data from several practical assignments from subjects of the degree in Computer Science at the University of León. In order to do so, a commercial VCS was used. Then, MoEv was employed to get an optimized model. Each element is further outlined below.

2.1. Version Control System

All the considered practical assignments required the use of a VCS; in particular, a Git repository. Git follows a distributed scheme where each copy of the repository also encloses the whole change history, unlike other systems that follow the client-server models [16]. As the repositories were public, the GitHub Classroom platform was considered so as to use the organizing functionalities that it presents and also to offer private repositories for students [17]. GitHub is a website and service that uses the Git revision control system facilitating the development of software projects. Otherwise, GitHub Classroom provides different useful functionalities for lecturers such as task assignments for students or groups of students all in the same hosting service.

2.2. Input Data

In this paper, the considered data were extracted from the practical assignments proposed in different subjects of the Computer Science Degree at the University of León. In particular, the subjects that participated in this experience were Operating Systems Extension, Computer Programming I, and Computer Organization.

There were two subjects for first-year students: Computer Programming I and Computer Organization. In the case of Computer Programming I, it deals with the fundamentals of software programming, that is design, implementation, and testing. Therefore, students acquire the competence of designing at a certain abstract level for different algorithms and patterns. In addition to this, students learn to design and code test cases.

Computer Organization is also a first-year course, and its objectives are related to the fundamentals of computers such as datapath, control unit, memory, or input/output. In the practical lessons, students learn how to program in both C and assembler languages.

Finally, the Operating Systems Extension is a second-year course where students deal with operating systems. More specifically, students acquire knowledge about how the main memory works

(volatile) and how the file system is managed (non-volatile). This subject also addresses security issues in operating systems.

The reason to select the previously mentioned subjects was that all of them involve practical assignments where students have to develop a program in a certain language. Moreover, the use of a Git repository is mandatory in order to store both source code and documentation. Table 1 shows the number of samples for each subject.

Table 1. Number of samples in the dataset.

Subject	Year	Course	Practical Assignments	Support
Computer Programming I	1°	2018–2019	3	372
Computer Organization	1°	2018–2019	5	240
Operating Systems Extension	2°	2016–2017	1	72
		2017–2018	1	47

Regarding the feature extraction, the dataset was formed as follows. Considering the practical assignments developed by students and its activity on the repository, different variables were considered:

- An identifier to make students anonymous (*id*): it provided differentiation among the samples.
- Commits counter (*commits*): it represented the number of commit operations made by a given student.
- Number of days where at least one commit was done (*days*).
- Average number of commit operations per day (*commits/day*).
- Number of additional lines appended to the source code during the assignment development (*additions*).
- Number of deleted lines removed from the source code during the assignment development (*deletions*).
- Number of issues opened (*issues-opened*) during the assignment.
- Number of issues closed (*issues-closed*) during the assignment.

In [15], the authors proposed to collect some additional variables extracted from the source code: the total number of source code lines, the number of subroutines, the number of variables or objects, the number of classes, the number of loops, the cyclomatic complexity, the number of files, some object-oriented programming related features (i.e., inheritance and polymorphism usage), the number of README files, dependencies, etc. They also considered variables computed by the measure of the interaction with VCS. These data may be later used to analyze the quality of the source code and, thus, study the evolution of students' programming skills. Specifically, in this work, the following additional variables were considered: the number of source code files written in C (including header files), Java, and the assembler; as well as the number of source code lines in each file.

2.3. Models Design and Evaluation

In this work, the MoEv tool was employed in order to achieve the model that performed the best prediction. The followed methodology was presented in [4,5]. Figure 1 shows a scheme of the applied methodology.

The feature vector of each sample was formed by two kinds of variables. One type was those variables extracted from raw data given by an SIS or an LMS directly. In this work, the feature vector had the following raw data elements: *id*, *commits*, *additions*, *deletions*, *issues-opened*, and *issues-closed*. In addition to this, source code metrics also are considered as raw data. Besides raw data, there were synthetic data represented by the following variables: *days* (number of days with at least a commit operation) and *commits/day* (average commit operations per day).

Once the feature vector was built, a target variable (class) was included. That was a component that determined the class to which a certain feature vector belonged. This class element allowed the

model to train and test the supervised learning procedure. The considered target was a binary variable with a Pvalue if a given student will pass the course and a Fvalue if a given student will fail.

Following the methodology shown in Figure 1, after collecting the data step, those features that were more significant had to be identified so as to obtain a classification model that assessed the target variable. This stage, named feature selection, was a procedure that determined the contribution of each feature, identifying the higher influences when a classification or prediction was done. By performing a feature selection before building the model, the overfitting was reduced, improving the accuracy and, therefore, the training time.

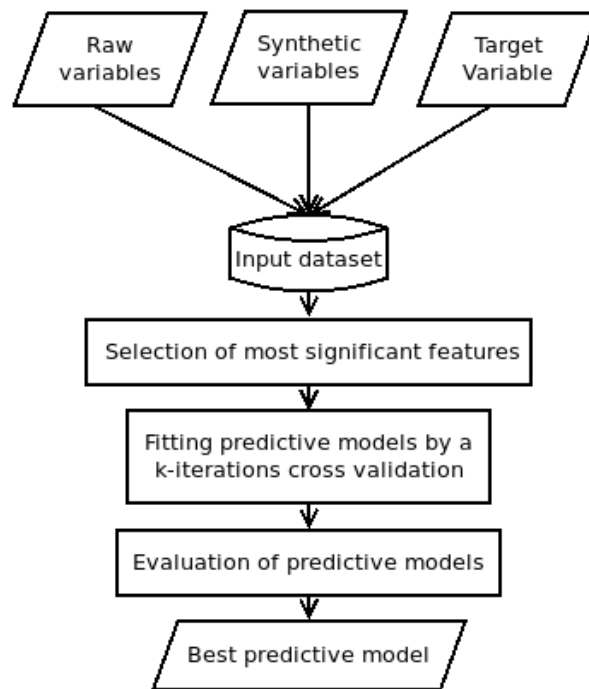


Figure 1. MoEv operation scheme as shown in [5].

According to the scores presented in [14], the tree-base algorithm achieved good rates for the database considered in this paper. Thus, the feature selection was computed by employing an extremely randomized tree [18]. The decision criterion to select a feature was based on the value of the Gini coefficient (\mathcal{G}) computed for each feature; if \mathcal{G} was higher than a certain threshold, the feature was then significant; in this paper, the threshold value was fixed to 0.1, obtained empirically.

After the feature selection, several prediction models were fitted. We used 80% of the input data to train the models and 20% to test them. The following Machine Learning (ML) algorithms were considered since they are well-known in this field:

- Adaptive Boosting (AB). Ensemble methods are techniques that combine different basic classifiers turning a weak learner into a more accurate method. Boosting is one of the most successful types of ensemble methods and AB one of the most popular boosting algorithms.
- Classification And Regression Tree (CART). A decision tree is a method that predicts the label associated with an instance by traveling from a root node of a tree to a leaf [19]. It is a non-parametric method in which the trees are grown in an iterative, top-down process.
- K-Nearest Neighbors (KNN). Although the nearest neighbor concept is the foundation of many other learning methods, notably unsupervised, supervised neighbor-based learning is also available to classify data with discrete labels. It is a non-parametric technique that classifies new observations based on the distance to the observation in the training set. A good presentation of the analysis was given in [20,21].

- Linear Discriminant Analysis (LDA). This is a parametric method that assumes that distributions of the data are multivariate Gaussian [21]. Furthermore, LDA assumes knowledge of population parameters. In another case, the maximum likelihood estimator can be used. LDA uses Bayesian approaches to select the category, which maximizes the conditional probability; see [22–24].
- Logistic Regression (LR). Linear methods are intended for regressions in which the target value is expected to be a linear combination of the input variables. LR, despite its name, is a linear model for classification rather than regression. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.
- Multi-Layer Perceptron (MLP). An artificial neural network is a model inspired by the structure of the brain. Neural networks are used when the type of relationship between inputs and outputs is not known. The network is organized into layers of nodes (an input layer, an output layer, and one or more hidden layers). These layers are organized in a directed graph so that each layer is fully connected to the next one. An MLP is a modification of the standard linear perceptron whose best characteristic is that it is able to distinguish data that are not linearly separable. An MLP uses back-propagation for training the network; see [25,26].
- Naive Bayes (NB). This method is based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features; see [21,27].
- Random Forest (RF). This is a classifier consisting of a collection of decision trees, in which each tree is constructed by applying an algorithm to the training set and an additional random vector that is sampled via bootstrap re-sampling [28].

Finally, experiments were carried out with the MoEv tool that supports k-iteration cross-validation so as to identify the prediction algorithm more appropriate for a given problem. To perform it, MoEv calculated a set of well-known Key Performance Indicators (KPIs). Each model performance was measured by considering the accuracy of classification (see Equation (1)):

$$accuracy = \frac{\sum T_p + \sum T_n}{\sum \text{total data}} \quad (1)$$

T_p being the number of students who pass the course and the model predicts so (true positives) and T_n the number of students who do not pass the course and the model predicts so (true negatives).

The three models with the highest accuracy classification score were pre-selected for in-depth evaluation by considering the following KPIs: Precision (P), Recall (R), and F1-score obtained through the confusion matrix.

The Precision (P) is computed as follows (Equation (2)):

$$P = \frac{\sum T_p}{\sum T_p + \sum F_p} \quad (2)$$

F_p being the number of students who do not pass the course and the model says they pass (false positives).

The Recall (R) is defined in Equation (3), where F_n is the number of students who pass the course and the model says they fail (false negatives):

$$R = \frac{\sum T_p}{\sum T_p + \sum F_n} \quad (3)$$

The F1-score relates those metrics since it is the harmonic mean of precision and recall, as shown in Equation (4).

$$F_1 = 2 \frac{P \times R}{P + R} \quad (4)$$

3. Results

Feature importance was computed with the Gini coefficient, as is shown in Figure 2. Features with a low Gini coefficient were discarded ($\mathcal{G} < 0.1$). Selected features were days ($\mathcal{G} = 0.27$), lines ($\mathcal{G} = 0.24$), additions ($\mathcal{G} = 0.16$), commits ($\mathcal{G} = 0.14$), and files ($\mathcal{G} = 0.1$).

Table 2 gathers the accuracy calculated by using a 10-iteration execution of MoEv. As shown in the table, the best results were obtained with RF (0.78), KNN (0.72), and AB (0.7).

In Figure 3, the confusion matrices for RF, KNN, and AB models are presented.

Precision (P), Recall (R), and F_1 -score obtained through the confusion matrix from Figure 3 are gathered in Table 3.

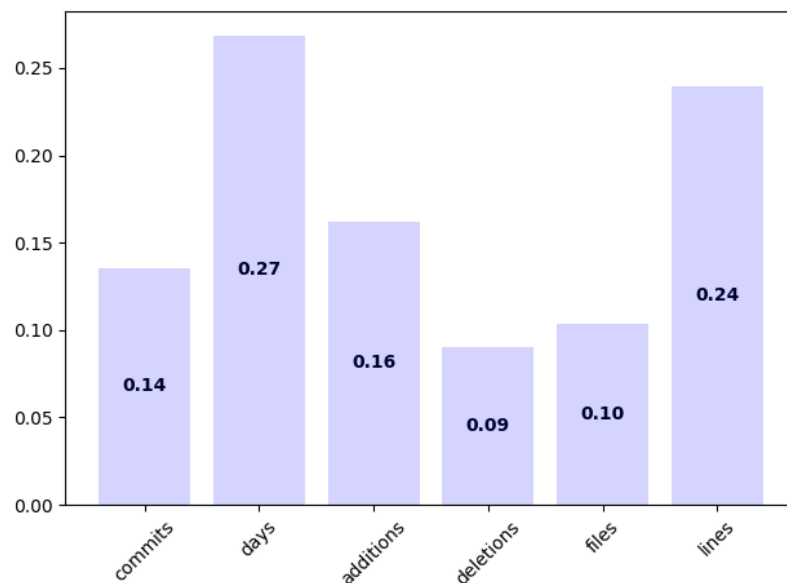


Figure 2. Feature importance.

Table 2. Accuracy classification scores.

	Classifier							
	RF	KNN	AB	LR	LDA	CART	NB	MLP
Score	0.78	0.72	0.70	0.67	0.67	0.67	0.63	0.63

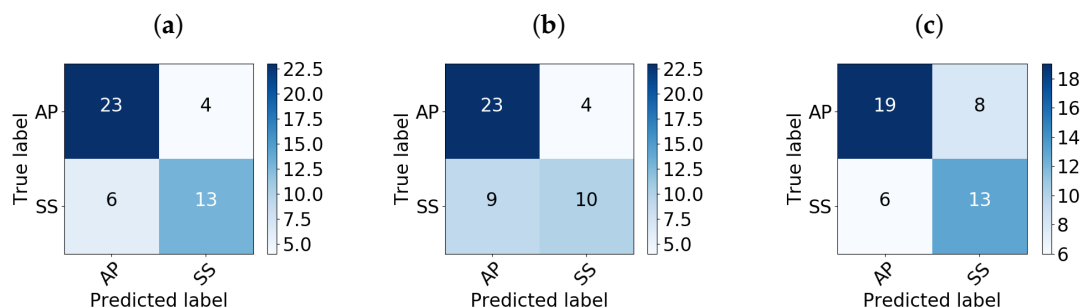


Figure 3. Confusion matrix for RF (a), KNN (b), and AB (c) models.

Table 3. Precision, recall and F_1 -score for the RF, KNN, and AB models.

Classifier	Class	P	R	F_1 -Score	Support
RF	P	0.79	0.85	0.82	27
	F	0.76	0.68	0.82	19
	avg/total	0.78	0.78	0.78	46
KNN	P	0.72	0.85	0.78	27
	F	0.71	0.53	0.61	19
	avg/total	0.72	0.72	0.71	46
AB	P	0.76	0.70	0.73	27
	F	0.62	0.68	0.65	19
	avg/total	0.70	0.70	0.79	46

4. Discussion

Feature importance in Figure 2 showed that the number of days with at least one commit operation (days) was the most discriminant feature ($\mathcal{G} = 0.27$). This is logical since working in a software solution usually implies different phases and milestones. A low value in this feature points out that the student had not achieved the expected milestones. In addition to this, the number of days with at least one commit operation had higher importance than the number of commit operations itself; this had to do with the fact that several milestones usually cannot be achieved the same day. Having multiple commit operations the same day usually means that students are using several commit operations for a single milestone. This may happen if there are unexpected errors due to untested software or even that the students are trying to cheat the model.

The next discriminant feature was the number of source code lines ($\mathcal{G} = 0.24$). This allowed us to assert that the contents of a repository may be quite useful to predict academic success. Including more additional features such as the ones proposed in [15] in addition to the number of files and lines may result in higher accuracy. However, it is important to point out that this requires a static analysis of the source code. This analysis is different for each programming language, so it complicates considerably the data collection, although programming paradigms are quite common between languages. This is an important drawback since data collection is actually quite complicated and time demanding. However, the number of programming languages that are used during the Computer Science degree is not so high, so the method presented in this paper can be easily adapted to other subjects.

Regarding the models' accuracy, RF, KNN, and AB had the highest scores, as shown in Table 2. RF was also the model with the highest accuracy score in [5]. Table 4 shows the complete comparison between the accuracy scores obtained in this work shown in Table 2 and the accuracy scores obtained in [5]. The first line at Table 4 shows the accuracy score obtained in this work. The second line shows the accuracy scores obtained in [5]. It is important to point out that all models increased their accuracy scores. This was due to the fact that in this work, the training dataset was much bigger and complete (since it included data from several subjects) compared to the dataset used in [5]. In addition, considering additional significant features also implied an accuracy improvement.

Table 4. Accuracy scores' comparison between the models proposed in this work (first line) and the models proposed in [5] (second line).

	Classifier							
	RF	KNN	AB	LR	LDA	CART	NB	MLP
Score of the models proposed at this work	0.78	0.72	0.70	0.67	0.67	0.67	0.63	0.63
Score of the models proposed in [5]	0.70	0.60	0.40	0.60	0.60	0.60	0.50	0.40

Finally, considering the most accurate models, i.e., RF, KNN, and AB, a deeper analysis through their confusion matrices was given in order to analyze their sensitivity for detecting Ps: i.e., the rate of true

Ps that the model classified incorrectly. Figure 3 and Table 3 show that the RF classifier obtained better values for precision (P), recall (R), and F_1 -score than KNN and AB in both the test and validation stages.

5. Conclusions

The methodology presented in this paper allowed lecturers to predict the results students achieved by supervising their daily performance using a VCS. The employed models were fitted by using a dataset that gathered data from different subjects of the Computer Science degree at the University of León. All the considered subjects had two common items: all of them required students to develop software solutions and to use a VCS for storing source code and documentation.

The dataset included variables from the students' interaction with VCSs such as the number of commit operations, source code lines added and deleted, etc. In addition, the dataset included data related to the contents of students' repositories such as the number of source code files and lines. These additional variables, although not coming from the interaction with VCS, were source code features; thus, the models ensured an optimal generalization.

The results presented in Section 3 and discussed in Section 4 demonstrated that the models' accuracy scores increased by using a bigger dataset, not only including additional variables, but also much more samples. The accuracy was also similar to the results obtained with models trained with datasets containing specific features, and thus, and contrary to these models, the ones proposed in this paper ensured an optimal generalization.

Further works should face the usage of additional features collected within the source code itself to analyze the quality of the source code and, thus, study the evolution of students' programming skills.

Author Contributions: Conceptualization, Á.M.G.-H.; methodology, Á.M.G.-H., C.F.L., L.S.G., and M.Á.C.G.; software, Á.M.G.-H., A.G.F., and G.E.C.; validation, C.F.L., L.S.G., and M.Á.C.G.; investigation, Á.M.G.-H., C.F.L., and L.S.G.; resources, C.F.L.; data curation, A.G.F., and G.E.C.; writing—original draft preparation, Á.M.G.-H. and L.S.G.; writing—review and editing, C.F.L., A.G.F., G.E.C., and M.Á.C.G.; visualization, Á.M.G.-H.; supervision, M.Á.C.G.; project administration, Á.M.G.-H.; funding acquisition, C.F.L. All authors read and agreed to the published version of the manuscript.

Funding: The research described in this article was partially funded by the Support Plan for Teaching Innovation Groups (PAGID) of the University of León 2018 and by the European Commission under Grant 2018-1-ES01-KA201-050939: ROBOSTEAM—Integrating Steam and Computational Thinking development by using Robotics and Physical Devices.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

Abbreviations

List of abbreviations used in this paper:

AB	Adaptive Boosting
CART	Classification And Regression Tree
ICT	Information and Communications Technology
KNN	K-Nearest Neighbors
KPI	Key Performance Indicator
LA	Learning Analytics
LDA	Linear Discriminant Analysis
LMS	Learning Management System
LR	Logistic Regression
ML	Machine Learning
MLP	Multi-Layer Perceptron
MoEv	Module Evaluator

NB	Naive Bayes
RF	Random Forest
SIS	Student Institutional System
VCS	Version Control System

References

- Siemens, G.; Gasevic, D. Guest editorial-Learning and knowledge analytics. *Educ. Technol. Soc.* **2012**, *15*, 1–2.
- Siemens, G.; Dawson, S.; Lynch, G. *Improving the Quality and Productivity of the Higher Education Sector: Policy and Strategy for Systems-Level Deployment of Learning Analytics*; Society for Learning Analytics Research for the Australian Office for Learning and Teaching: Canberra, Australia, 2013.
- Gašević, D.; Dawson, S.; Rogers, T.; Gasevic, D. Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success. *Internet High. Educ.* **2016**, *28*, 68–84. [\[CrossRef\]](#)
- Guerrero-Higueras, Á.M.; DeCastro-García, N.; Matellán, V.; Conde, M.Á. Predictive models of academic success: a case study with version control systems. In Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain, 24–26 October 2018; pp. 306–312.
- Guerrero-Higueras, Á.M.; DeCastro-García, N.; Rodríguez-Lera, F.J.; Matellán, V.; Conde, M.Á. Predicting academic success through students' interaction with Version Control Systems. *Open Comput. Sci.* **2019**, *9*, 243–251. [\[CrossRef\]](#)
- Guerrero-Higueras, Á.M.; DeCastro-García, N.; Matellán, V. Detection of Cyber-attacks to indoor real time localization systems for autonomous robots. *Robot. Auton. Syst.* **2018**, *99*, 75–83. [\[CrossRef\]](#)
- Kovacic, Z. Predicting student success by mining enrolment data. *Res. High. Educ. J.* **2012**, *15*, 1–20.
- Agudo-Peregrina, Á.F.; Iglesias-Pradas, S.; Conde-González, M.Á.; Hernández-García, Á. Can we predict success from log data in VLEs? Classification of interactions for learning analytics and their relation with performance in VLE-supported F2F and online learning. *Comput. Hum. Behav.* **2014**, *31*, 542–550. [\[CrossRef\]](#)
- Barber, R.; Sharkey, M. Course correction: Using analytics to predict course success. In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, Vancouver, BC, Canada, 29 April–2 May 2012; pp. 259–262.
- Spinellis, D. Version control systems. *IEEE Softw.* **2005**, *22*, 108–109. [\[CrossRef\]](#)
- Fischer, M.; Pinzger, M.; Gall, H. Populating a release history database from version control and bug tracking systems. In Proceedings of the International Conference on Software Maintenance, Amsterdam, The Netherlands, 22–26 September 2003; pp. 23–32.
- Pilato, C.M.; Collins-Sussman, B.; Fitzpatrick, B.W. *Version Control with Subversion: Next Generation Open Source Version Control*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2008.
- Torvalds, L.; Hamano, J. Git: Fast Version Control System. Available online: <http://git-scm.com> (accessed on 21 February 2020).
- Guerrero-Higueras, A.M.; Matellán-Olivera, V.; Esteban-Costales, G.; Fernández-Llamas, C.; Rodríguez-Sedano, F.J.; Ángel, C.M. Model for evaluating student performance through their interaction with version control systems. In Proceedings of the Learning Analytics Summer Institute (LASI), New York, NY, USA, 11–13 June 2018.
- Guerrero-Higueras, Á.M.; Sánchez-González, L.; Fernández-Llamas, C.; Conde, M.Á.; Lera, F.J.R.; Sedano, F.J.R.; Costales, G.E.; Matellán, V. Prediction of academic success through interaction with version control systems. In Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain, 24–26 October 2018; pp. 284–289.
- De Alwis, B.; Sillito, J. Why are software projects moving from centralized to decentralized version control systems? In Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering, Vancouver, BC, Canada, 17 May 2009; pp. 36–39.
- Griffin, T.; Seals, S. Github in the classroom: Not just for group projects. *J. Comput. Sci. Coll.* **2013**, *28*, 74–74.
- Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [\[CrossRef\]](#)
- Friedman, J.; Hastie, T.; Tibshirani, R. *The Elements of Statistical Learning*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 1.
- Devroye, L.; Györfi, L.; Lugosi, G. *A Probabilistic Theory of Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 31.

21. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
22. Bishop, C.M. Pattern recognition. *Mach. Learn.* **2006**, *128*, 1–58.
23. Koller, D.; Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*; MIT Press: Cambridge, MA, USA, 2009.
24. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; MIT Press: Cambridge, MA, USA, 2012.
25. Rummelhart, D.E. Learning internal representations by error propagation. In *Parallel Distributed Processing*; 1986.
26. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.* **1989**, *2*, 303–314. [[CrossRef](#)]
27. Zhang, H. The optimality of naive Bayes. *AA* **2004**, *1*, 3.
28. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Artikel

Penilaian Keberhasilan Akademik melalui Versi Sistem kontrol

Angel Manuel Guerrero-Higueras[✉], Camino Fernandez Llamas^{ID}, Lidia Sánchez González^{ID}, Alexis Gutierrez Fernandez^{ID}, Gonzalo Esteban Costales^{ID} dan Miguel Angel Conde González^{ID}

Grup robotika, dept. Mekanik, Ilmu Komputer, dan Teknik Dirgantara, Universitas León, Campus de Vegazana s/n, 24071 León, Spanyol; camino.fernandez@unileon.es (CFL); lidia.sanchez@unileon.es (LSG); alexis.gutierrez@unileon.es (AGF); gestic@unileon.es (GEC); mcong@unileon.es (M.Á.CG)

* Korespondensi: am.guerrero@unileon.es

Diterima: 16 Januari 2020; Diterima: 18 Februari 2020; Diterbitkan: 21 Februari 2020



Abstrak: Penggunaan sistem kontrol versi merupakan keterampilan yang sangat dibutuhkan oleh para profesional teknologi informasi dan komunikasi. Dengan demikian, penggunaannya harus didorong oleh lembaga pendidikan. Karya ini menunjukkan bahwa adalah mungkin untuk menilai apakah seorang siswa dapat lulus mata pelajaran yang berhubungan dengan ilmu komputer dengan memantau interaksinya dengan sistem kontrol versi. Makalah ini mengusulkan metodologi yang membandingkan kinerja beberapa model pembelajaran mesin untuk memilih model prediksi yang tepat untuk penilaian prestasi siswa. Agar sesuai dengan model prediksi, tiga mata pelajaran dari Gelar Ilmu Komputer di Universitas León dianggap untuk mendapatkan kumpulan data: organisasi komputer, pemrograman komputer, dan ekstensi sistem operasi. Aspek umum dari mata pelajaran ini adalah tugas mereka, yang didasarkan pada pengembangan satu atau beberapa program dengan bahasa pemrograman seperti C atau Java. Untuk memantau tugas praktis dan kinerja individu, repositori Git digunakan yang memungkinkan siswa untuk menyimpan kode sumber, dokumentasi, dan versi kontrol pendukung. Menurut pengalaman yang disajikan, terdapat korelasi yang sangat besar antara tingkat interaksi setiap siswa dengan nilai yang dicapai.

Kata kunci: pemrograman komputer; sistem kontrol versi; pembelajaran mesin; analisis pembelajaran

1. Perkenalan

Learning Analytics (LA) didefinisikan dalam [1] sebagai pengukuran, pengumpulan, analisis, dan pelaporan data tentang peserta didik dan konteksnya untuk memahami dan mengoptimalkan proses pembelajaran dan lingkungannya. Penggunaan LA memberikan banyak keuntungan seperti melakukan data mining dalam dunia pendidikan dan juga membantu dosen untuk mengoptimalkan pengajarannya karena mendapatkan feedback yang aktual dan aktual dari setiap proses belajar mahasiswa. Oleh karena itu, dengan menganalisis data dari Sistem Kelembagaan Siswa (SIS) atau Sistem Manajemen Pembelajaran (LMS), informasi tentang interaksi siswa dikumpulkan. Data tersebut memungkinkan dosen untuk mengidentifikasi pola yang menunjukkan apakah sumber daya memadai atau jika hasil pembelajaran dapat dicapai.

Menurut literatur terbaru, penelitian LA secara luas diterapkan untuk memperkirakan hasil akademik. Secara khusus, dalam [2], probabilitas siswa untuk lulus atau gagal dalam suatu kursus ditentukan. Dengan menggunakan analitik prediktif, para siswa yang peluangnya untuk lulus mata pelajaran lebih rendah dapat diidentifikasi, sehingga intervensi dapat disesuaikan untuk mengurangi kegagalan akademik. Dengan kata lain, dampak terhadap pendidikan benar-benar positif. Namun, metode ini tergantung pada bidang di mana mereka diterapkan, fitur sampel, atau sumber data [3], serta apa yang mempengaruhi akurasi yang diperoleh. Dalam metodologi yang ada yang menggunakan algoritma pembelajaran, kualitas data merupakan persyaratan penting;

kelemahan seperti itu menghindari pendeteksian pengaruh fitur yang berbeda pada keberhasilan akademik, serta merancang alat siap pakai yang dapat diterapkan secara umum.

Sebaliknya, alat Modul Evaluator (MoEv) yang disajikan dalam [4,5] menangani masalah ini dengan mengevaluasi berbagai algoritma pembelajaran parametrik dan non-parametrik, secara otomatis mengidentifikasi yang lebih akurat untuk masalah tertentu. MoEv mengikuti metodologi yang diusulkan dalam [6], tetapi menambahkan langkah penting untuk menggeneralisasi model: ekstraksi fitur dilakukan secara otomatis dengan memilih variabel yang paling diskriminan. Untuk merancang alat umum yang memungkinkan untuk menganalisis sumber data apa pun, tahap pertama adalah mengidentifikasi informasi paling berharga untuk analisis. Biasanya, model prediktif pada pendidikan mempertimbangkan sumber data seperti SIS [7], data jejak yang direkam oleh LMS dan lingkungan belajar lainnya [8], atau sumber data hybrid yang dibentuk oleh kombinasinya [9].

Di sisi lain, penggunaan Version Control System (VCS) adalah keterampilan yang sangat diperlukan bagi para profesional Teknologi Informasi dan Komunikasi (TIK) dan dituntut oleh perusahaan [10]. Sehingga banyak digunakan oleh dosen pada saat tugas praktek sehingga memberikan pengalaman belajar yang bermakna.

Untuk memfasilitasi pengembangan perangkat lunak, ada alat khusus yang mengelola perubahan kode yang disebut kontrol versi [11]. Di bidang ini, versi berarti bagaimana produk pada waktu tertentu. Selain itu, alat ini biasanya menyediakan manajemen versi karena ini adalah tugas yang sulit jika dilakukan secara manual. VCS tidak hanya harus menyediakan penyimpanan untuk kode sumber, gambar, atau dokumentasi, tetapi juga dapat diedit, menyimpan catatan setiap tindakan yang dilakukan untuk mengembalikan perubahan kapan saja jika diperlukan. Fitur terakhir ini sangat penting karena memungkinkan untuk mengembalikan elemen ke keadaan sebelumnya, yang cukup umum dalam pemrograman. Beberapa contoh VCS yang populer adalah CVS, Subversion [12], atau Git [13].

Penulis bertujuan untuk memperkirakan prestasi akademik siswa dengan mengawasi kinerja harian mereka dengan VCS. Untuk itu, ada dua masalah yang harus dihadapi. Yang pertama adalah jika keberhasilan akademik terkait dengan variabel yang dipertimbangkan yang dihitung dari interaksi siswa dengan VCS. Kedua, jika memungkinkan untuk merancang model yang menganalisis fitur-fitur tersebut untuk memperkirakan keberhasilan siswa secara otomatis.

Hasil awal dari pendekatan ini disajikan di [4,5,14]. Bukti konsep tentang penggunaan fitur yang diekstraksi dari interaksi dengan VCS untuk memprediksi keberhasilan akademik disajikan pada [14]. Untuk membangun model prediksi, satu kumpulan data mengumpulkan data dari Subjek Ekstensi Sistem Operasi dari gelar Ilmu Komputer di Universitas León. Secara khusus, dataset berisi data dari kursus 2016-2017. Dalam [4,5], untuk membangun model prediksi yang lebih akurat, penulis menggunakan dua kumpulan data berbeda yang juga mengumpulkan data dari Subjek Ekstensi Sistem Operasi. Dataset pertama, yang berisi data dari kursus 2016-2017, digunakan untuk menyesuaikan dan menguji model. Yang kedua, yang berisi data dari kursus 2017-2018, digunakan untuk mengevaluasi bahwa model tersebut memastikan generalisasi yang optimal. Mata kuliah ini dipilih karena mahasiswa dituntut untuk menyelesaikan tugas praktikum yang berimplikasi pada pengembangan solusi perangkat lunak sehingga dapat lulus mata kuliah tersebut. Dalam tugas praktis yang diperlukan dari subjek ini, VCS digunakan untuk menyimpan kode sumber dan dokumentasi. Mengenai masalah pertama yang dijelaskan sebelumnya, kedua karya menunjukkan bahwa beberapa variabel yang terkait dengan interaksi siswa dengan VCS adalah diskriminan. Berkaitan dengan yang kedua yang disebutkan sebelumnya, kedua karya tersebut menyajikan model yang memungkinkan prediksi keberhasilan siswa dengan menganalisis variabel tersebut. Perbedaan utama antara [4] dan [5] berkaitan dengan isi kumpulan data. Dalam [4], dataset berisi tidak hanya variabel yang terkait dengan interaksi dengan VCS, tetapi juga beberapa variabel tambahan, yang meningkatkan akurasi model. Dalam [5], hanya variabel yang terkait dengan interaksi dengan VCS yang digunakan. Keakuratan model yang dijelaskan dalam [4] lebih tinggi, tetapi penulis menganggap model yang disajikan dalam [5] lebih menarik karena dapat digunakan untuk membangun alat yang umum dan siap pakai.

Untuk memastikan generalisasi yang optimal dari model prediksi, penggunaan dataset yang jauh lebih besar diusulkan pada [15]. Sebuah dataset baru dibangun tidak hanya mengumpulkan data dari mata pelajaran Ekstensi Sistem Operasi, tetapi juga dari mata pelajaran Pemrograman Komputer I dan Organisasi Komputer. Mata pelajaran ini juga termasuk dalam gelar Ilmu Komputer di Universitas León. Seperti dalam

Ekstensi Sistem Operasi, siswa harus mengembangkan solusi perangkat lunak menggunakan VCS untuk menyimpan kode sumber dan dokumentasi.

Selain data yang terkait dengan interaksi dengan VCS, pekerjaan di [15] yang diusulkan termasuk beberapa variabel tambahan dalam dataset yang diekstraksi dari kode sumber. Ini adalah jumlah total baris kode sumber dari solusi perangkat lunak, jumlah subrutin yang digunakan, atau jumlah variabel atau objek, antara lain. Penulis menganggap bahwa variabel ini dapat digunakan untuk menganalisis kualitas kode sumber dan, dengan demikian, mempelajari evolusi keterampilan pemrograman siswa.

Dalam karya ini, penulis membangun model prediksi baru dengan menggunakan dataset yang disajikan pada [15]. Model-model baru menyajikan akurasi yang lebih tinggi daripada yang dijelaskan dalam [5] dan akurasi yang sama dengan model di [4]. Penting untuk menunjukkan bahwa model yang dioptimalkan yang diusulkan di sini, bertentangan dengan model yang disajikan dalam [4], memastikan generalisasi yang optimal karena mereka hanya menggunakan variabel yang terkait dengan interaksi dengan VCS dan isinya.

Organisasi makalah ini adalah sebagai berikut: Bagian 2 mencakup deskripsi evaluasi empiris dari algoritma klasifikasi yang dipertimbangkan, detail pengalaman, bahan, dan metode yang digunakan; Bagian 3 menunjukkan hasil yang diperoleh; pembahasan hasil disajikan pada Bagian 4; Bagian 5 mengumpulkan kesimpulan dan pekerjaan di masa depan.

2. Bahan-bahan dan metode-metode

Bagian ini mencakup deskripsi semua elemen yang digunakan untuk merancang dan mengevaluasi model untuk memperkirakan keberhasilan akademik siswa melalui interaksi mereka dengan VCS. Pertama, kumpulan data dibangun dengan mengumpulkan data dari beberapa tugas praktis dari mata pelajaran gelar Ilmu Komputer di Universitas León. Untuk melakukannya, VCS komersial digunakan. Kemudian, MoEv digunakan untuk mendapatkan model yang dioptimalkan. Setiap elemen diuraikan lebih lanjut di bawah ini.

2.1. Sistem Kontrol Versi

Semua tugas praktis yang dipertimbangkan membutuhkan penggunaan VCS; khususnya, repositori Git. Git mengikuti skema terdistribusi di mana setiap salinan repositori juga menyertakan seluruh riwayat perubahan, tidak seperti sistem lain yang mengikuti model client-server [16]. Karena repositori bersifat publik, platform GitHub Classroom dipertimbangkan untuk menggunakan fungsionalitas pengorganisasian yang disajikannya dan juga menawarkan repositori pribadi untuk siswa [17]. GitHub adalah situs web dan layanan yang menggunakan sistem kontrol revisi Git yang memfasilitasi pengembangan proyek perangkat lunak. Jika tidak, GitHub Classroom menyediakan fungsionalitas yang berguna yang berbeda untuk dosen seperti tugas tugas untuk siswa atau kelompok siswa semua dalam layanan hosting yang sama.

2.2. Memasukan data

Dalam makalah ini, data yang dipertimbangkan diambil dari tugas praktis yang diusulkan dalam mata pelajaran yang berbeda dari Gelar Ilmu Komputer di Universitas León. Secara khusus, mata kuliah yang mengikuti kegiatan ini adalah Penyuluhan Sistem Operasi, Pemrograman Komputer I, dan Organisasi Komputer.

Ada dua mata pelajaran untuk mahasiswa tahun pertama: Pemrograman Komputer I dan Organisasi Komputer. Dalam kasus Pemrograman Komputer I, ini berkaitan dengan dasar-dasar pemrograman perangkat lunak, yaitu desain, implementasi, dan pengujian. Oleh karena itu, siswa memperoleh kompetensi merancang pada tingkat abstrak tertentu untuk algoritma dan pola yang berbeda. Selain itu, siswa belajar merancang dan membuat kode kasus uji.

Organisasi Komputer juga merupakan kursus tahun pertama, dan tujuannya terkait dengan dasar-dasar komputer seperti jalur data, unit kontrol, memori, atau input/output. Dalam pelajaran praktis, siswa belajar bagaimana memprogram dalam bahasa C dan assembler.

Akhirnya, Ekstensi Sistem Operasi adalah kursus tahun kedua di mana siswa berurusan dengan sistem operasi. Lebih khusus lagi, siswa memperoleh pengetahuan tentang cara kerja memori utama

(volatile) dan bagaimana sistem file dikelola (non-volatile). Subjek ini juga membahas masalah keamanan dalam sistem operasi.

Alasan memilih mata pelajaran yang disebutkan sebelumnya adalah karena semuanya melibatkan praktik tugas di mana siswa harus mengembangkan program dalam bahasa tertentu. Selain itu, penggunaan Repositori Git adalah wajib untuk menyimpan kode sumber dan dokumentasi. Tabel 1 menunjukkan jumlah sampel untuk setiap mata pelajaran.

Tabel 1. Jumlah sampel dalam dataset.

Subjek	Kursus Tahun	Dukungan Tugas Praktik	
Pemrograman Komputer I	1y 2018–2019 1y	3	372
Organisasi Komputer	2018–2019 2016–	5	240
Ekstensi Sistem Operasi	2y 2017 2017–	1	72
	2018	1	47

Mengenai ekstraksi fitur, dataset dibentuk sebagai berikut. Mengingat tugas praktis yang dikembangkan oleh siswa dan aktivitasnya di repositori, variabel yang berbeda dipertimbangkan:

- Sebuah pengenal untuk membuat siswa anonim (id): memberikan perbedaan antara sampel.
- Penghitung komit (komit): ini mewakili jumlah operasi komit yang dibuat oleh yang diberikan murid.
- Jumlah hari di mana setidaknya satu komit dilakukan (hari).
- Jumlah rata-rata operasi komit per hari (komit/hari).
- Jumlah baris tambahan yang ditambahkan ke kode sumber selama pengembangan penugasan (penambahan).
- Jumlah baris dihapus dihapus dari kode sumber selama pengembangan tugas (penghapusan).
- Jumlah masalah yang dibuka (issues-opened) selama penugasan.
- Jumlah masalah yang ditutup (issues-closed) selama penugasan.

Dalam [15], penulis mengusulkan untuk mengumpulkan beberapa variabel tambahan yang diekstraksi dari sumbernya kode: jumlah total baris kode sumber, jumlah subrutin, jumlah variabel atau objek, jumlah kelas, jumlah loop, kompleksitas siklomatik, jumlah file, beberapa fitur terkait pemrograman berorientasi objek (yaitu, pewarisan dan penggunaan polimorfisme), jumlah file README, dependensi, dll. Mereka juga mempertimbangkan variabel yang dihitung oleh ukuran interaksi dengan VCS. Data ini nantinya dapat digunakan untuk menganalisis kualitas kode sumber dan, dengan demikian, mempelajari evolusi keterampilan pemrograman siswa. Secara khusus, dalam karya ini, variabel tambahan berikut dipertimbangkan: jumlah file kode sumber yang ditulis dalam C (termasuk file header), Java, dan assembler; serta jumlah baris kode sumber di setiap file.

2.3. Desain dan Evaluasi Model

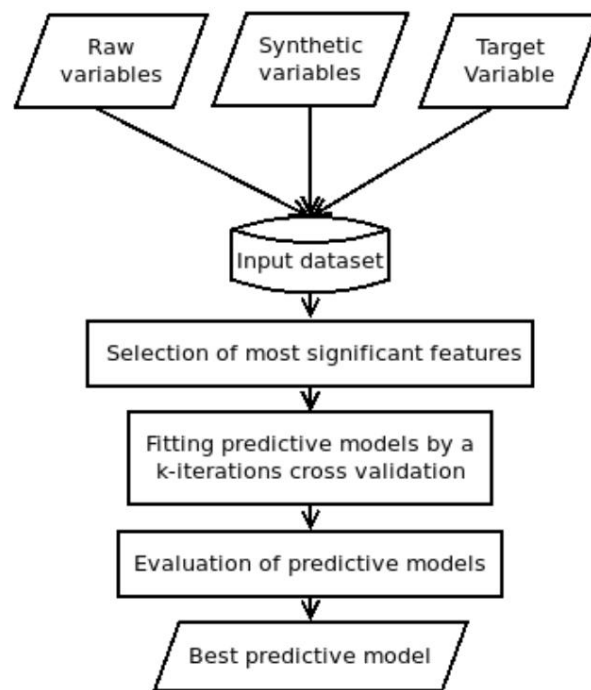
Dalam pekerjaan ini, alat MoEv digunakan untuk mencapai model yang melakukan prediksi terbaik. Metodologi yang diikuti disajikan dalam [4,5]. Gambar 1 menunjukkan skema metodologi yang diterapkan.

Vektor ciri dari setiap sampel dibentuk oleh dua macam variabel. Salah satu jenisnya adalah itu variabel yang diekstraksi dari data mentah yang diberikan oleh SIS atau LMS secara langsung. Dalam karya ini, fitur vektor memiliki elemen data mentah berikut: id, komit, penambahan, penghapusan, masalah dibuka, dan masalah-tertutup. Selain itu, metrik kode sumber juga dianggap sebagai data mentah. Selain mentah data, ada data sintetis yang diwakili oleh variabel berikut: hari (jumlah hari dengan at setidaknya operasi komit) dan komit/hari (operasi komit rata-rata per hari).

Setelah vektor fitur dibangun, variabel target (kelas) dimasukkan. Itu adalah komponen yang menentukan kelas yang dimiliki oleh vektor fitur tertentu. Elemen kelas ini memungkinkan

model untuk melatih dan menguji prosedur pembelajaran terawasi. Target yang dipertimbangkan adalah variabel biner dengan nilai P jika siswa yang diberikan akan lulus kursus dan Nilai F jika siswa yang diberikan akan gagal.

Mengikuti metodologi yang ditunjukkan pada Gambar 1, setelah langkah pengumpulan data, fitur-fitur yang lebih signifikan harus diidentifikasi untuk mendapatkan model klasifikasi yang menilai variabel target. Tahap ini, yang disebut seleksi fitur, adalah prosedur yang menentukan kontribusi setiap fitur, mengidentifikasi pengaruh yang lebih tinggi ketika klasifikasi atau prediksi dilakukan. Dengan melakukan seleksi fitur sebelum membangun model, overfitting berkurang, meningkatkan akurasi dan, oleh karena itu, waktu pelatihan.



Gambar 1. Skema operasi MoEv seperti pada [5].

Menurut skor yang disajikan dalam [14], algoritma tree-base mencapai tingkat yang baik untuk database yang dipertimbangkan dalam makalah ini. Dengan demikian, pemilihan fitur dihitung dengan menggunakan pohon yang sangat acak [18]. Kriteria keputusan untuk memilih fitur didasarkan pada nilai koefisien Gini (G) yang dihitung untuk setiap fitur; jika G lebih tinggi dari ambang batas tertentu, maka fitur tersebut signifikan; dalam makalah ini, nilai ambang batas ditetapkan menjadi 0,1, diperoleh secara empiris.

Setelah pemilihan fitur, beberapa model prediksi dipasang. Kami menggunakan 80% data input untuk melatih model dan 20% untuk mengujinya. Algoritma Machine Learning (ML) berikut dipertimbangkan karena terkenal di bidang ini:

- Peningkatan Adaptif (AB). Metode ensemble adalah teknik yang menggabungkan pengklasifikasi dasar yang berbeda mengubah pelajar yang lemah menjadi metode yang lebih akurat. Boosting adalah salah satu jenis metode ensemble yang paling sukses dan AB salah satu algoritma boosting yang paling populer.
- Pohon Klasifikasi Dan Regresi (CART). Pohon keputusan adalah metode yang memprediksi label yang terkait dengan sebuah instance dengan melakukan perjalanan dari simpul akar pohon ke daun [19]. Ini adalah metode non-parametrik di mana pohon ditanam dalam proses top-down yang berulang.
- K-Nearest Neighbors (KNN). Meskipun konsep tetangga terdekat adalah dasar dari banyak metode pembelajaran lainnya, terutama pembelajaran tanpa pengawasan, pembelajaran berbasis tetangga yang diawasi juga tersedia untuk mengklasifikasikan data dengan label diskrit. Ini adalah teknik non-parametrik yang mengklasifikasikan pengamatan baru berdasarkan jarak ke pengamatan di set pelatihan. Sebuah presentasi yang baik dari analisis diberikan dalam [20,21].

- Analisis Diskriminan Linier (LDA). Ini adalah metode parametrik yang mengasumsikan bahwa distribusi data adalah Gaussian multivariat [21]. Selanjutnya, LDA mengasumsikan pengetahuan tentang parameter populasi. Dalam kasus lain, penduga kemungkinan maksimum dapat digunakan. LDA menggunakan pendekatan Bayesian untuk memilih kategori, yang memaksimalkan probabilitas bersyarat; lihat [22-24].
- Regresi Logistik (LR). Metode linier dimaksudkan untuk regresi di mana nilai target diharapkan menjadi kombinasi linier dari variabel input. LR, terlepas dari namanya, adalah model linier untuk klasifikasi daripada regresi. Dalam model ini, probabilitas yang menggambarkan kemungkinan hasil dari percobaan tunggal dimodelkan menggunakan fungsi logistik.
- Multi-Lapisan Perceptron (MLP). Jaringan saraf tiruan adalah model yang terinspirasi oleh struktur otak. Jaringan saraf digunakan ketika jenis hubungan antara input dan output tidak diketahui. Jaringan diatur ke dalam lapisan node (lapisan input, lapisan output, dan satu atau lebih lapisan tersembunyi). Lapisan-lapisan ini diatur dalam grafik berarah sehingga setiap lapisan terhubung sepenuhnya ke lapisan berikutnya. MLP adalah modifikasi dari perceptron linier standar yang karakteristik terbaiknya adalah mampu membedakan data yang tidak dapat dipisahkan secara linier. MLP menggunakan back-propagation untuk melatih jaringan; lihat [25,26].
- Naive Bayes (NB). Metode ini didasarkan pada penerapan teorema Bayes dengan asumsi "naif" independensi antara setiap pasangan fitur; lihat [21,27].
- Hutan Acak (RF). Ini adalah classifier yang terdiri dari kumpulan pohon keputusan, di mana setiap pohon dibangun dengan menerapkan algoritma ke set pelatihan dan vektor acak tambahan yang diambil sampelnya melalui bootstrap re-sampling [28].

Terakhir, eksperimen dilakukan dengan alat MoEv yang mendukung k-iteration cross-validation sehingga dapat mengidentifikasi algoritma prediksi yang lebih tepat untuk masalah yang diberikan. Untuk melakukan itu, Kementerian Lingkungan Hidup menghitung satu set Key Performance Indicators (KPI) yang terkenal. Setiap kinerja model diukur dengan mempertimbangkan akurasi klasifikasi (lihat Persamaan (1)):

$$\text{akurasi} = \frac{\text{total data} \times (\text{Tp} + \text{Tn})}{\text{total data}} \quad (1)$$

Tp adalah jumlah siswa yang lulus mata kuliah tersebut dan model memprediksi demikian (positif benar) dan Tn jumlah siswa yang tidak lulus mata kuliah dan model memprediksi demikian (negatif benar).

Tiga model dengan skor klasifikasi akurasi tertinggi dipilih sebelumnya untuk evaluasi mendalam dengan mempertimbangkan KPI berikut: Precision (P), Recall (R), dan F1-score yang diperoleh melalui matriks konfusi.

Presisi (P) dihitung sebagai berikut (Persamaan (2)):

$$P = \frac{\text{Tp}}{\text{Tp} + \text{Fp}} \quad (2)$$

Fp adalah banyaknya siswa yang tidak lulus mata kuliah tersebut dan model menyatakan lulus (positif palsu).

Recall (R) didefinisikan dalam Persamaan (3), di mana Fn adalah jumlah siswa yang lulus mata kuliah dan model mengatakan mereka gagal (negatif palsu):

$$R = \frac{\text{Tp}}{\text{Tp} + \text{Fn}} \quad (3)$$

Skor F1 menghubungkan metrik tersebut karena merupakan rata-rata harmonik presisi dan recall, seperti yang ditunjukkan pada Persamaan (4).

$$F1 = 2 \times \frac{P \times R}{P + R} \quad (4)$$

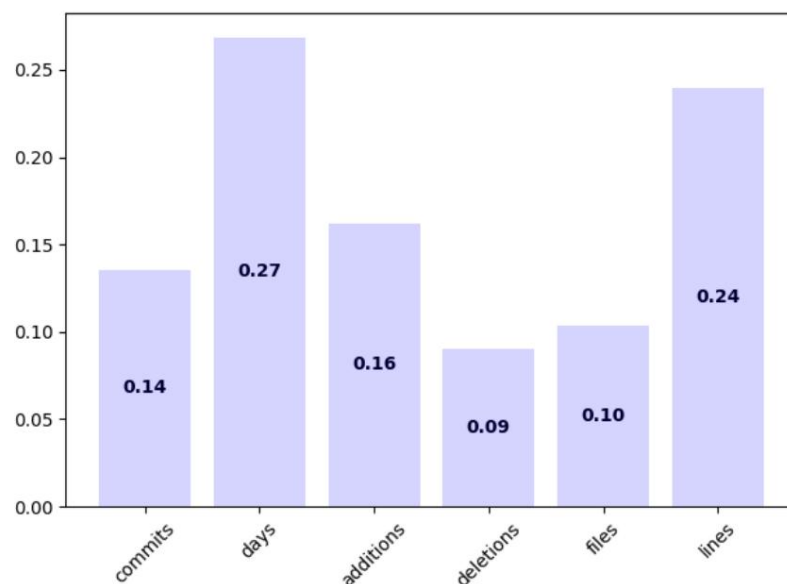
3. Hasil

Kepentingan fitur dihitung dengan koefisien Gini, seperti yang ditunjukkan pada Gambar 2. Fitur dengan koefisien Gini yang rendah dibuang ($G < 0,1$). Fitur yang dipilih adalah hari ($G = 0,27$), garis ($G = 0,24$), penambahan ($G = 0,16$), komit ($G = 0,14$), dan file ($G = 0,1$).

Tabel 2 mengumpulkan akurasi yang dihitung dengan menggunakan eksekusi MoEv 10 iterasi. Seperti yang ditunjukkan pada tabel, hasil terbaik diperoleh dengan RF (0,78), KNN (0,72), dan AB (0,7).

Pada Gambar 3, matriks kebingungan untuk model RF, KNN, dan AB disajikan.

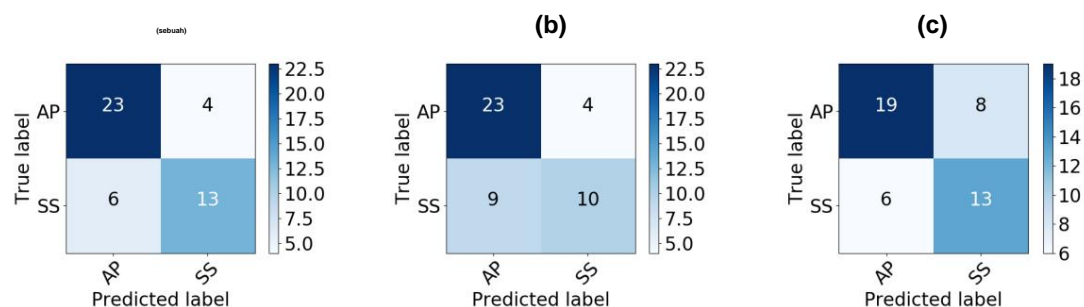
Precision (P), Recall (R), dan F1-score yang diperoleh melalui confusion matrix dari Gambar 3 adalah :
dikumpulkan pada Tabel 3.



Gambar 2. Pentingnya fitur.

Tabel 2. Skor klasifikasi akurasi.

	Penggolong						
	RF	KNN	AB	LR	LDA	CART	NB MLP
Skor	0.78	0.72	0.70	0.67	0.67	0.67	0.63 0.63



Gambar 3. Confusion matrix untuk model RF (a), KNN (b), dan AB (c) .

Tabel 3. Precision, recall dan F1 -score untuk model RF, KNN, dan AB.

Penggolong	Kelas	Dukungan Skor F1 PR		
RF	P	0,79 0,85	0,82	27
	F	0,76 0,68	0,82	19
	rata-rata/total	0,78 0,78	0,78	46
KNN	P	0,72 0,85	0,78	27
	F	0,71 0,53	0,61	19
	rata-rata/total	0,72 0,72	0,71	46
AB	P	0,76 0,70	0,73	27
	F	0,62 0,68	0,65	19
	rata-rata/total	0,70 0,70	0,79	46

4. Diskusi

Pentingnya fitur pada Gambar 2 menunjukkan bahwa jumlah hari dengan setidaknya satu operasi komit (hari) adalah fitur yang paling diskriminan ($G = 0,27$). Ini logis karena bekerja dalam solusi perangkat lunak biasanya menyiratkan fase dan tonggak yang berbeda. Nilai yang rendah dalam fitur ini menunjukkan bahwa siswa belum mencapai milestone yang diharapkan. Selain itu, jumlah hari dengan setidaknya satu operasi komit memiliki kepentingan yang lebih tinggi daripada jumlah operasi komit itu sendiri; ini harus dilakukan dengan fakta bahwa beberapa tonggak biasanya tidak dapat dicapai pada hari yang sama. Memiliki banyak operasi komit pada hari yang sama biasanya berarti bahwa siswa menggunakan beberapa operasi komit untuk satu tonggak sejarah. Ini mungkin terjadi jika ada kesalahan tak terduga karena perangkat lunak yang belum diuji atau bahkan bahwa siswa mencoba untuk mencontek model.

Fitur pembeda berikutnya adalah jumlah baris kode sumber ($G = 0,24$). Ini memungkinkan kami untuk menegaskan bahwa isi repositori mungkin cukup berguna untuk memprediksi keberhasilan akademis. Termasuk lebih banyak fitur tambahan seperti yang diusulkan di [15] selain jumlah file dan baris mungkin menghasilkan akurasi yang lebih tinggi. Namun, penting untuk menunjukkan bahwa ini memerlukan analisis statis dari Kode sumber. Analisis ini berbeda untuk setiap bahasa pemrograman, sehingga sangat rumit pengumpulan data, meskipun paradigma pemrograman cukup umum antar bahasa. Ini adalah sebuah kelemahan penting karena pengumpulan data sebenarnya cukup rumit dan memakan waktu. Namun, jumlah bahasa pemrograman yang digunakan selama gelar Ilmu Komputer tidak begitu tinggi, sehingga metode yang disajikan dalam makalah ini dapat dengan mudah disesuaikan dengan mata pelajaran lain.

Mengenai akurasi model, RF, KNN, dan AB memiliki skor tertinggi, seperti yang ditunjukkan pada Tabel 2. RF juga merupakan model dengan skor akurasi tertinggi pada [5]. Tabel 4 menunjukkan perbandingan lengkap antara skor akurasi yang diperoleh dalam pekerjaan ini ditunjukkan pada Tabel 2 dan skor akurasi yang diperoleh di [5]. Baris pertama pada Tabel 4 menunjukkan skor akurasi yang diperoleh dalam pekerjaan ini. Baris kedua menunjukkan skor akurasi yang diperoleh pada [5]. Penting untuk menunjukkan bahwa semua model meningkatkan skor akurasi. Hal ini disebabkan fakta bahwa dalam pekerjaan ini, dataset pelatihan jauh lebih besar dan lengkap (karena menyertakan data dari beberapa mata pelajaran) dibandingkan dengan dataset yang digunakan pada [5]. Sebagai tambahan, mempertimbangkan fitur tambahan yang signifikan juga menyiratkan peningkatan akurasi.

Tabel 4. Perbandingan skor akurasi antara model yang diusulkan dalam karya ini (baris pertama) dan model yang diusulkan dalam [5] (baris kedua).

	Penggolong					
	RF	KNN	AB	LR	LDA	CART NB MLP
Skor model yang diusulkan pada pekerjaan ini 0,78 Skor	0,72	0,70	0,67	0,67	0,67	0,63 0,63
model yang diusulkan dalam [5] 0,70	0,60	0,40	0,60	0,60	0,60	0,50 0,40

Akhirnya, dengan mempertimbangkan model yang paling akurat, yaitu, RF, KNN, dan AB, analisis yang lebih dalam melalui matriks kebingungan mereka diberikan untuk menganalisis sensitivitas mereka untuk mendeteksi Ps: yaitu, tingkat true

Ps bahwa model diklasifikasikan salah. Gambar 3 dan Tabel 3 menunjukkan bahwa pengklasifikasi RF memperoleh nilai presisi (P), recall (R), dan skor F1 yang lebih baik daripada KNN dan AB baik pada tahap pengujian maupun validasi.

5. Kesimpulan

Metodologi yang disajikan dalam makalah ini memungkinkan dosen untuk memprediksi hasil yang dicapai siswa dengan mengawasi kinerja harian mereka menggunakan VCS. Model yang digunakan dipasang dengan menggunakan kumpulan data yang mengumpulkan data dari berbagai mata pelajaran dari gelar Ilmu Komputer di Universitas León. Semua mata pelajaran yang dipertimbangkan memiliki dua item umum: semuanya mengharuskan siswa untuk mengembangkan solusi perangkat lunak dan menggunakan VCS untuk menyimpan kode sumber dan dokumentasi.

Dataset termasuk variabel dari interaksi siswa dengan VCS seperti jumlah operasi komit, baris kode sumber ditambahkan dan dihapus, dll. Selain itu, dataset termasuk data yang terkait dengan isi repositori siswa seperti jumlah file dan baris kode sumber.

Variabel tambahan ini, meskipun tidak berasal dari interaksi dengan VCS, adalah fitur kode sumber; dengan demikian, model memastikan generalisasi yang optimal.

Hasil yang disajikan di Bagian 3 dan dibahas di Bagian 4 menunjukkan bahwa skor akurasi model meningkat dengan menggunakan kumpulan data yang lebih besar, tidak hanya menyertakan variabel tambahan, tetapi juga lebih banyak sampel. Akurasinya juga mirip dengan hasil yang diperoleh dengan model yang dilatih dengan kumpulan data yang berisi fitur spesifik, dan dengan demikian, dan bertentangan dengan model ini, model yang diusulkan dalam makalah ini memastikan generalisasi yang optimal.

Pekerjaan lebih lanjut harus menghadapi penggunaan fitur tambahan yang dikumpulkan dalam kode sumber itu sendiri untuk menganalisis kualitas kode sumber dan, dengan demikian, mempelajari evolusi keterampilan pemrograman siswa.

Kontribusi Penulis: Konseptualisasi, .MG-H.; metodologi, .MG-H., CFL, LSG, dan M.Á.CG; perangkat lunak, .MG-H., AGF, dan GEC; validasi, CFL, LSG, dan M.Á.CG; investigasi, .MG-H., CFL, dan LSG; sumber daya, CFL; kurasi data, AGF, dan GEC; menulis—persiapan draf asli, .MG-H. dan LSG; menulis—meninjau dan mengedit, CFL, AGF, GEC, dan M.Á.CG; visualisasi, .MG-H.; pengawasan, M.Á.CG; administrasi proyek, .MG-H.; akuisisi pendanaan, CFL Semua penulis membaca dan menyetujui versi naskah yang diterbitkan.

Pendanaan: Penelitian yang dijelaskan dalam artikel ini sebagian didanai oleh Support Plan for Teaching Innovation Groups (PAGID) dari University of León 2018 dan oleh Komisi Eropa di bawah Grant 2018-1-ES01-KA201-050939: ROBOSTEAM—Integrating Steam and Pengembangan Computational Thinking dengan menggunakan Robotika dan Perangkat Fisik.

Konflik Kepentingan: Para penulis menyatakan tidak ada konflik kepentingan. Para penyandang dana tidak memiliki peran dalam desain penelitian; dalam pengumpulan, analisis, atau interpretasi data; dalam penulisan naskah; maupun dalam keputusan untuk mempublikasikan hasilnya.

Singkatan

Daftar singkatan yang digunakan dalam makalah ini:

AB	Peningkatan Adaptif
Klasifikasi CART Dan Pohon Regresi	
TIK	Teknologi Informasi dan Komunikasi
KNN K-Tetangga Terdekat	
KPI	Indikator kinerja utama
LA	Analisis Pembelajaran
Analisis Diskriminan Linier LDA	
Sistem Manajemen Pembelajaran LMS	
LR	Regresi logistik
Pembelajaran Mesin ML	
Perceptron Multi-Lapisan MLP	
Evaluator Modul MoEv	

NB Naive Bayes
 Hutan Acak RF
 SIS Sistem Kelembagaan Mahasiswa
 Sistem Kontrol Versi VCS

Referensi

1. Siemens, G.; Gasevic, D. Editorial tamu-Learning and knowledge analytics. Pendidikan teknologi. Soc. **2012**, *15*, 1-2.
2. Siemens, G.; Dawson, S.; Lynch, G. Meningkatkan Kualitas dan Produktivitas Sektor Pendidikan Tinggi: Kebijakan dan Strategi Penerapan Learning Analytics di Tingkat Sistem; Society for Learning Analytics Research untuk Australian Office for Learning and Teaching: Canberra, Australia, 2013.
3. Gašević, D.; Dawson, S.; Rogers, T.; Gasevic, D. Analisis pembelajaran tidak boleh mempromosikan satu ukuran cocok untuk semua: Efek dari kondisi instruksional dalam memprediksi keberhasilan akademik. Internet Tinggi. Pendidikan **2016**, *28*, 68–84. [\[CrossRef\]](#)
4. Guerrero-Higueras, .M.; DeCastro-García, N.; Matellán, V.; Conde, M.. Model prediktif keberhasilan akademik: studi kasus dengan sistem kontrol versi. Dalam Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spanyol, 24-26 Oktober 2018; hal.306–312.
5. Guerrero-Higueras, .M.; DeCastro-García, N.; Rodriguez-Lera, FJ; Matellán, V.; Conde, M.. Memprediksi keberhasilan akademik melalui interaksi siswa dengan Sistem Kontrol Versi. Buka Komputasi. Sci. **2019**, *9*, 243–251. [\[CrossRef\]](#)
6. Guerrero-Higueras, .M.; DeCastro-García, N.; Matellán, V. Deteksi serangan Cyber ke sistem lokalisasi waktu nyata dalam ruangan untuk robot otonom. Robot. Oton. Sistem **2018**, *99*, 75–83. [\[CrossRef\]](#)
7. Kovacic, Z. Memprediksi keberhasilan siswa dengan menambang data pendaftaran. Res. Tinggi. Pendidikan J. **2012**, *15*, 1–20.
8. Agudo-Peregrina, .F.; Iglesias-Pradas, S.; Conde-González, M.Á.; Hernández-García, . Bisakah kita memprediksi kesuksesan dari data log di VLE? Klasifikasi interaksi untuk analitik pembelajaran dan hubungannya dengan kinerja dalam F2F dan pembelajaran online yang didukung VLE. Hitung. Bersenandung. Perilaku **2014**, *31*, 542–550. [\[CrossRef\]](#)
9. Tukang Cukur, R.; Sharkey, M. Koreksi kursus: Menggunakan analitik untuk memprediksi keberhasilan kursus. Dalam Prosiding Konferensi Internasional ke-2 tentang Analisis dan Pengetahuan Pembelajaran, Vancouver, BC, Kanada, 29 April–2 Mei 2012; hal.259–262.
10. Spinellis, D. Sistem kontrol versi. IEEE Softw. **2005**, *22*, 108–109. [\[CrossRef\]](#)
11. Fischer, M.; Pinzger, M.; Gall, H. Mengisi database riwayat rilis dari kontrol versi dan sistem pelacakan bug. Dalam Prosiding Konferensi Internasional tentang Pemeliharaan Perangkat Lunak, Amsterdam, Belanda, 22–26 September 2003; hal.23–32.
12. Pilato, CM; Collins-Sussman, B.; Fitzpatrick, Kontrol Versi BW dengan Subversion: Next Generation Open Kontrol Versi Sumber; O'Reilly Media, Inc.: Sebastopol, CA, AS, 2008.
13. Torvalds, L.; Hamano, J. Git: Sistem Kontrol Versi Cepat. Tersedia online: <http://git-scm.com> (diakses pada 21 Februari 2020).
14. Guerrero-Higueras, AM; Matellán-Olivera, V.; Esteban-Costales, G.; Fernández-Llamas, C.; Rodríguez-Sedano, FJ; ngel, CM Model untuk mengevaluasi kinerja siswa melalui interaksi mereka dengan sistem kontrol versi. Dalam Prosiding Learning Analytics Summer Institute (LASI), New York, NY, AS, 11–13 Juni 2018.
15. Guerrero-Higueras, .M.; Sánchez-González, L.; Fernández-Llamas, C.; Conde, M.Á.; Lera, FJR; Sedano, FJR; Costales, GE; Matellán, V. Prediksi keberhasilan akademik melalui interaksi dengan sistem kontrol versi. Dalam Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spanyol, 24-26 Oktober 2018; hal. 284–289.
16. De Alwis, B.; Sillito, J. Mengapa proyek perangkat lunak berpindah dari sistem kontrol versi terpusat ke sistem kontrol versi terdesentralisasi ? Dalam Prosiding Lokakarya ICSE 2009 tentang Aspek Koperasi dan Manusia pada Rekayasa Perangkat Lunak , Vancouver, BC, Kanada, 17 Mei 2009; hal.36–39.
17. Griffin, T.; Seals, S. Github di dalam kelas: Tidak hanya untuk proyek kelompok. J. Hitung. Sci. Kol. **2013**, *28*, 74-74.
18. Geurts, P.; Ernst, D.; Wehenkel, L. Pohon yang sangat acak. Mach. Mempelajari. **2006**, *63*, 3-42. [\[CrossRef\]](#)
19. Friedman, J.; Tergesa-gesa, T.; Tibshirani, R. Unsur-unsur Pembelajaran Statistik, edisi ke-2; Musim Semi: Berlin/Heidelberg, Jerman, 2009; Jilid 1.
20. Devroye, L.; Györfi, L.; Lugosi, G. Teori Probabilistik Pengenalan Pola; Musim Semi: Berlin/Heidelberg, Jerman, 2013; Jilid 31.

21. Duda, RO; Hart, PE; Bangau, Klasifikasi Pola Ditjen; John Wiley & Sons: Hoboken, NJ, AS, 2012.
22. Bishop, Pengenalan pola CM. Mach. Mempelajari. **2006**, 128, 1–58.
23. Koller, D.; Friedman, N. Model Grafis Probabilistik: Prinsip dan Teknik; MIT Pers: Cambridge, MA, Amerika Serikat, 2009.
24. Murphy, Pembelajaran Mesin KP: Perspektif Probabilistik; MIT Press: Cambridge, MA, AS, 2012.
25. Rummelhart, DE Belajar representasi internal dengan propagasi kesalahan. Dalam Pemrosesan Terdistribusi Paralel; 1986.
26. Cybenko, G. Pendekatan dengan superposisi fungsi sigmoidal. Matematika. Kontrol. Sistem Sinyal **1989**, 2, 303–314. [\[CrossRef\]](#)
27. Zhang, H. Optimalitas dari Naive Bayes. AA **2004**, 1, 3.
28. Breiman, L. Hutan acak. Mach. Mempelajari. **2001**, 45, 5-32. [\[CrossRef\]](#)



© 2020 oleh penulis. Penerima Lisensi MDPI, Basel, Swiss. Artikel ini adalah artikel akses terbuka yang didistribusikan di bawah syarat dan ketentuan Creative Commons Attribution

(CC BY) lisensi (<http://creativecommons.org/licenses/by/4.0/>).

Tugas 2 Kakas Perangkat Lunak

Tugas Individu. Dikumpulkan paling lambat tanggal 17 Oktober 2022 jam 20.00 WITA.
Dikumpulkan dalam bentuk softcopy. File yang dikumpul jurnal utama yang di jadikan reverensi utama untuk di review dan format laporan ini dalam satu FILE.

1. FILE jurnal Asli + Jurnal Yang sudah di Translate + Format Tugas dibawah ini dalam satu file (dikirim ke link yang telah ditentukan).

Catatan (Lewat dari batas pengumpulan nilai tugas 1=0)

1. Cari sebuah paper yang terkait dengan Version Control System (*International*).
2. Tahun Jurnal yang boleh digunakankan 5 Tahun Terakhir.
3. Baca dan buat analisisnya
4. Buat laporan sesuai format di bawah ini.

Nama dan NIM : Jusni (60900120021)

1.

Judul paper: Academic Success Assessment through Version Control Systems (Penilaian Keberhasilan Akademik melalui Versi Sistem control)

Dipublikasikan di (Tempat Publikasi) : applied science (Robotics group, dept. of Mechanical, Computer Science, and Aerospace Engineering, University of León, Campus de Vegazana)

A. Deskripsikan secara singkat paper yang anda baca

Paper ini berisi tentang penggunaan version control system yang sangat diperlukan bagi para professional teknologi informasi dan komunikasi. Dimana ini banyak digunakan oleh dosen pada saat penugasan praktek. Hal ini dilakukan untuk memperkirakan prestasi akademik siswa dengan mengawasi kinerja harian mereka dengan VCS. Untuk itu, ada dua masalah yang harus dihadapi. Yang pertama adalah jika keberhasilan akademik terkait dengan variabel yang dipertimbangkan yang dihitung dari interaksi siswa dengan VCS. Kedua, jika memungkinkan untuk merancang model yang menganalisis fitur-fitur tersebut untuk memperkirakan keberhasilan siswa secara otomatis.

B. Beri penilaian

1. Apakah paper tersebut dipresentasikan dan dituliskan dengan jelas, mudah dibaca dan memiliki struktur penulisan yang runut ☒ Yes ☐ No
2. Apakah mengandung materi yang tidak relevan di dalamnya? ☐ Yes ☒ No
3. Apakah judulnya sesuai menggambarkan isi paper? ☒ Yes ☐ No
4. Apakah abstrak telah menggambarkan tujuan penelitian, menyebut metode yang digunakan dan menunjukkan hasil yang diperoleh ☒ Yes ☐ No
5. Apakah bagian pendahuluan telah memuat dengan jelas masalah yang akan dipecahkan? ☒ Yes ☐ No

6. Apakah bagian pendahuluan telah memuat hasil-hasil penelitian sebelumnya yang relevan dan up to date ? ☒ Yes ☐ No
7. Apakah metode telah dijelaskan dengan rinci sehingga pembaca/peneliti lain bisa mengulangnya kembali? ☒ Yes ☐ No
8. Apakah pada bagian diskusi telah menyertakan penelitian-penelitian terkait? ☒ Yes ☐ No
7. Apakah refensi-referensi yang diacu relevan, up to date dan dapat diperoleh dengan mudah? ☒ Yes ☐ No
8. Apakah ilustrasi tabel dan gambar yang dicantumkan memang tepat dibutuhkan dan tidak ada pengulangan informasi dalam tabel dan gambar ? ☒ Yes ☐ No
9. Apakah keterangan pada caption gambar dan tabel telah lengkap dan sesuai menggambarkan isi gambar dan tabel ? ☒ Yes ☐ No
10. Apakah kesimpulan yang dibuat telah tepat, logis/masuk akal, dan dapat dipertanggungjawabkan sesuai dengan hasil pengujian/pengamatan/studi yang dilakukan ☒ Yes ☐ No
11. Apakah anda setuju dengan klaim dari penulis tentang kebaruan atau kenotribusi dari paper tersebut? ☒ Yes ☐ No

Beri alasannya kenapa anda setuju atau tidak

Alasan saya setuju dengan adanya klaim kebaruan untuk paper ini karena pembahasan dari paper ini tentang teknologi dan teknologi itu terbarukan sehingga pastinya penelitian ini butuh pembaruan dalam penelitian yang akan datang.

12. Apakah paper tersebut mengandung ide yang orisinal ☒ Yes ☐ No
Beri komentar anda.

Yah, walaupun mungkin ide ini sudah ada sebelumnya tapi keoriginalan ide dilihat Ketika ide tersebut sudah di eksekusi oleh peneliti dan menghasilkan sebuah hasil.

13. Apakah menurut anda ada materi dalam paper ini yang seharusnya dihilangkan? ☐ Yes ☒ No

Jika ya, apa?
.....

14. Dampak ilmiahnya ☐ Tinggi ☒ Sedang ☐ tidak ada
15. Keterkaitan antara judul, abstrak, masalah, hasil, dan kesimpulan ☒ Sangat terkait ☐ Cukup ☐ Kurang
16. Manfaat bagi komunitas/masyarakat ☐ Tinggi ☒ Sedang ☐ Rendah

C. Tuliskan catatan, komentar, saran anda mengenai paper ini dari mulai cara penulisan, kejelasan pendefinisian masalah, kesesuaian dan ketepatan metode yang digunakan, penyajian hasil, kedalaman analisis hasil dalam diskusi, penarikan kesimpulan, dan lain-lain

1. Cara penulisan, tata bahasa , dll

Komentar:

Dari cara penulisan dan tata Bahasa dua jurnal diatas, cukup tersusun dengan baik dan jelas. Mulai dari abstrak, pendahuluan, penulis, metode. Pembahasan, hasil, kesimpulan serta daftar pustakanya

2. Pendefinisian masalah

Komentar:

Penggunaan sistem kontrol versi merupakan keterampilan yang sangat dibutuhkan oleh para profesional teknologi informasi dan komunikasi. Dengan demikian, penggunaannya harus didorong oleh lembaga pendidikan. Karya ini menunjukkan bahwa adalah mungkin untuk menilai apakah seorang siswa dapat lulus mata pelajaran yang berhubungan dengan ilmu komputer dengan memantau interaksinya dengan sistem kontrol versi.

3. Metode yang digunakan

Komentar:

Metode atau Langkah-langkah yang digunakan

- System control versi
- Memasukkan data
- Desain dan Evaluasi Model

4. Penyajian hasil

Komentar:

Penyajian hasil yang di implementasikan sangat rinci karena disertakan dengan diagram hasil serta menggunakan rumus-rumus yang sesuai serta untuk meyakinkan pembaca peneliti menambahkan table hasil.

5. Kedalaman analisis dan diskusi

Komentar:

Kedalaman Analisis dan diskusi sangat rinci dikarenakan pada bagian ini dijelaskan fitur yang dihasilkan dari masalah. Serta analisis yang digunakan pada diskusi ini sangat berbeda untuk setiap pemrogramannya sehingga sangat rumit pengumpulan datanya.

6. Penarikan kesimpulan

Komentar:

Metodologi yang disajikan dalam makalah ini memungkinkan dosen untuk memprediksi hasil yang dicapai siswa dengan mengawasi kinerja harian mereka menggunakan VCS. Model yang digunakan dipasang dengan menggunakan kumpulan data yang mengumpulkan data dari berbagai mata pelajaran dari gelar Ilmu Komputer di Universitas León. Semua mata pelajaran yang dipertimbangkan memiliki dua item umum: semuanya mengharuskan siswa untuk mengembangkan solusi perangkat lunak dan menggunakan VCS untuk menyimpan kode sumber dan dokumentasi.

Hasil yang disajikan di Bagian 3 dan dibahas di Bagian 4 menunjukkan bahwa skor akurasi model meningkat dengan menggunakan kumpulan data yang lebih besar, tidak hanya menyertakan variabel tambahan, tetapi juga lebih banyak sampel. Akurasinya juga mirip dengan hasil yang diperoleh dengan model yang dilatih dengan kumpulan data yang berisi fitur spesifik, dan dengan

demikian, dan bertentangan dengan model ini, model yang diusulkan dalam makalah ini memastikan generalisasi yang optimal.

- D. Tuliskan ide kelompok anda peluang-peluang yang mungkin untuk melanjutkan atau memperbaiki paper ini (minimal 3) **TIDAK USAH DIKERJA****

Selamat mengerjakan
Firmansyah Ibrahim
Program Studi Sistem Informasi
Fakultas Sain dan Teknologi