Journal of Infrastructure, Policy and Development 2024, 8(13), 4216.
https://doi.org/10.24294/jipd4216

Article

Development of a recommendation engine for university study programme selection: A regression-based approach

Aleksei Iurasov, Olga Iurasova, Robert Leščinskij*

Vilnius Gediminas Technical University, Vilnius 10223, Lithuania
*** Corresponding author:** Robert Leščinskij, Robert.lescinskij@vilniustech.lt

CITATION

Iurasov A, Iurasova O, Leščinskij R. (2024). Development of a recommendation engine for university study programme selection: A regression-based approach. Journal of Infrastructure, Policy and Development. 8(13): 4216. https://doi.org/10.24294/jipd4216

ARTICLE INFO

Received: 14 January 2014
Accepted: 19 September 2014
Available online: 8 November 2024

Abstract: This paper is the third in a series focused on bridging the gap between secondary and higher education. Our primary objective is to develop a robust theoretical framework for an innovative e-business model called the Undergraduate Study Programme Search System (USPSS). This system considers multiple criteria to reduce the likelihood of exam failure or the need for multiple retakes, while maximizing the chances of successful program completion. Testing of the proposed algorithm demonstrated that the Stochastic Gradient Boosted Regression Trees method outperforms the current method used in Lithuania for admitting applicants to 47 educational programs. Specifically, it is more accurate than the Probabilistic Neural Network for 25 programs, the Ensemble of Regression Trees for 24 programs, the Single Regression Tree for 18 programs, the Random Forest Regression for 16 programs, the Bayesian Additive Regression Trees for 13 programs, and the Regression by Discretization for 10 programs.

Keywords: educational data analytics; educational data mining; post-secondary education; education market innovation

1. Introduction

In an era of globalization and boundless educational choices, the selection of an appropriate undergraduate study program has become a critical milestone for school graduates worldwide. With an overwhelming 30,586 universities (according to the Webometrics Ranking of World Universities (2021), Countries Arranged by Number of Universities in Top Ranks,) offering a staggering number of study programs, the implications of a misinformed decision can be detrimental, hindering productivity, efficiency, and ultimately, the achievement of life goals (Kazi and Akhlaq, 2017). As the authors confront this challenge, the paper embarks on a transformative journey to introduce the theoretical foundation of an e-service model-the Undergraduate Study Programme Search System (USPSS).

This is the third article in a series building upon the success of initial research that:

- uncovered a significant pairwise correlation between school graduates' characteristics and their tertiary education outcomes (Iurasov, 2022). Armed with this empirical insight, our hypothesis posits that by leveraging advanced Data Science forecasting methods, USPSS can surpass traditional pairwise correlations, elevating the accuracy of its predictions. The integration of individualized forecasts into the USPSS e-business model is destined to revolutionize the education landscape by matching school graduates with study

programmes to foster improved learning outcomes, student satisfaction, and future career prospects;

- defined the USPSS e-business model: essential principles, stakeholders, competitors and competitive advantages, services that will shape the USPSS and sources of its income, as well as the equation for tailoring undergraduate study programme recommendations to school graduates. At its core, USPSS will operate as a Recommendation Engine, skillfully assessing diverse parameters to tailor undergraduate study programme recommendations to individual school graduates. These parameters encompass a comprehensive array of factors, including admission likelihood, dropout potential, course completion difficulties, employment prospects, and university rankings, among others (Iurasov and Iurasov, 2022);
- optimized LAMA BPO algorithm-the Lithuanian national methodology for adjusting different types of school grades into a single scale. This result is a necessary pre-condition for building highly accurate predictive models of USPSS Recommendation Engine (Iurasov, 2022).

The paper now focuses on the pivotal issue of crafting the parameter calculation algorithm for the Recommendation Engine.

To achieve this ambitious goal, the current research spans an extensive range of Data Science methodologies:

1) Classification methods pave the way for forecasting discrete events, such as dropout and student's decision to change study programme (limited number of values ('Yes' or 'No');
2) regression methods illuminate the path to predict continuous targets, like the average university grade;
3) anomaly detection techniques detect deviations from typical admission profiles (limited number of values: "too weak for admission", "student characteristics look the same as those of the students who studied before", "too good to choose this programme"), yielding crucial insights for optimal study programme recommendations.

Given the comprehensive scope of Data Science methods and algorithms for USPSS development (note that additional algorithms will be addressed in subsequent articles of this series), our current article focuses on establishing the theoretical foundation for the Recommendation Engine. The objective is to provide high school graduates with the most fitting study programs, thereby enhancing their university learning outcomes, with a particular emphasis on regression-type methods:

1) The average university grade. The parameter helps the USPSS to predict how successfully a would-be student will complete a particular undergraduate study programme, i.e., recommendation for the educational product most likely to be consumed successfully is formulated.
2) The number of failures in completing course units when the grade of course units is less than 5 of 10 (in a 10-point system). It helps to assess the difficulty of studying, i.e., recommendation for the educational product most likely to be consumed without serious problems is proposed.
3) The number of retakes to complete course units (the same reasoning as in the case of the number of failures).

Research objectives are as follows:

1) to review the available literature on the application of Recommendation Engines in marketing university study programmes and Data Science methods for predicting university learning results;
2) to create the theoretical basis of the Recommendation Engine, manage to recommend study programmes based on several criteria reducing the likelihood of the failed exams or retaking them multiple times and to increase the likelihood of successful completion;
3) to implement the Recommendation Engine based on data from Lithuanian universities (Vilnius Gediminas Technical University, VGTU, 2,1426 student records; Lithuanian Sports University, LSU, 1674 student records) and from school electronic diary Manodienynas (serves 597 Lithuanian educational institutions);
4) to test and analyse algorithm performance metrics of the Recommendation Engine;
5) to suggest directions for further research.

2. Literature review

The 'Netflix Prize' science challenge, which took place from 2 October 2006 to 21 September 2009, had a profound impact on the field of Recommendation Engine development. Following the success of the competition, where the Collaborative Filtering technique was prominently employed by the winning team (Bell Kor's Pragmatic Chaos), this algorithm gained significant popularity within the data science community (Hahsler et al., 2015). Subsequently, numerous authors adopted it as a standard Recommendation Engine technique or utilized it in conjunction with other methods (Arora et al., 2014; Girase et al., 2017; Huynh et al., 2018; Meenakshi and Satpal, 2019; Rivera et al., 2018).

Collaborative filtering, a predictive technique, assumes commonalities in personal interests and analyzes the preferences of other respondents with similar characteristics. While Collaborative Filtering was notably applied in the 'Netflix Prize' based on user movie ratings, researchers have explored the development of other rating-based Recommendation Engines. This approach necessitates a distinct set of methods and data structures compared to our research. For instance, Bokde et al. (2015) aimed to assist students seeking admission to an Engineering College, focusing on ranking criteria such as infrastructure facility, teachers, placements, admission difficulty, and campus life–aligning with the goals of our current research.

Similarly, Sneha et al. (2016) designed a Recommendation Engine to aid in selecting a Master's degree study program. They measured similarity between the preferences of prospective students (collected as rating criteria) and ratings given by previous students to the university. Their ranking criteria, including specialization, financial budget, and interest, complement those used by Bokde et al. (2015). However, it's worth noting that our current paper does not rely on past ratings given by students. The data structure employed in our research is entirely different, relying on demographic data and school grades.

The use of ratings is understandable for the movie Recommendation Engine: researchers simply do not have other relevant information. However, a student response to university ratings is affected by multiple cultural and behavioural factors such as

1) the presence of courtesy bias when the student feels a desire to be polite towards the Alma Mater;
2) biases toward yea or nay-saying (extreme response style);
3) the presence of social desirability bias when students may not respond truthfully but simply provide answers that make them look good, etc.

Student limited outlook is another drawback, because for ranking a university, students do not have a comparison base. Highly ranked universities (from the QS World University Ranking) may receive a moderate grade while an unknown provincial university can achieve high ranking scores. The aforementioned drawbacks have led to extraneous variations in rating scale scores and compromised the validity of the obtained results.

In the case of university education, a large amount of unbiased and exact information like tuition fees, learning results (course unit grades, number of failures in completing course units, number of retakes to complete course units, etc.) is available. Therefore, our focus is on the studies aimed at predicting university learning outcomes.

Given the intricate nature of forecasting, numerous studies delve into the peculiarities of employing Data Science methods to predict university learning outcomes. Existing literature can be systematically analyzed and categorized based on the research aim, object (target variable), methods employed, and accuracy metrics used for forecasting. This analytical approach provides valuable context and emphasizes the relevance of Data Science methods and accuracy metrics to the present research.

Furthermore, considerations such as the field of study (e.g., art, engineering, business, and sport), dataset size, and independent variables utilized in forecasting models are crucial. These factors influence the applicability of approaches in the current study. It's worth noting that while not every scientific article contains a comprehensive set of such information, the majority provide a substantial amount.

Hanandeh et al. (2020) categorized predicted learning results (expected university course unit grades) into five groups: 'Excellent', 'Very good', 'Good', 'Pass', and 'Fail' using classification forecasting methods, including the Decision Tree (J48 algorithm) and Naïve Bayes. Notably, they achieved higher forecasting accuracy (46.8%) with the decision tree method. Since the aim of their study, advising students to choose a university and study program to attain high grades, aligns partly with the current research, it's essential to consider its specifics. However, the applicability of their study is constrained by a misalignment between its aim and the data structure used for developing the forecasting model. The model, built on enrolment data from eight universities in Jordan, incorporated information such as the number of credit hours completed by the student. During data pre-processing, rows with zero credit hours were excluded, rendering the forecasting model intolerant to high school graduates. This limitation prevents the utilization of the model for advising high school graduates on university and study program choices for achieving high grades. The

model's requirement for data about enrolled students impedes its use for prospective high school graduates. Omitting this information hampers the model's functionality. However, the presence of such data signals that the student has already been admitted and is actively studying at the university, indicating a transition from high school graduate to university student.

Usman et al. (2020) emphasized the importance of feature selection methods for improving the accuracy of predicting learning results. They applied Naïve Bayes classifiers for forecasting modelling based on a dataset of 543 students from the Department of Statistics, University of Abuja. In the present study, they used data on school grades as independent variables covering five course units: English, Mathematics, Physics, Chemistry and Biology. Two feature selection methods, embracing correlation-based feature selection and Backward feature elimination were tested for higher forecasting accuracy. The backward feature elimination technique showed higher accuracy (91.1602%) than correlation-based feature selection (90.6077%).

Srivastava et al. (2018), similarly to Usman et al. (2020), proposed feature selection to improve the performance of the classification model aimed at supporting open elective course unit selection for Engineering students. The researchers applied K-Nearest Neighbors, Support Vector machine, Decision Tree and Naïve Bayes classifiers to the dataset of 1988 engineering students. Both K-Nearest Neighbors and Support Vector Machine showed the same high level of accuracy which made 98.81%. Interestingly, most engineering students choose to take non-engineering course units such as Psychology, Digital Marketing, Photography, E-commerce, etc., to expand their knowledge base.

To forecast the results of student performance, Injadat et al. (2020) divided the predicted grades into three classes: 'Good', 'Fair' and 'Weak'. The employed forecasting methods represented K-Nearest Neighbor, Random Forest, Support Vector Machine, Multinomial Logistic Regression, Naïve Bayes and Neural Networks. To improve forecasting accuracy, they applied a hyper-parameter optimization technique to maximize the average Gini index. Based on relatively small datasets from the University of Genoa (115 first-year students) and the University of Western Ontario (486 second-year students), the researchers managed to achieve 93.1% forecasting accuracy.

Akçapınar et al. (2019) added Decision Tree and CN2 Rules to the aforementioned algorithms (Injadat et al., 2020). In pursuit of the same goal as Usman et al. (2020) and Injadat et al. (2020), they managed to achieve 89% accuracy by training forecasting models based on student interaction data from an online learning environment.

In contrast to the current research, the aforementioned authors (see Appendix A) binned the predicted learning results (entrance exam results, upcoming course unit grades at the university, failures in completing a course unit, etc.), and therefore classification methods were used instead of numerical forecasting methods. Forecasting student performance using the classification of the forthcoming grades into a limited number (usually three or five) of possible classes increases the value of prediction accuracy indicators and decreases individual targeting possibilities of

recommendations based on forecasting results. It happens because distinguishing individual learning results inside groups, e.g., 'Good' or 'Weak' is hardly possible.

It is not a problem for the researchers having a very limited number of actions to choose from. For example, if they aim to decide whether a particular student needs additional attention from the teacher delivering an e-learning course unit, they do not need to forecast precise grades (Injadat et al., 2020; Mythili and Shanavas, 2014).

However, since the current research aims to address individual learning results and target personal suggestions of undergraduate study programmes, predicting a precise result is necessary. This dictates the use of regression methods for predicting numerical values. For example, Martins et al. (2019) applied Random Forest and feature selection algorithms to forecast academic performance (weighted average of course unit grades considering the ECTS of completed and failed course units) based on data collected from 4530 students. Conforming to the current research, Martins et al. (2019) used the root mean squared error (RMSE) as a model evaluation metric and the feature selection algorithm to choose a feature set for constructing a predictive model, which allows excluding demographic, socioeconomic and access data variables from the predictive model. However, since the goal was to predict the learning results of the admitted students, the researchers used data on student performance from the previous terms to predict the expected learning results. Therefore, the structure of independent variables differs from our research that refers to school learning results only. The final set of independent variables used in the predictive model consists of the number of the ECTS failed in the academic term, the code of the degree course unit, the average mark of the course units passed in the academic term, the number of the ECTS passed in the academic term, a fraction of the ECTS credited to the student, the code of the school, the number of the ECTS credits awarded for the course unit/module in the degree, the type of enrolment for the degree course unit, the academic year of the academic term considered, the year of enrolment, student's year in the academic term considered.

Sawant et al. (2019), similarly to Martins et al. (2019), used data on the marks of the previous term to predict the expected marks based on the regression type of the Decision Tree forecasting method. The researchers used a small dataset of 262 student records from Shivaji University thus achieving 81% accuracy.

Moreno-Marcos et al. (2019) studied the impact of schoolchildren participation and activities in Small Private Online Courses (SPOC) from KU Leuven on the expected scores of the university admission test. For that purpose, they combined student SPOC clickstream data with admission test data. Four forecasting methods, including Random Forest, Generalized Linear Model, Support Vector Machines and Decision Tree were applied. The scientists used small datasets consisting of 114 records from 2016/2017 and 116 records from 2017/2018 academic years and managed to achieve relatively high accuracy in predicting success in the admission test. Lower RMSE (0.11) was observed in the model developed based on the Support Vector Machine method.

The applicability of the studies analysed above (see Appendix A) to the current research is limited by differences in

1) the discretization of the target variable, because the majority of the researchers binned the predicted learning results making it impossible to individually target undergraduate programmes for high school graduates;
2) research goal, as the majority of the researchers aimed their studies to support teacher decision-making whether a particular student needed additional attention to avoid a poor result; only single research (Hanandeh et al., 2020) addressed the problem of choosing an undergraduate study programme by the criterion of maximizing Grade Point Average (GPA);
3) the volume of data, because processing relatively small samples of students made difficulties in the identification of stable dependencies and relationships;
4) the structure of the data used for forecasting since the majority of research uses university rather than school learning results of the admitted students considering the ECTS;
5) However, despite the aforementioned drawbacks, Hyper-parameter optimization and Backward feature elimination have proven to be effective methods and have been incorporated into the current research.

3. Materials and methods

Many authors mentioned in the review of the existing literature use only one or two forecasting methods (Appendix A). Usually, the choice of such methods is not explained, except for the Decision Tree, favored for its interpretability.

Literature review suggests an additional hypothesis: no optimal forecasting method predicts student learning results. Student learning patterns have different trends in different study programmes at different universities. Therefore, each particular case requires a different forecasting method.

Therefore, there is a need for a recommendation engine capable of automating the modeling process. It should be able to train and optimize hundreds of thousands of models using different machine learning methods and settings (parameter optimization, Injadat et al., 2020), as well as sets of independent variables (feature selection: Akçapınar et al., 2019; Martins et al., 2019; Usman et al., 2020) for each study program at each university. The model with the lowest forecasting error will be selected for implementation. After each examination session and admission, these operations must be repeated for each university and study program. This necessity arises due to changes in data that occur after examination sessions and admission, which dictate the need to update the predictive models.

For the Undergraduate Study Programme Search System creation, student data was collected from 2 Lithuanian universities and the school electronic diary containing information from 597 Lithuanian educational institutions. All information is non-personal without names and IDs.

The universities provide information about Bachelor programs for 24,237 students from VGTU and for 1948 students from LSU. The original data (presented in **Table 1**) include:

Table 1. Data on students from the selected universities.

Features	VGTU	LSU
Country where the enrollee graduated	25	8
admission year to the university	2009–2019	timeframe 2011–2019
Graduation year	2013–2019	2013–2019
gender	Male/female	Male/female
Student age at the admission to the university	range 17–54	range 19–58
drop-outs	"TRUE"/"FALSE"/"CHANGE"	"TRUE"/"FALSE"/"CHANGE"
the number of failures of a subject	0 to 49	0 to 46
the number of retakes	0 to 94	0 to 38
the average grade in university	5 to 10	5 to 9.94

Note: The average grade in university is calculated by averaging all university grades for completed courses, taking into account course credits as weights.

School electronic diary provides information about school marks with course names (Lithuanian, Mathematics, Foreign language, Physics, Chemistry, History, Information technologies, Geography, Biology, Native language, Second foreign language, Astronomy, Drawing, Political Sciences, Music, Physical education, Painting, Moral education (Ethics/Religion), Art).

School marks are in a 10-point scale (from 4 to 10).

In the database, there are mistakes and missing values. So, the database was cleaned.

Rows with missing important information were deleted:

1) missing value about the Country where the enrollee graduated, admission year to the university, Graduation year, Gender, Student age at the admission year, drop-outs, the number of subject failures, and the number of retakes;
2) if the university applicant has less than 4 subjects with marks.

After rows with missing values were removed, there were 22,245 students from VGTU and for 1772 students left in the database.

Further, the rows that included inconsistencies and contradictions were deleted.

Finally, the cleaned database contains information about VGTU (21,426) and LSU (1674) students.

Then an experiment was performed to select algorithms for further optimization and integration into the Recommendation Engine. The machine learning algorithms for predictive model development were examined to forecast the outcomes.

The algorithms with a minimum forecasting error were selected for forecasting such numerical parameters as average university grade, the number of resits and failures to complete the courses (Iurasov, 2022).

The algorithms with a minimum forecasting error were selected for further optimization and integration into the Recommendation Engine. Seven different forecasting methods for each outcome were selected (see **Figure 1**):

1) Single Regression Tree (SRT) is a forecasting method based on the decision tree algorithm, where the target variable can take continuous values (typically real

numbers). The method was described by "Classification and Regression Trees" (Breiman et al., 1984). Advantages include fast, reliable learning, fast and easy interpretation, tolerance for missing data;

2) Ensemble of Regression Trees (ERT) is a forecasting method, which composes a weighted ensemble of multiple regression trees and is applied by using a simple mean of individual predictions (Loh, 2014). Each regression tree model is learned on a different set of records of pupil's data and a different set of pupil's attributes. The advantages of combining multiple regression trees include relatively lower forecasting error, overfitting avoidance, fast and scalable handling of Big Data, and missing data tolerance (Kazemi and Sullivan, 2014);
3) Bayesian Additive Regression Trees (BART) is a forecasting method based on a meta classifier that improves the performance of a regression-based classifier. By choosing the optimal value for the shrinkage (learning rate) parameter, it is possible to prevent overfitting (Management Association, 2017) and achieve a smoothing effect (see the fourth step of the Recommendation Engine algorithm for more information). The method follows the algorithm described by Friedman and Jerome (2002). Advantages include missing data tolerance, a pliable advance to fit a variety of regression models and algorithms while avoiding powerful parametric assume;
4) Stochastic Gradient Boosted Regression Trees (SGBRT) is an explicit regression gradient boosting algorithm that uses shallow regression trees, a particular form of boosting for building an ensemble of trees and optimizing any differentiable loss function. The SGBRT performs learning of the regression tree models on a different set of pupil's records (row sampling) and a different set of pupil's attributes (attribute sampling) similar to the ERT. The implementation follows the algorithm in section 4.4 (Friedman and Jerome, 2001). The advantages of SGBRT include a relatively higher forecasting accuracy, and missing data tolerance;
5) Random Forest Regression (RFR) is a forecasting method based on an ensemble of regression trees similar to ERT, where a random subset of pupil attributes is used to determine the most discriminative thresholds as the splitting rule. As in ERT and SGBRT, each regression tree model is learned on a different set of pupil records and attributes. More trees reduce the prediction variance; however, it can lead to high computational costs. The predicted value is a simple mean of the individual regression tree predictions. The method follows the algorithm described by Breiman (Friedman and Jerome, 2001). The advantages of RFR include a lower forecasting error, suggested efficient estimates of the trial mistake, effective for estimating missing data;
6) Regression by Discretization (RBD) is a forecasting method that uses an ensemble of classifiers on data with a discretized class attribute (transferring continuous variables into discrete counterparts). The method follows the algorithm described by Frank and Bouckaert (2009). The advantages of RBD are missing data tolerance and supporting conditional density estimation by building a univariate density estimator from the target values in the training data, weighted by the class probabilities;

7) Probabilistic Neural Network (PNN) is a feedforward neural network based on the Dynamic Decay Adjustment (DDA) method on labelled data using Constructive Training of PNN as the underlying algorithm (Berthold and Diamond, 1998). This method generates rules based on numeric data. The advantage of PNN is the ability to calculate likelihood scores for prediction, spending less time on efficient training than other implementations of neural networks, and missing data tolerance.

Figure 1. Predictive model creation to forecast the number of fails to complete courses in the study programme "Air Traffic Control" based on KNIME Analytics Platform.

One of the distinct advantages of the methods mentioned above is missing data tolerance. As described in the first article of this series (Iurasov & Iurasov, 2022), applicants for a Bachelor's programme must submit grades in four school courses per programme. Consequently, many of the school grades in the university student databases are missing values (shown as red question marks in **Table 2**). Hence the significance of missing data tolerance.

Table 2. A chunk of a dataset with red question marks indicating missing values.

LIETUVIU K.	MATEMATI KA	UŽSIENIO K.	FIZIKA	CHEMIJA	ISTORIJA	INFORMACINĖS TECHNOLGIJOS	GEOGRAFIJ A	BILOGI JA	GIMTOJ I KALBA
6.257	5.186	4.3	6.1	?	3.3	7.45	?	?	?
7.986	7.986	7.343	7.779	6.5	?	7.3	?	?	?
7.093	7.107	8.243	7.107	5.9	6.571	6.5	?	?	?
5.007	7.429	5.5	7.75	5.9	5.9	8.7	?	?	?
4.614	8.45	6.164	7.45	6.5	5.9	8.7	?	?	?
6.91	6.55	7.25	5.55	5.9	7.9	7.2	?	5.9	7
6.91	6.55	7.25	5.55	5.9	7.9	7.2	?	5.9	7
8.943	8.064	8.957	8.7	5.9	7.2	7.55	?	?	?

Table 2. (*Continued*).

LIETUVIU K.	MATEMATI KA	UŽSIENIO K.	FIZIKA	CHEMIJA	ISTORIJA	INFORMACINĖS TECHNOLGIJOS	GEOGRAFIJ A	BILOGI JA	GIMTOJ I KALBA
4.65	5.343	5.007	4.4	?	5.3	7.2	?	?	?
7.771	6.393	6.964	5.9	5.3	6.564	4.4	?	5.9	5.3

As seen in **Figure 1**, ERT was identified as the lowest RMSE modelling method for predicting the number of failures to complete the courses in the "Air traffic control" study programme delivered at VGTU. As a result, the ERT model was developed and saved together with other related information (green traffic light signs, in a row of nodes associated with Tree Ensemble Learner node). The right part of **Figure 1** shows the optimal settings determined during Hyper-Parameter Optimization and the resulting RMSE (mentioned values are underlined with red lines):

1) the number of models (N_models), i.e., the number of regression trees to be learned, is 30. A "reasonable" value can range from very few (say 10) to many thousands, although a value between 100 and 500 suffices for most datasets;
2) the number of tree levels to be learned (MaxLevels) is 7. For instance, a value of 1 would only split the (single) root node (decision stump);
3) the objective value is the RMSE of the model with the aforementioned optimal settings (2.76).

The algorithm of the Recommendation Engine includes 15 basic steps. Steps from 3 to 9 are shown in **Figure 1**:

Step 1. Data Cleaning: This involves processing missing values, handling missing students, and filtering out duplicate records and records of students who dropped out and did not graduate. For instance, the average university grade will be calculated only for students who have completed their education.

Step 2. Educational Data Pre-processing: If some undergraduate programs have changed their names, historical student data from the past needs to be used, but the actual name of the program today should be recorded. Old study program names were replaced with new ones. Data on closed study programs were filtered out as these will not be recommended to school graduates. After these steps, student records decreased from 21,426 to 18,720 at VGTU and from 1674 to 1532 at LSU.

Step 3. Data Separation: This is done to calculate the accuracy of the developed models. The data are divided into training (80%, to create forecasting models) and a test dataset (20%, to assess their accuracy). Random sampling of all rows with a fixed seed is used to ensure reproducibility.

Step 4. Determining Optimal Modeling Settings: The Hyper-Parameter Optimization algorithm, identified during the literature review as effective for improving forecast accuracy (Injadat et al., 2020), is employed. This algorithm consists of two phases: a warm-up phase involving the random selection and evaluation of parameter combinations, and an actual Bayesian optimization phase selecting parameter combinations based on past estimates. The Tree-structured Parzen Estimator Approach is used for optimization. The second phase aims to find promising parameter combinations, ultimately resulting in the optimal combination of parameters. The number of parameters to be optimized varies from two to four,

depending on the machine learning method. For example, settings for BART include optimizing the parameter "shrinkage rate" (see **Figure 2**):

- the start value of the shrinkage rate was set to 0.005 to define the start point of the parameter space. Values below 0 do not make sense, as they would imply that the algorithm is running the opposite direction from the gradient (Friedman, 2002);
- the stop value of the shrinkage rate was set to 1 to define the endpoint of the parameter space because values above 1 do not make sense, as they would imply no convergence;
- the shrinkage rate optimization step size (0.005) was determined to limit the number of possible parameter values. Experiments with VGTU and LSU data revealed the following range of optimal shrinkage rates: from 0.26 (forecasting models to predict: 1) number of retakes to complete the courses in study programmes "Aviation Mechanics Engineering" and "Event Engineering"; 2) average university grade for study programme "Transport Engineering") to 1 (forecasting model to predict the number of fails to complete the courses of the study programme "Creative Industries").

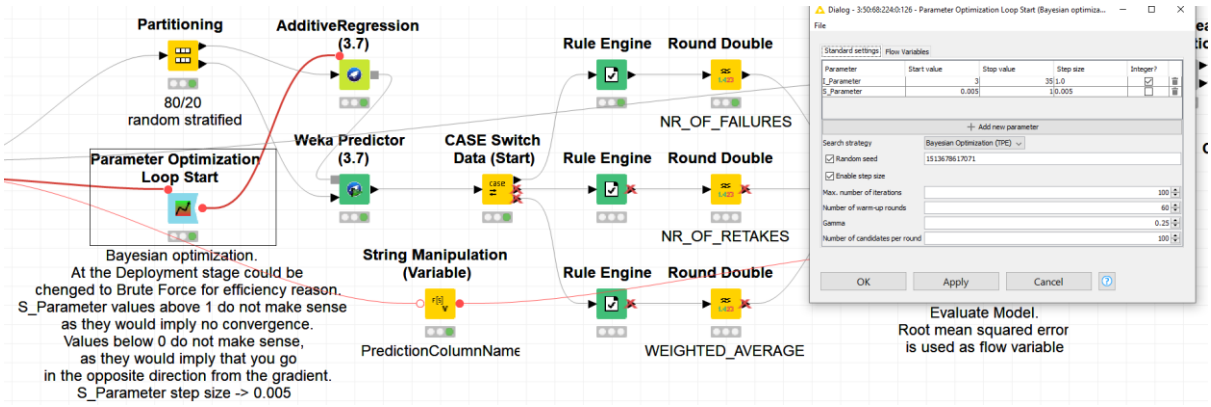

Figure 2. The settings of Hyper-Parameter Optimization loop for BART forecasting method.

Step 5. To enhance prediction accuracy and determine which pupil characteristics each specific forecasting model requires, Backward Feature Elimination has been implemented. This algorithm, identified during the literature review as effective in Educational Data Mining (Usman et al., 2020), is an iterative approach. It begins with selecting all student characteristics. In each iteration, the feature with the least significant impact on the forecasting model's performance is removed (Salappa et al., 2007). This iterative feature elimination process helps identify only those pupil characteristics significantly influencing the forecasting result.

Specified Lower Bound. Since Backward Feature Elimination subtracts pupil characteristics, a lower bound for the number of selected characteristics was specified. This lower bound was set to 7 (**Figure 3**) to prevent the development of a model with a minimal number of pupil characteristics (e.g., one or two) that may exhibit the lowest error and be chosen for deployment in the Recommendation Engine. Such instances can occur when training the model on data from a new study program with few rows.

Minimum Pupil Characteristics. At least 7 pupil characteristics, including gender, nationality, age at the time of admission, and the four grades of compulsory courses

containing non-missing values, were considered. Often, even more student data features contain non-missing values, as many school graduates apply for different study programs, requiring the declaration of grades for subjects different from those required for their first choice. This lower bound ensures that numerous attributes are considered when creating the model.

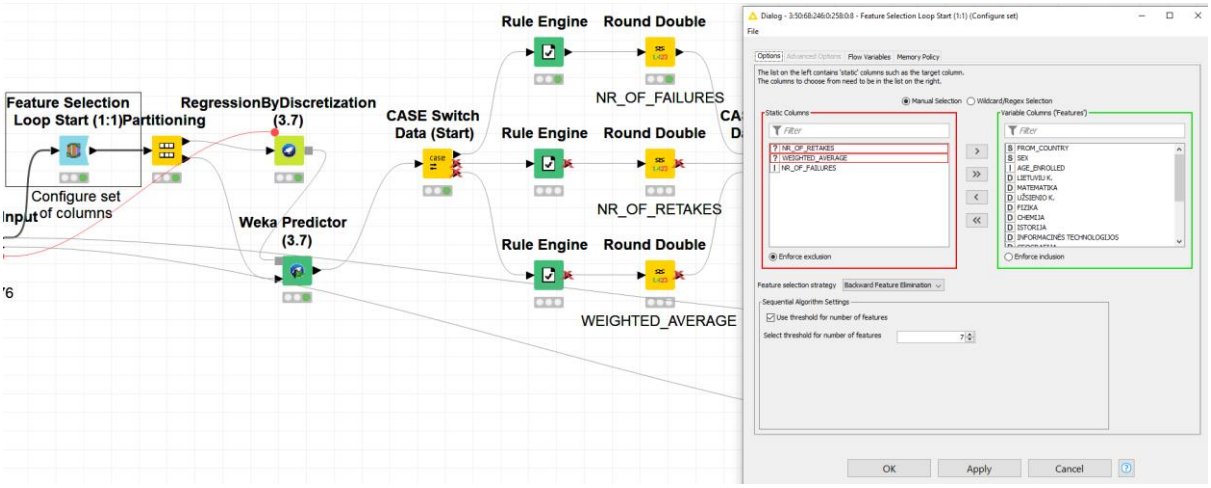

Figure 3. The settings of the Backward Feature Elimination loop for the RBD forecasting method.

A set of features is automatically selected to minimize prediction error, resulting in more accurate model outcomes. For instance, the optimized data structure for forecasting the number of failures to complete courses in the 'Air Traffic Control' study program includes age at the time of university admission and school grades for subjects such as foreign language, History, IT, Geography, Biology, Ethics, Native language, Second foreign language, Music, and Physical Education. Using additional data, such as gender and grades for Chemistry, increases the Root Mean Square Error (RMSE) of the forecasting model (see **Figure 4**).

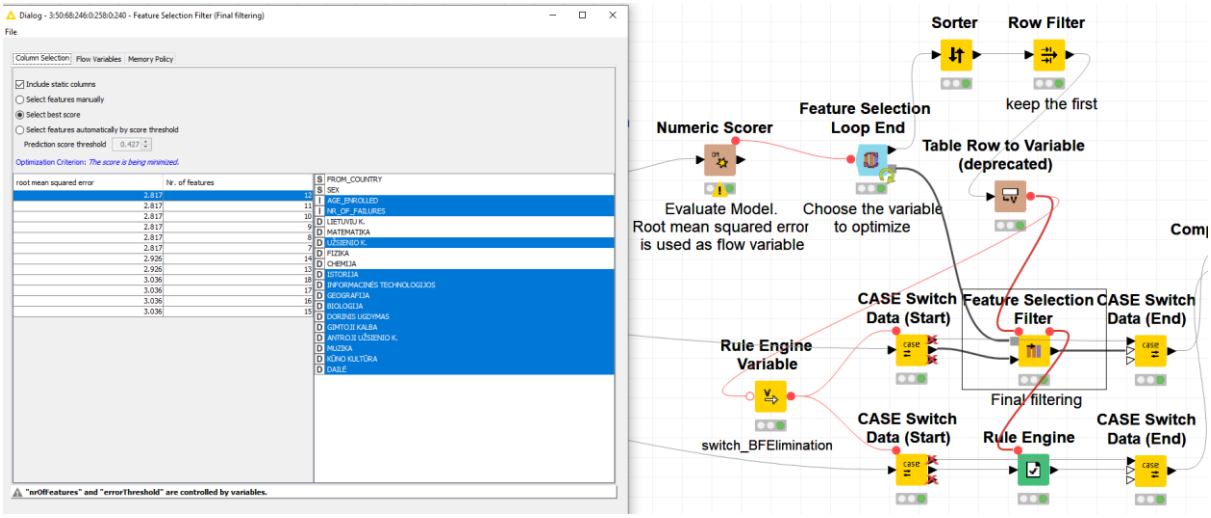

Figure 4. Structure of pupil characteristics for the RBD forecasting modeling with minimal RMSE.

Step 6. Choosing the most accurate forecasting model (Field 2 in **Figure 1**). The RMSE was used as an accuracy criterion to determine the accuracy of models. The RMSE is the standard deviation of the residuals (prediction errors). Residuals measure

how far from the regression line data points are. The RMSE is a measure of the residual spread. In other words, it tells us how concentrated the data are around the best fit line. The lower the RMSE of the model, the more accurate forecast it produces. The result of each forecasting model is a particular predicted learning outcome (average university grade, number of failures, and number of retakes) of future students at each undergraduate study program. To calculate the RMSE, 20% of initial data: a test dataset (created during step 3) was used. Those data are not used for forecasting model learning. Therefore, we can feed the test dataset to models to forecast learning results and, after that, evaluate the model accuracy by comparing predicted learning results versus actual ones.

Step 7. Training the most accurate forecasting model (from step 6), with previously identified optimal settings (from step 4) and the optimal data structure (from step 5). This time, training is conducted on the full array of available data (Field 3 in **Figure 1**) without dividing it into learning and test datasets. Therefore, the accuracy of the created model should be even higher than the one achieved in stages 4, 5, and 6. However, it is impossible to determine the RMSE of the final model, as all data has already been used. It is incorrect to assess the accuracy of the model based on the data used for its training: a competently constructed model will demonstrate about 100% accuracy.

Step 8. Saving the model for future deployment by using the "Model Writer" node (the model is inserted into a file: Field 4 in **Figure 1**). The structure of the model file name (partly visible on the right side of **Figure 1**): university acronym (e.g., "VGTU") + ";" + study programme name (e.g., "Air Traffic Control") + ";" + forecasted parameter of USPSS equation (e.g. "NR_OF_FAILURES") + the number of modeling method, starting from 0 for PNN to 6 for RBD + ";" + extension ".zip". For example, "VGTU;Air Traffic Control;NR_OF_FAILURES;3.zip".

Step 9. Saving information about the optimal data structure (determined at stage 5) was used to develop a forecasting model. The data structure used to train forecasting models of different predicted learning results for different study programmes is different. Therefore, the USPSS needs to extract from electronic school diaries only useful information for forecasting during the deployment stage. The file name construction follows the same path as the model, except the extension "csv" is added.

Step 10. Saving the information: the modelling method and the optimization settings that turned out to generate the most accurate forecasting model for each university study programme (Field 5 in **Figure 1**), RMSE of the model provided with 80% of the data used to train the model, etc., are saved. This information is used later to monitor the operation and debugging of the entire system.

The 10 steps are repeated for each study programme of each university and each criterion of the Recommendation Engine. Forecasting models should be stored in the Recommendation Engine and will be used to assist pupils.

Step 11. After pupils start searching where to pursue their degree (by clicking a button in the school electronic diary menu), the electronic diary will send a Representational State Transfer Application Programming Interface (REST API) request to the Recommendations Engine with part of the non-personal data of a specific pupil in JavaScript Object Notation (JSON) container. The structure of the container is shown in **Figure 5** (Field 4).

Step 12. To determine the order of study programmes on the recommendation list, university learning results need to be forecasted. For that purpose, each forecasting model must be fed with non-personal data of a pupil and interpreted. To interpret the forecasting models on the server, the Recommendation Engine uses KNIME Server, Field 1 in **Figure 5** shows the IP address of the KNIME Server where the REST API requests are served. The algorithm collects all CSV files from the directory "Study programs" (Field 3 in **Figure 5**). Each file describes the study programmes and is titled with an abbreviation of the university. The algorithm extracts forecasting models from directory "Models" (Field 2 in **Figure 5**) to predict learning results by iterating over each university and study program. Each model is then fed with pupil data from JSON container (Field 4 in **Figure 5**).

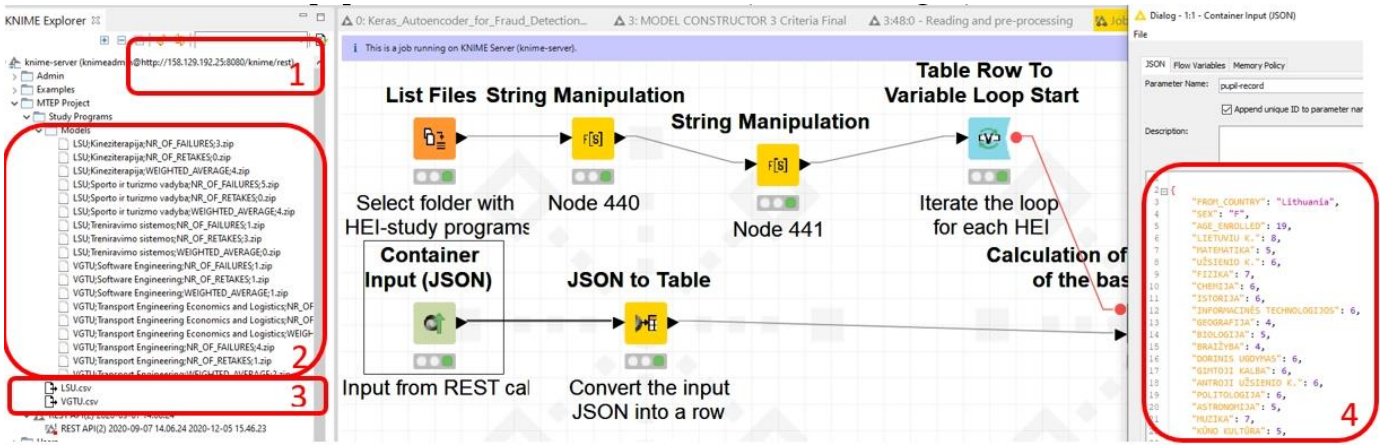

Figure 5. The initial stage of processing an incoming REST API request from the school electronic diary.

Step 13. According to step 8, the model's title consists of the university abbreviation, the name of the study program, the name of the predicted indicator, and the index of the modelling method (Field 2 in **Figure 5**). This information is used to select the correct interpretation predictor, interpret the model, and collect the predicted results (**Figure 6**).

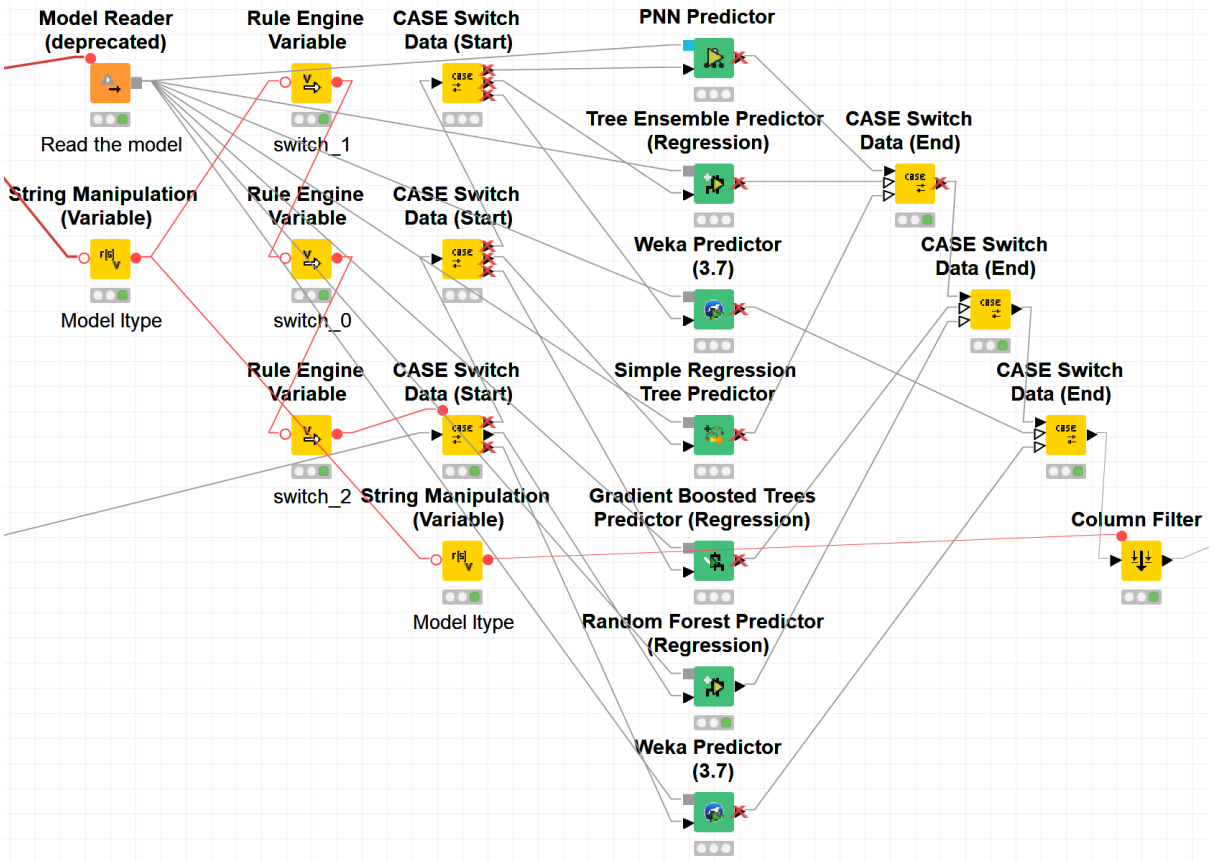

Figure 6. Interpretation of the model "VGTU;Transport Engineering Economics and Logistics;NR_OF_RETAKES;5" (modeling index of 5 indicates that the model needs interpretation in the RFR Predictor).

Step 14. After predicting the learning results, the Recommendation Engine calculates the rating of each study programme (Field 1 in **Figure 7**). The average university grade positively affects the rating, while the number of failures and retakes to complete the courses has an adverse effect. The list of individually suggested study programmes is formed by sorting them in descending order of their rating (Field 2 in **Figure 7**) and sent back to the school electronic diary in a JSON container (Field 3 in **Figure 7**).

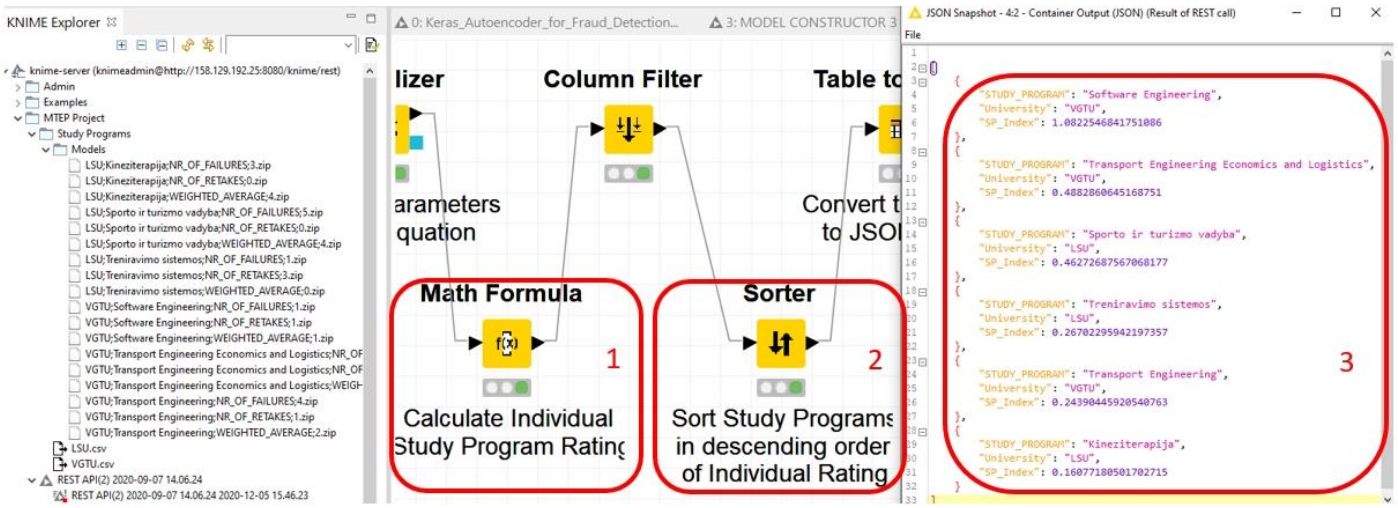

Figure 7. Compiling and sending back the list of recommended study programs.

Step 15. The school electronic diary displays the list of recommended study programmes to the pupil.

To add a new university to the Recommendation Engine, you need to repeat steps 1–10 and save:

1) description of study programmes in a CSV file (named after the abbreviation of a specific university) in the "Study programs" folder (Field 3 in **Figure 5**);
2) forecasting models in the "Models" folder, named as described in step 8 (Field 2 in **Figure 5**).

Further development of the system involves improvements for marketing purposes:

1) to make the resulting list look like a search engine results page with the study programme names that have become hyperlinks to their respective landing pages on the university websites;
2) to supplement the list with indicators by which the final rating is calculated; this will help the school graduate to understand the logic of the system;
3) to allow the pupil to play what/if scenarios by changing the school course grades (Field 4 in **Figure 5**) to the planned/intended ones, and see how the list of proposed study programme changes;
4) to let pupils change the weight coefficients themselves to consider personal preferences and the significance of each parameter in the final rating. For example, the number of retakes is not essential; pupils are ready to retake the exams many times. Then they reduce the weighting coefficient of this parameter, automatically increasing the importance of other parameters in the final rating.

4. Results

During the 10th step of the proposed algorithm, parameters for forecasting models and other related information have been aggregated and stored. The table Model performance metrics on VGTU and LSU data (Appendix B) systematizes the obtained information and identifies the study programme, the predicted parameter, the most accurate forecasting method, the RMSE of the model provided with 80% of the data used to train the model, etc. However, the actual (deployed in the Recommendation Engine) models are trained using 100% of the available data (Step 8). This provides higher accuracy in forecasting but does not make it possible to correctly calculate the RMSE. Therefore, further analysis covers the RMSE of the models built at Step 6, rather than at Step 8, bearing in mind that RMSE will be relatively higher than that of the actual model.

To summarize the results calculation of the average RMSE was carried out considering the number of students in each study program, divided by the total number of university students.

Average RMSE of models to predict Number of failures to complete the courses, calculated considering proportion of students from the study programm of the total number of university students, by following equation:

$$\text{Average RMSE}_{\text{course}}^{\text{fail compl}} = \sum_{k=1}^{M} \left(\frac{\text{RMSE}_{\text{course}}^{\text{fail compl}}{}_{k} \times N\_\text{stud\_prog}_{k}}{M \times N\_\text{total}} \right)$$

where

$\mathrm{RMSE}_{course}^{fail\ compl}{}_{k}$ is RMSE of the forecasting model predicting Number of failures to complete the courses for study program *k*;

M–number of study programs;

N_stud_progk–number of students enrolled in study program *k*;

N_total–the total number of students in the university database.

The similar equation used to calculate average RMSE of models to predict Number of retakes to complete the courses:

$$\text{Average RMSE}_{\text{course}}^{\text{retake}} = \sum_{k=1}^{M}\left(\frac{\text{RMSE}_{\text{course}_{\text{k}}}^{\text{retake}} \times N\_\text{stud\_prog}_{k}}{M \times N\_\text{total}}\right)$$

where

$\mathrm{RMSE}_{\mathrm{course}_k}^{\mathrm{retake}}$ is RMSE of the forecasting model predicting Number of retakes to complete the courses for study programm *k*.

Similar algorithm is used for calculation of RMSE of Average university grade.

$$\text{Average RMSE}_{\text{grade}}^{\text{university}} = \sum_{k=1}^{M}\left(\frac{\text{RMSE}_{k}^{\text{grade}} \times N\_\text{stud\_prog}_{k}}{M \times N\_\text{total}}\right)$$

where

$\mathrm{RMSE}_{k}^{grade}$ is RMSE of the forecasting model predicting Average university grade for study program *k*.

The analysed information shows that:

1) the validity of an additional hypothesis (formed based on the literature review) indicates that there is no optimal method for predicting student learning results. For 153 study programme/predicted parameter combinations, the SGBRT proved to be more accurate 47 times, PNN–25, ERT–24, SRT–18, RFR–16, BART 13 and RBD 10 times, respectively, i.e., in each specific case, it is necessary to carry out forecasting modelling, Hyper-Parameter Optimization, Backward Feature Elimination and only then compare the accuracy of the models and choose the most accurate technique for each study programme and the predicted parameter;
2) the accuracy of the models is acceptable for use in the Recommendation Engine;
3) a relatively large number of student records in the dataset (VGTU) does not necessarily provide higher accuracy in the model. **Table 3** shows that forecasting accuracy for VGTU study programmes comparing to LSU study programmes, is higher only in the models built to predict the Average university grade;

Table 3. Comparative analysis of forecasting error.

Predicted parameters	**Boundaries of the original data array**		**Accuracy Error of forecasting models**		
	Lower bound	**Upper bound**	**Average RMSE (VGTU)**	**Average RMSE (LSU)**	**Average RMSE (VGTU&LSU)**
Number of failures to complete course units	0	49	3.07	2.47	2.98
Number of retakes to complete course units	0	94	5.39	3.44	5.09
Average university grade	5	10	0.54	1.39	0.67

4) the wider is the range of the values (boundaries of the original data array) of the predicted parameter, the higher is model error. The RMSE quantifies the average

discrepancy between the predicted values from a statistical model and the actual observed values (see **Table 3**).

The proposed system utilizes Data Science algorithms to analyze the academic performance of both university and high school students. It then recommends the most suitable undergraduate study programs for high school students. These recommendations are based on multiple criteria designed to minimize the risk of dropping out, failing exams, needing multiple resits, or switching programs. The system aims to enhance the likelihood of successful admission and graduation, as well as improve the chances of securing the desired job.

5. Discussion

The study proposes an algorithm for forecasting university learning results within the Recommendation Engine, aiming to recommend undergraduate programs with a high probability of successful completion to high school graduates. Our system will be useful for advising students to achieve higher overall academic performance and graduate on time.

However, the mentioned university learning results only cover 3 out of 10 parameters in the USPSS equation (author, year). Future research directions include developing algorithms for calculating the remaining parameters.

Additional research avenues involve broadening the scope beyond demographics and school learning results. Consideration of career interests, employment prospects, psychological characteristics, and other related information is under exploration. Integrating all these development areas into a cohesive system is the focus of ongoing research.

Research and development have substantiated the hypothesis that Data Science methods and algorithms can lay the foundation for a new approach to university study program choices. This approach aims to match high school graduates with undergraduate study programs, thereby improving expected learning results, student satisfaction, and future career prospects based on analyzed data from universities and electronic school diaries.

The author presents the theoretical basis of the Recommendation Engine, capable of recommending study programs based on various criteria to reduce the likelihood of exam failures or multiple retakes while increasing the likelihood of successful completion. Testing and analyzing the performance indicators of the Recommendation Engine demonstrate the feasibility of implementing the proposed system with sufficient accuracy.

There are some limitations to this work that need to be set. The used dataset is comparatively small, and the current gender study program distribution of participants may have a gender bias based on dataset features. Data from universities and schools are not shared by market participants. To successfully develop forecasting models for the Recommendation Engine, the USPSS requires data on the individual performance of high school and university students. Currently, such data are internal and inaccessible to other participants in the educational market.

Next, this USPSS might not perform well on courses with recent changes in popularity. Updating the model regularly every year could help alleviate this issue. Another limitation of our work is that our USPSS can not perform for new study programs. This is due to the lack of data on training and completion of new educational programs. In future work, we will explore how to prevent the recomendation mistakes based on study program changing or for new educational programs.

IT algorithms and economic incentives will be addressed in the upcoming articles of this series. Further research will narrow the gap between secondary and tertiary education by completing the USPSS for high school graduates considering additional recommendation criteria and more information about students. This e-business model will:

1) disrupt the traditional arrangement of the educational market based on the offline exhibitions of university education (obsolete form of educational marketing of the 20th century) and directory websites (catalogues) with the unstructured text annotations of study programmes;
2) increase the efficiency of university education thus ensuring higher learning results and student satisfaction;
3) provide students with the tools of the objective assessment and comparison of various study programmes to satisfy their needs and individually tailor the lists of undergraduate study programmes offered by universities. Instead of using keywords for searching the names of course units and study programmes (as in the case of directory websites), the USPSS is aimed at delivering the final product like an interesting job and a successful career. Hence, it will contribute to ensuring the smooth transition of school graduates to the labour market.

To create the algorithm, 80% of the obtained database was used, and the results were tested on the remaining 20%. The article's results are based on forecasts and their testing on those 20% of students from the obtained database. Therefore, the described results are based on a comparison of forecasts and real data on student learning.

However, there are no real calculations or evidence of successful intervention using the developed system, as there is no data on the completion of studies by students enrolled in the curriculum based on the system's recommendations.

Additionally, this study aims to develop a theoretical framework, so the forecasts are made relative to student grades from the existing database. Further research will focus on clarifying and adjusting this system based on practical results.

Author contributions: Conceptualization, AI and RL; methodology, AI; software, OI; validation, OI, AI and RL; formal analysis, AI; investigation, OI and AI; resources, AI; data curation, OI; writing—original draft preparation, RL and AI; writing—review and editing, RL and OI; visualization, AI; supervision, AI; project administration, RL; funding acquisition, OI and RL. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors declare no conflict of interest.

References

Akçapınar, G., Altun, A., Aşkar, P. (2019). Using learning analytics to develop early-warning system for at-risk students. International Journal of Educational Technology in Higher Education, 16(40). https://doi.org/10.1186/s41239-019-0172-z

Arora, G., Kumar, A., Devre, G. S., Ghumare, A. (2014). Movie Recommendation System Based on Users' Similarity. International Journal of Computer Science and Mobile Computing, 3(4), 765–770.

Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B. (2011). Algorithms for hyper-parameter optimization. Neural Information Processing Systems Foundation, Curran Associates Inc. pp. 2546–2554.

Berthold, M. R., Diamond, J. (1998). Constructive training of probabilistic neural networks. Neurocomputing, 19(1–3). https://doi.org/10.1016/S0925-2312(97)00063-5

Bokde, D., Girase, S., Mukhopadhyay, D. (2015). An Approach to a University Recommendation by Multi-criteria Collaborative Filtering and Dimensionality Reduction Techniques. In: Proceedings of the 2015 IEEE International Symposium on Nanoelectronic and Information Systems; 21–23 December 2015. pp. 231–236.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–35. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J., Olshen, R. A., Stone, C. J. (1984). Classification and Regression Trees (Wadsworth Statistics/Probability), 1st ed. Chapman and Hall/CRC.

Frank, E., Bouckaert, R. R. (2009). Conditional Density Estimation with Class Probability Estimators. Springer Publidhing.

Friedman, Jerome H. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5). https://doi.org/10.1214/aos/1013203451

Friedman, Jerome H. (2002). Stochastic gradient boosting. Computational Statistics & Data Analysis, 38(4), 367–378.

Girase, S., Powar, V., Mukhopadhyay, D. (2017). A user-friendly college recommending system using user-profiling and matrix factorization technique. In: Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA); 5–6 May 2017; Greater Noida, India.

Hahsler, M. (2015). Recommenderlab: A framework for developing and testing recommendation algorithms. Available online: https://www.Researchgate.Net/Publication/237246291_recommenderlab_A_Framework_for_Developing_and_Testing_Recommendation_Algorithms. (accessed on 20 August 2024).

Hanandeh, F., Al-Shannaq, M. Y., Alkhaffaf, M. M. (2020). Using Data Mining Techniques with Open Source Software to Evaluate the Various Factors Affecting Academic Performance: A Case Study of Students in the Faculty of Information Technology. International Journal of Open Source Software and Processes, 7(2), 72–92.

Huynh, T. M., Huynh, H. H., Tran, V. T., Huynh, H. X. (2018). Collaborative filtering recommender system base on the interaction multi-criteria decision with ordered weighted averaging operator. In: Proceedings of the 2nd International Conference on Machine Learning and Soft Computing-ICMLSC' 18; 2–4 February 2018; New York, NY, USA. pp. 45–49.

Injadat, M., Moubayed, A., Nassif, A. B., Shami, A. (2020). Multi-split optimized bagging ensemble model selection for multi-class educational data mining. Applied Intelligence, 50(12). https://doi.org/10.1007/s10489-020-01776-3

Iurasov, A. (2022). New e-business model: Undergraduate study program search system. International journal of learning and change, 14(5/6), 500–514. https://doi.org/10.1504/ijlc.2021.10035252

Iurasov, A., Iurasov, A. (2022). Forecasting of successful completion of university study programs: Data preprocessing and optimization of LAMA BPO algorithm. Applied business: Issues & solutions, 1, 32–41. https://doi.org/10.57005/ab.2022.1.5

Kazemi, V., Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition; 23–28 June 2014; Columbus, OH, USA.

Kazi, A. S., Akhlaq, A. (2017). Factors Affecting Students' Career Choice. Journal of Research and Reflections in Education, 11(2), 187–196.

Loh, W.-Y. (2014). Fifty Years of Classification and Regression Trees. International Statistical Review, 82(3). https://doi.org/10.1111/insr.12016

Information Resources Management Association. (2017). Artificial Intelligence: Concepts, Methodologies, Tools, and Applications. IGI Global.

Martins, M. P. G., Miguéis, V. L., Fonseca, D. S. B., Alves, A. (2019). A Data Mining Approach for Predicting Academic Success–A Case Study. In: Information Technology and Systems: Proceedings of ICITS 2019. Springer.

Meenakshi, E., Satpal, D. (2019). Recommendation Engine: A Best Way for Providing Recommendation of Any Items on the Internet. International Journal of Engineering Research & Technology, 7(12), 1–7.

Moreno-Marcos, P. M., De Laet, T., Muñoz-Merino, P. J., et al. (2019). Generalizing Predictive Models of Admission Test Success Based on Online Interactions. Sustainability, 11(18). https://doi.org/10.3390/su11184940

Mythili, M. S., Mohamed Shanavas, A. R. (2014). An Analysis of students' performance using classification algorithms. IOSR Journal of Computer Engineering, 16(1). https://doi.org/10.9790/0661-16136369

Rivera, A. C., Tapia-Leon, M., Lujan-Mora, S. (2018). Recommendation Systems in Education: A Systematic Mapping Study. In: Proceedings of the International Conference on Information Technology & Systems (ICITS 2018); 10–12 January 2018; Libertad City, Ecuador. pp. 937–947.

Salappa, A., Doumpos, M., Zopounidis, C. (2007). Feature selection algorithms in classification problems: An experimental evaluation. Optimization Methods and Software, 22(1), 199–212. https://doi.org/10.1080/10556780600881910

Sawant, T. U., Pol, U. R., Patankar, P. S. (2019). Educational data mining prediction model using decision tree algorithm. International Journal of Emerging Technologies and Innovative Research, 6(5), 306–313.

Sneha, M., Priya, J., Shubhangi, B., Priyanka, I. (2016). Recommendation System for MS. International Journal for Innovative Research in Science & Technology, 2(11), 460–470.

Srivastava, S., Karigar, S., Khanna, R., Agarwal, R. (2018). Educational Data Mining: Classifier Comparison for the Course Selection Process. In: Proceedings of the 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE); 11–12 July 2018; Kuala Lumpur, Malaysia.

Usman, M. M., Owolabi, O., Ajibola, A. (2020). Feature Selection: It Importance in Performance Prediction. International Journal of Engineering Science and Computing, 10(5), 25625–25632.

Webometrics Ranking of World Universities (2021). Countries arranged by Number of Universities in Top Ranks. Available online: http://www.webometrics.info/en/node/54 (accessed on 25 August 2024).

Appendix A

Table A1. Applicability of university learning outcome prediction studies to current research.

Discretization of the target variable	Aim	Target variable	More accurate method/ other methods	Accuracy metrics	Independent variables	Dataset size, students	Student's field of study	Authors	Applicability
Yes (Classification approach)	To give advice to students in which university and study programme they have to study to get high grades	Accumulative averages classified by groups (Excellent, Very good, Good, Pass, Fail)	Decision Tree (J48 algorithm)/ Naïve Bayes	Accuracy is 46.8%	University, Study program, High school average GPA, Faculty, Admission type, Nationality, Number of credit hours finished by the student, Gender	14,719	Information Technology	Hanandeh et al., 2020	The aim of the study partly coincides with the aim of the current research, but the applicability of this study is very limited by the contradiction between its aim and the structure of the data used to develop the forecasting model
	To support open elective course selection of the admitted students	Open elective course	K-Nearest Neighbors and Support Vector machine/ Decision Tree and Naïve Bayes	Accuracy is 98.81%	Admission year, study program, previously allotted open elective details	1988	Engineering	Srivastava et al., 2018	Applicability is very limited by differences in research aim, discretization of the target variable, and the structure of the data used for forecasting
	To decide whether a particular student needs additional attention from the teacher to avoid a poor result	Course final grades classified by groups (Good, Fair, Weak)	Multinomial Logistic Regression/ K-Nearest Neighbors, Random Forest, Support Vector Machine, Naïve Bayes, Neural Network, Hyper-parameter optimization	Accuracy is 93.1%	Course marks at two different course delivery stages: 20% and 50% mark	601	Engineering	Injadat et al., 2020	Hyper-parameter optimization has proven to be effective and incorporated into ongoing research
		Course final grades classified by groups (Failed, Passed)	K-Nearest Neighbors + correlation-based feature selection/ Naïve Bayes, Decision Tree, Support Vector Machines, Random Forest, Neural Network, and CN2 Rules	Accuracy is 89%	Students' interaction data from the online learning environment	76	Computer hardware	Akçapınar et al., 2019	Feature selection has proven to be effective and incorporated into ongoing research
		Final year grade classified by groups (First Class, Second upper Class, Second lower class, Third class, Pass, Not graduate)	Naïve Bayes + Backward feature elimination/ correlation-based feature selection	Accuracy is 91.16%	School grades of five courses: English, Mathematics, Physics, Chemistry, Biology	543	Not specified by the authors	Usman et al., 2020	Backward feature elimination has proven to be effective and incorporated into ongoing research

Table A1. (*Continued*).

Discretization of the target variable	Aim	Target variable	More accurate method/ other methods	Accuracy metrics	Independent variables	Dataset size, students	Student's field of study	Authors	Applicability
No (Regression approach)	To decide whether a particular student needs additional attention from the teacher to avoid a poor result	The weighted average of the course grades, taking into account the ECTS of the courses completed and failed	Random Forest and feature selection	RMSE, R2	Student performance from previous semesters	4,530	Engineering students of Polytechnic Institute of Bragança	Martins et al., 2019	Applicability is limited by differences in research aim and the structure of the data used for forecasting
		Final semester result	Decision Tree	RMSE is 2.7, R2 is 0.81	Student performance from previous semesters	262	Not specified by the authors	Sawant et al., 2019	
	To predict who will pass the admission test	University admission test score	Support Vector Machines/ Random Forest, Generalized Linear Model, and Decision Tree	RMSE is 0.11	SPOC clickstream data, admission test results	230	School pupils	Moreno-Marcos et al., 2019	

Appendix B

Table B1. Models performance metrics on VGTU and LSU data.

Undergraduate study program	Number of students	Number of students without dropping out	Forecasted parameters					
			Number of failures to complete the courses		Number of retakes to complete the courses		Average university grade	
			Minimum RMSE	Used Data Science algorithm	Minimum RMSE	Used Data Science algorithm	Minimum RMSE	Used Data Science algorithm
Vilnius Gediminas Technical University								
Air Traffic Control	143	99	2.610	SRT	3.107	SGBRT	0.423	ERT
Aircraft Piloting	152	139	1.964	BART	3.287	RBD	0.388	SGBRT
Architecture	793	658	3.075	SRT	3.240	SGBRT	0.558	ERT
Automation and Control	95	79	3.285	SGBRT	7.626	RBD	0.500	ERT
Aviation Mechanics Engineering	528	326	3.438	SGBRT	6.063	SRT	0.554	ERT
Avionics	477	344	3.672	PNN	7.439	BART	0.696	ERT
Bioengineering	508	388	3.698	SGBRT	4.555	ERT	0.721	SGBRT
Biomechanics	307	212	3.053	SGBRT	6.325	RBD	0.570	SGBRT
Building Energetics	411	315	5.042	SRT	13.805	ERT	0.608	BART
Business Analytics	10	8	0.000	PNN	1.414	PNN	0.065	ERT
Business Logistics	273	181	3.469	SGBRT	0.707	SRT	0.538	RFR
Business Management	1200	649	4.293	PNN	8.553	ERT	0.447	BART
Civil Engineering	1459	913	6.035	SRT	10.614	BART	0.657	RFR
Computer Engineering	220	126	3.571	SRT	6.523	RBD	0.578	SGBRT
Construction and Real Estate Management	39	20	2.525	BART	4.269	SRT	0.316	PNN
Creative Industries	1018	818	3.869	SGBRT	3.844	ERT	0.536	RFR
Data Analysis Technology	100	73	1.720	PNN	1.969	SGBRT	0.690	PNN
Digital Manufacturing	179	121	2.034	BART	6.225	SGBRT	0.471	ERT
Economics Engineering	468	325	4.344	SGBRT	6.401	ERT	0.555	SGBRT
Electrical Energetics engineering	47	41	1.291	SGBRT	1.708	ERT	0.629	SRT

Table B1. (*Continued*).

Undergraduate study program	Number of students	Number of students without dropping out	Forecasted parameters					
			Number of failures to complete the courses		Number of retakes to complete the courses		Average university grade	
			Minimum RMSE	Used Data Science algorithm	Minimum RMSE	Used Data Science algorithm	Minimum RMSE	Used Data Science algorithm
Vilnius Gediminas Technical University								
Electronics Engineering	605	360	3.772	SGBRT	9.482	ERT	0.738	ERT
Entertainment Industries	403	296	3.943	PNN	3.012	PNN	0.700	SGBRT
Environmental Protection Engineering	659	394	3.801	SGBRT	9.170	RFR	0.727	ERT
Event Engineering	122	71	4.262	SGBRT	2.639	SGBRT	0.469	RBD
Financial Engineering	374	306	3.064	SRT	2.121	ERT	0.546	BART
Fire Protection	171	100	1.978	SGBRT	6.148	SGBRT	0.376	RFR
Geodesy	388	276	3.404	SRT	10.640	SGBRT	0.578	RFR
Industrial Product Design	138	102	1.512	BART	2.299	SGBRT	0.372	SGBRT
Information Systems	405	216	3.807	SRT	4.178	SGBRT	0.531	ERT
Information Systems Engineering	555	327	3.897	SGBRT	0.500	PNN	0.708	RFR
Information Technologies	41	33	1.195	PNN	2.646	SGBRT	0.416	SRT
Information and Communication Technologies	37	26	0.798	ERT	5.268	BART	0.527	SGBRT
Mathematics of modern technologies	129	76	2.179	SGBRT	7.978	RBD	0.398	RFR
Mechanical Engineering	550	331	4.260	SRT	7.571	BART	0.580	BART
Mechatronics and Robotics	180	137	0.000	SRT	4.758	RFR	0.516	SGBRT
Multimedia Design	1107	717	4.889	SGBRT	6.384	ERT	0.140	SGBRT
Organization Management	324	249	3.582	SGBRT	6.596	RBD	0.626	ERT
Production Engineering and Management	285	189	3.400	SGBRT	4.326	RFR	0.689	SGBRT

Table B1. (*Continued*).

Undergraduate study program	Number of students	Number of students without dropping out	Forecasted parameters					
			Number of failures to complete the courses		Number of retakes to complete the courses		Average university grade	
			Minimum RMSE	Used Data Science algorithm	Minimum RMSE	Used Data Science algorithm	Minimum RMSE	Used Data Science algorithm
Vilnius Gediminas Technical University								
Road, Railway and Urban Engineering	528	344	4.557	SGBRT	10.870	RFR	0.587	SGBRT
Security Systems Engineering	21	12	2.490	RBD	1.095	SGBRT	0.404	PNN
Software Engineering	595	481	1.118	SGBRT	0.000	RFR	0.709	RFR
Transport Engineering	1403	880	4.253	SGBRT	8.561	ERT	0.665	SGBRT
Transport Engineering Economics and Logistics	1273	795	3.050	SGBRT	7.966	RFR	0.709	RBD
Average (VGTU)			3.074		5.393		0.540	
Lithuanian Sports University								
Physical activity and public health	95	37	0.827	PNN	1.936	RBD	0.769	PNN
Physical activity and a healthy lifestyle	75	34	2.620	RFR	2.248	PNN	1.251	PNN
Physiotherapy	397	139	3.715	SRT	3.367	PNN	1.806	SGBRT
Physical Education	146	84	3.973	ERT	6.260	BART	1.622	PNN
Sports recreation and tourism	93	37	2.471	SRT	4.790	PNN	1.438	PNN
Sports and tourism management	104	48	2.517	RFR	4.215	PNN	1.916	SGBRT
Applied physical activity	34	8	0.756	PNN	1.323	PNN	0.966	PNN
Training systems	588	325	2.846	ERT	3.352	SRT	1.362	PNN
Average (LSU)			2.466		3.437		1.391	
Average (VGTU&LSU)			2.979		5.086		0.673	