Check for updates

Result Prediction Using Data Mining

Hasan Sarwar[1], Dipannoy Das Gupta[2](✉), Sanzida Mojib Luna[3], Nusrat Jahan Suhi[3], and Marzouka Tasnim[3]

[1] United International University, Dhaka, Bangladesh
[2] Edusoft Consultants Limited, Dhaka, Bangladesh
1405082.ddg@ugrad.cse.buet.ac.bd
[3] Military Institute of Science and Technology, Dhaka, Bangladesh

Abstract. Data mining is being used in various fields to dig out important information; it can be very effective in the field of education as well for gaining important information from a large dataset that can be used to improve the educational environment. This paper is focused on an approach consisting of several well-known and widely used algorithms on training data set to predict students' grade for a particular course based on his/her previous results. Further analysis has been carried out considering several errors and accuracy factors of the resulted data in comparison with the actual data.

Keywords: Education · Result prediction

1 Introduction

Prediction of results of a student based on previous academic results is a fairly investigated topic in the research literature. Usually, students' performances or results are predicted from the results of previous semesters and other academic attributes using data mining techniques [1, 3–5, 7]. Traditionally results are still considered to be an indicator of student performance, especially, in the case of graduate students. So there are still enough opportunities to work deep in this field that will help students to get better grades in exams. Many factors have been identified that affect the performance of a student. The factors are not only limited to academic fields but also encompass socio-economic variables, backgrounds, and cultural parameters [3, 6].

Few works on education, mostly higher education, are found in the literature in various considerations. Brijesh and Saurabh [1] have analyzed students' performance (End Semester Marks) by fixing some variables related to student performance like previous semester mark, class test grade, assignment, etc. Suhirman, Zain, Haruna, Tutut [2] have presented a review on data mining. It may be used for supporting the academic decisions in educational field. The paper has discussed recent works on data mining in educational field and given outlines over future researches. Edin and Mirza [3] have done prediction on student performance not only with academic variables but also with socio-economic factors. They have worked with the impact of family, income, gender, high school results as well as current course attributes on a student's performance. Course grade was the

K. Arai (Ed.): *Intelligent Computing*, LNNS 284, pp. 295–307, 2021.
https://doi.org/10.1007/978-3-030-80126-7_23

indicator of performance. Oyebade et al. [4] have used data mining for predicting the number of times a student will repeat a course. Neural network has been used as a data mining tool in this research. They have selected 30 attributes relating to the course itself, the teacher, and the particular student as predictor features. Behrouz, Deborah, Gerd, William [5] have presented a method to classify students in order to predict their final grades. The research has been executed depending on features extracted from logged data in an educational web-based system. The features used in this research are mostly connected with students' overall condition on academic performance including the number of corrected answers, time taken to answer a question, number of tries for homework, etc. Umesh and Pal [6] have shown a technique to find performer and underperformer of institutions using the Bayes Classification method. Here they have used caste, language, and class as attributes. Mueen, Zafar, Manzoor [7] have accomplished a study on the data set of two universities. This study has predicted students' academic performance based on general forum participation and academic attributes. Moreover, they have also shown a set of dominant predictor attributes in this performance prediction. All these works have tried to predict something regarding student performance or instructor performance.

Some university programs offer a fixed set of courses for a student for the next semester. Some other universities follow open credit type course offerings where a student is able to choose his or her desired courses from a list of offered courses. In the case of fixed selection, students generally have less options to choose his/her own course. In case of open credit, selection criteria for picking a course for the next semester vary among students. The interest for a particular topic or the intention to get easy marks may guide a student to select his or her next set of courses. Even in case of fixed setting, if a student could has been informed about the requirement of his effort to come up with a good result that would generate a positive contribution towards learning. Sometimes it gets too late to take proper preparation for a particular course and at the end of the semester; it is found that due to this course the result has turned unsatisfactory. No such studies are found that have analyzed the best sequence of suggestions. Another important criterion is that there are discipline-oriented courses that are tagged with one or more prerequisite courses. It means that a student can take a course if he or she has completed the required pre-requisite course(s). However, no such research is found where it shows that if there is any dependency of a course result on the already completed set of courses. Mostly, pre-requisite courses are set by using the experience of faculty members. Prerequisite courses are perceived as the foundation knowledge required to complete the main course. In this work, we have been able to show that student performance varies with an individual's achievement of grades of a set of courses in the earlier semesters. It means the grade of a course is affected by not only the pre-requisite course but also all the other courses that he or she has finished earlier on. To perform this task we have used data mining techniques and machine learning algorithms. This finding suggests that students will be better equipped in making a decision to select their courses for the next semester or put more effort on a particular course and be better able to come up with good results.

In this work, our goal is to make students enable to take more appropriate decisions with regard to emphasize on the right course during the start or throughout the semester. So we have tried to use established algorithms and found their effectiveness to predict students' performance based on the previous courses which will in a way help students to achieve better result.

2 Classification Model

Classification is one of the most fundamental tasks of data mining. Classification is the process of predicting the class of some given data points. As an instance, classification model predicts any kind of category or class such as whether a fruit will be considered as an apple or a banana. Here the attributes of the fruits, namely, size, color, taste are used to predict fruit class. This simple concept of classifying an entity in a specific group can be extended to any other entities, which is the beauty of classification models. There are several classification algorithms or models in the field of machine learning that might be used for prediction.

2.1 Naïve Bayes

A Naive Bayes classifier is a probabilistic machine learning model. This classifier acts based on Bayes theorem. The assumption made here is that the predictors/features don't depend on each other. That is presence of one particular feature does not affect the other. Hence it is called naive.

2.2 J-48

Decision tree is another type of classification. There are two approaches of implementing a decision tree-based classifier. Univariate decision tree is one of them. Splitting is performed by using one attribute at internal nodes in this strategy. J48 algorithm is used to build such tree [8]. In this procedure, the first step is the construction of the tree. Second step is all about information gain. Third step consists of pruning.

2.3 K*

K* is a Heuristic Search Algorithm for Finding the k Shortest Paths. In the execution of K* algorithm, A* algorithm is used to search in graph G and Dijkstra to search in P(G) [9]. Here P(G) is a directed weighted graph formed form G. K* does not require the graph to be obviously available. Parts of the graph are generated when it is necessary. Another advantage is found due to the heuristic function. The function guides K* to perform better. These are the two advantages of K* over K shortest path [9].

2.4 Random Tree

Random tree is a set of large number of individual decision tree. All trees act like an ensemble. Random tree is also called random forest. Each individual tree in the random forest comes out with a class prediction. The tree which has most voted class becomes model's prediction [10].

3 Performance Measure Metrics

Some performance measure metrics are available to evaluate the performance of classification models.

3.1 Kappa Statistics

The kappa statistic is used to control only those instances that may have been correctly classified by chance. This can be calculated using both the observed (total) accuracy and the random accuracy.

3.2 Root Mean Square Error

In measuring the error of a model in predicting data, Root Mean Square Error (RMSE) is a standard approach. It actually indicates the deviation of predicted data from observed data. From the view of heuristic, RMSE can be illustrated as the difference between observed and predicted quantity. The concentration of data around the line of best fit can be deduced from RMSE [11].

3.3 Relative Absolute Error

Relative Absolute Error (RAE) is another procedure for measuring the performance of a classifier model. It is calculated with the following formula [12]:

$$RAE_i = \frac{\sum_{j=1}^{n} \left| P_j - T_j \right|}{\sum_{j=1}^{n} \left| T_j - T \right|} \tag{1}$$

Here P_j is the value, predicted by an individual program for jth sample case out of n sample cases; the target value is expressed with T_j for sample case j; and T is calculated by the formula [13]:

$$T = \frac{1}{n} \sum_{j=1}^{n} T_j \tag{2}$$

3.4 Root Relative Squared Error

The root relative squared error (RRSE) functions like Relative Absolute Error. More specifically, numerator is the total squared error and denominator is the total squared error of simple predictor. RRSE can be calculated with the following formula:

$$RRSE_i = \frac{\sqrt{\sum_{j=1}^{n} \left(P_j - T_j \right)^2}}{\sqrt{\sum_{j=1}^{n} \left(T_j - T \right)^2}} \tag{3}$$

Here P_j is the value, predicted by an individual program for jth sample case out of n sample cases; the target value is expressed with T_j for sample case j; and T is calculated by the formula [13]:

$$T = \frac{1}{n} \sum_{j=1}^{n} T_j \tag{4}$$

3.5 Info Gain

Information gain is an important quantity. It is found by calculating a value for a feature. More precisely, subtracting the entropy of the distribution after split from the entropy of the distribution before split, info gain is calculated. The largest information gain indicates smallest entropy.

3.6 Relief Attribute

Relief Attribute measures the utility of an attribute. For this purpose, repeated sampling of an instance is needed. Moreover value consideration of the given attribute for the nearest instance is required [14]. Both discrete and continuous class data can be evaluated with it [15].

3.7 False Positive and False Negative

False positive can be defined as receiving a positive result for an experiment, while negative result is expected. It's also being called as a false alarm or false positive error [16]. From the viewpoint of classification model, a false positive is a result where the model predicts the positive class incorrectly. And for a false negative outcome, classifier inaccurately predicts negative label or class.

3.8 Confusion Matrix

A confusion matrix is a table that is very useful to demonstrate the performance of a classification model. This special kind of prediction table is displayed in two dimensions. They are actual and predicted. With them, identical sets of "classes" also exist in both dimensions [17].

4 Data Collection

In this work, sample data has been collected from a sample university. As we have prioritized the courses, it was essential to conduct the collection process over the courses of a certain program. Moreover, consideration has been taken for multiple batches of students to keep the data size large enough for the convenience of patterns. However, there were a good amount of pitfalls like outliers, missing values for dealing with this large number of datasets.

4.1 Data Migration

For starting data preprocessing, we had to import the data set from "xls" sheet to the database server. For this purpose, we collected data from our sample university in "xls" format. Then the data has been migrated into the database server using the import feature of the database server.

4.2 Data Aggregation

After migration, we performed data aggregation. As the raw data was in the different sheets in xls file, they were imported into different tables. For example, student-course mapping, student-semester mapping, student-program mapping were contained in separate tables. All the required information for our training was gathered in a single table. Here we have used the typical database aggregate functions to accomplish this task. Moreover, the concept of PIVOT has also come in handy during the execution of row-column interchange.

4.3 Data Cleaning

After performing aggregation, we found lots of null grades against the courses which were necessary to remove for ensuring better performance of our classifiers. A student found with any null grade has been removed from the dataset. Moreover, there were many instances where a student took a course multiple times for improvement. Here we have applied a searching algorithm for finding out the best grade. After that, the remaining entries for that course have been deleted.

4.4 Outlier Detection and Replacement

A few numbers of outliers have been found in the dataset. For instance, some students were absent for a particular course while some students had drop course issues. We have tried to detect such anomalies in the dataset. As one of our main objectives was to keep the data size large, our task was not limited to outlier detection. Moreover, we had replaced those outliers with their actual grade secured in the subsequent semesters if it was available.

4.5 Attribute Selection

As we have stated earlier, we are analyzing previous semester course works, our selected attributes were courses. We have processed the raw data of the past three semesters to turn it into a structured one. In this process, a total of 16 courses of the previous three semesters have been selected to make the prediction of fourth-semester courses' grade and so the prediction of fourth-semester result.

4.6 Export Data

After the completion of all data preprocessing tasks in the database server, we got our structured, clean, and expected data set to feed into the classifiers. Here for the purpose of grade prediction, we have collected five data sets. Each data set consists of 18 variables. Among them, one is Student ID. The remaining 17 variables are different courses. Among them 16 courses are training attributes that are from previous semesters. And one course is a predicted variable that is registered for the new semester. We will predict this new semester course grade. After collecting data, we divided them into two parts. We kept 90% of them for training purposes and 10% of them for testing and analyzing the accuracy. We have used J48, K-Star, Naive Bayes, and Random Tree for training and testing purposes. After conducting the training, we have applied them to the test data set and found out the predicted results from each algorithm. The predicted results were the Grade of each course of the new semester.

4.7 Input Parameter

For training phase, we have grades of sixteen different courses, student ID and the grade of the course we need as result of prediction.

Table 1. Sample of input parameters for training phase

Student ID	C1	C2	C3	C4	…………	C16	C27
S1	A	B	A−	A+	…………	B+	A+
S2	B−	A+	C	A	…………	C	C
S3	A−	B−	A+	C	…………	C+	B+
S4	A	A+	B	C	…………	B−	C+

In Table 1, we can see a small sample of training dataset that consists of eighteen different attributes. Those are Student ID, grades of 16 different courses and the grades of course that we want to predict. Here last course C27 is the predicted course. Dotted portion indicates more courses. We have used other four different data sets for predicting grades of other four courses, with same attributes. The Student ID is consistent for all the data sets.

Table 2. Sample of matched instances

Student ID	Actual grade	Naïve Bayes
S1	A−	A
S2	C+	C

4.8 Output Parameter

After the training phase, testing has been done on 10% data. Then we have found predicted grades of each course using J48, Naive Bayes, K-Star and Random Tree algorithms. In Fig. 1, we can see a sample output format and it is the prediction of grade using one of the algorithms. For this sample, it is J48. As we had data set for five different courses, we gathered predicted results of those five courses for each algorithm.

After we got predicted grades for all the five courses we were aiming for, we calculated individual student's grade using a simple python program. For this phase, we used Student ID and the predicted grades of the courses as input.

After calculating grade, we have compared it with actual grade for finding out the percentage of matched instances.

In the Table 2, we can see three attributes student ID, their actual Grade and the Grade that was calculated from the predicted courses' Grade using Naive Bayes algorithm.

inst#	actual	predicted	error	prediction
1	2:'C	' 3:B+	+	1
2	4:'C+	' 3:B+	+	0.5
3	6:'A	' 9:D	+	1
4	5:'B	' 4:C+	+	0.5
5	5:'B	' 9:D	+	1
6	1:'A+	' 4:C+	+	1
7	1:'A+	' 1:A+		0.359
8	9:'D	' 7:C-	+	1
9	6:'A	' 1:A+	+	0.341
10	8:'A-	' 1:A+	+	0.341
11	5:'B	' 5:B		0.4
12	9:'D	' 2:C	+	0.333
13	1:'A+	' 1:A+		0.632
14	4:'C+	' 3:B+	+	0.5
15	3:'B+	' 6:A	+	1
16	1:'A+	' 8:A-	+	0.429
17	4:'C+	' 7:C-	+	0.4

Fig. 1. Sample of output parameters after prediction of individual course grade

5 Result and Analysis

In this study, we have worked with the data taken from four semesters. We have taken the courses of the fourth semester as our predicted attributes. There are 11 possible class labels both for the predicted and predictor attributes. All the labels are composed of different existing grades namely, A+, A, A−, B+, B, B−, C+, C, C−, D, F. We have to repeat the training and testing process for five times due to having five courses in the fourth semester. After each process, the individual grade of a particular course of the fourth semester is predicted. Finally, from predicted grades of five courses, we have calculated semester grade for the individual student and compared with actual grade.

The whole process is executed under four classification models which we have stated earlier. WEKA 4.8, [18] has been used as our testbed. Cross-validation of 10 folds has been used while training the data set. Here 10 folds means each time we have taken 10 instances from the data set and applied algorithms into them for training and testing the dataset. Same cross-validation has been applied for each of the classification algorithms.

Our preliminary goal was to compare the predicted grade with the actual grade of the fourth semester. As we completed the processes for all algorithms, we did the comparison. Here we have set the definition of success in two different scales. One is 100% matching or matching without error. Another one is matching with 10% error.10% error means actual Grade is A-, but predicted Grade is A or B+. In the order of Grade, A- comes after B+ and before A. Both grades are 1 unit distant from A-. This 1 unit far prediction has been considered as matching with 10% error.

From Tables 3, 4, 5 and 6, we can see that if we use Naive Bayes and Random tree algorithm in 0% error condition, it can detect 25% of students' grade accurately, where K-star and J48 have lower success rate here. But when we do the same procedure with considering 10% error then the success rate increases. Here Naive Bayes and J48 algorithm shows 61% and 53% success rate which are much better than the previous result. Better result with the increase of error rate is quite expected. From the above discussion, we can conclude that Naive Bayes is showing overall better performance than the remaining three.

Now we will analyze our above results with performance evaluation metrics.

Table 3. GPA comparison result with Naïve Bayes

Error rate	Total instance	Matched instance	Success rate
0%	54	14	25.92%
10%	54	33	61.12%

Table 4. GPA comparison result with K-Star

Error rate	Total instance	Matched instance	Success rate
0%	54	9	16.67%
10%	54	25	46.1%

Table 5. GPA comparison result with J48

Error rate	Total instance	Matched instance	Success rate
0%	54	10	18.51%
10%	54	29	53.70%

5.1 0% Error

With the 0% error, we have tested a total of 54 students' grades. Of these 54 students, Naïve Bayes could match with 14 students' grades. J48 could match with10 students' grades. K Star could match with 9 students' grades. And Random Tree could match with 13 students' grades.

Table 6. GPA comparison result with Random Tree

Error rate	Total instance	Matched instance	Success rate
0%	54	13	24.07%
10%	54	26	48.4%

From Table 7, it is observed that Naïve Bayes is showing better performance from the perspective of 0% error. For Naïve Bayes three measurements of error (RMSE, RAE, RRSE) show less amount of error than the other three. Second to Naïve Bayes, Random Tree performs better. J48 and K* are ranked third and fourth respectively. Tables 8, 9 also reveal the best performance of Naïve Bayes. From the viewpoint of accuracy, precision, and recall Naïve Bayes dominates the other three. So Naïve Bayes is the best suit for this condition followed by Random Tree, J48, and K*.

Table 7. Performance measures of classifiers

Model	Kappa statistic	Root mean squared error	Relative absolute error	Root relative squared error
N. Bayes	0.75	5.44	40	74.07%
J48	0.83	5.99	44	81.48%
K*	0.81	6.124	45	83.33%
Rand. Tree	0.77	5.58	41	75.93%

Table 8. Performance measures of classifiers

Model	False positive FP	False negative FN	True positive TP	True negative TN	P = (TP + FP)	N = (TN + FN)
N. Bayes	40	14	14	40	54	54
J48	44	10	10	40	54	54
K*	45	9	9	45	54	54
Rand. Tree	41	13	13	41	54	54

Table 9. Performance measures of classifiers

Model	Accuracy $\frac{TP+TN}{P+N}$	Precision $\frac{TP}{TP+FP}$	Recall $\frac{TP}{P}$	Specificity $\frac{TN}{N}$
N. Bayes	0.259	0.26	0.259	0.741
J48	0.185	0.19	0.185	0.814
K*	0.167	0.17	0.167	0.833
Rand. Tree	0.241	0.24	0.241	0.759

5.2 10% Error

With the 10% error, we have tested the same total of previously taken 54 students' grades. Among 54, Naive Bayes could match with 33 students' grades. J48 could match with 29 students' grades. K Star could match with 25 students' grades. Random Tree could match with 26 students' grades.

From Table 10, it is observed that Naïve Bayes also performs better in the 10% error condition. For Naïve Bayes three measurements of error (RMSE, RAE, RRSE) show less amount of error than the other three. Second to Naïve Bayes, J48 performs better. Random Tree and K*are ranked third and fourth respectively. There is a swap in performance between Random Tree and J48 compared to the previous scenario of 0% error. However, Random Forest will stay ahead considering average error rates and success rate.

Table 10. Performance measures of classifiers

Model	Kappa statistic	Root mean squared error	Relative absolute error	Root relative squared error
N. Bayes	0.396	2.85	21	38.89%
J48	0.47	3.4	25	46.3%
K*	0.56	3.9	29	53.7%
Rand. Tree	0.53	3.8	28	51.85%

From Tables 11 and 12, it is also evident that Naïve Bayes is the best among the four models. From the view of accuracy, precision and recall, Naïve Bayes outperforms rest of the three classifiers. So Naïve Bayes can be chosen convincingly for this condition too. Then J48, Random Forest, K* will come respectively.

Though we have finally predicted the semester grade, it's not predicted in a direct manner. After predicting individual course grade, then the semester grade has been calculated. So under the abstraction of the semester grade, we have actually unfolded the new semester courses' grades. Our research goal is defined for two systems as we stated earlier. For the open credit system, our research can provide a set of courses for a

Table 11. Performance measures of classifiers

Model	False Positive FP	False Negative FN	True Positive TP	True Negative TN	P = (TP + FP)	N = (TN + FN)
N. Bayes	21	33	33	21	66	42
J48	25	29	29	25	58	50
K*	29	25	25	29	50	58
Rand. Tree	28	26	26	28	52	56

Table 12. Performance measures of classifiers

Model	Accuracy $\frac{TP+TN}{P+N}$	Precision $\frac{TP}{TP+FP}$	Recall $\frac{TP}{P}$	Specificity $\frac{TN}{N}$
N. Bayes	0.61	0.611	0.5	0.5
J48	0.53	0.537	0.5	0.5
K*	0.46	0.462	0.5	0.5
Rand. Tree	0.48	0.481	0.5	0.5

student which will be the best suit for him for the new semester based on the performance in his previous semester course works. If it is fixed credit system, our research will present a clear concept to the individual student regarding the preparation of the courses. And here is also the main factors are previous semester courses.

6 Conclusion

We have predicted the performance of students based on their previous academic records. Here a new idea has been tested and found to be meaningful as long as student result is important. We have shown that even there are little apparent dependencies of the result of one course on the previously completed courses; it is possible to predict the grade of the new course on the basis of the results of all the grades of previously taken courses. This result will help to guide a student to decide and dedicate his effort on a particular course and make him able to learn better. Our next aim is to train a large number of data in order to make the prediction more accurate along with considering some more features. There are scopes to consider other attributes to predict student grades. Moreover, the percentage of contribution of each predictor course on the outcome will also be determined in our future work.

References

1. Baradwaj, B.K., Pal, S.: Mining educational data to analyze students' performance. arXiv preprint arXiv:1201.3417 (2012)
2. Zain, J.M., Herawan, T.: Data mining for education decision support: a review. Int. J. Emerg. Technol. Learn. **9**(6), 4–19 (2014)
3. Osmanbegovic, E., Suljic, M.: Data mining approach for predicting student performance. Econ. Rev.: J. Econ. Bus. **10**(1), 3–12 (2012)
4. Oyedotun, O.K., Tackie, S.N., Olaniyi, E.O., Khashman, A.: Data mining of students' performance: Turkish students as a case study. Int. J. Intell. Syst. Appl. **7**(9), 20 (2015)
5. Minaei-Bidgoli, B., Kashy, D.A., Kortemeyer, G., Punch, W.F.: Predicting student performance: an application of data mining methods with an educational web-based system. In: 33rd Annual Frontiers in Education, 2003, FIE 2003, vol. 1, pp. T2A–13. IEEE (2003)
6. Pandey, U.K., Pal, S.: Data Mining: A prediction of performer or underperformer using classification. arXiv preprint arXiv:1104.4163 (2011)
7. Mueen, A., Zafar, B., Manzoor, U.: Modeling and predicting students' academic performance using data mining techniques. Int. J. Mod. Educ. Comput. Sci. **8**(11), 36 (2016)
8. Bhargava, N., Sharma, G., Bhargava, R., Mathuria, M.: Decision tree analysis on j48 algorithm for data mining. Proc. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **3**(6), 1114–1119 (2013)
9. Aljazzar, H., Leue, S.: K∗: a heuristic search algorithm for finding the k shortest paths. Artif. Intell. **175**(18), 2129–2154 (2011)
10. Yiu, T.: Towards data science-understanding random forest (2019). https://towardsdatascience.com/understanding-random-forest-58381e0602d2
11. Glen, S.: "RMSE: Root Mean Square Error" From StatisticsHowTo.com: Elementary Statistics for the rest of us! https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/
12. Glen, S.: "Relative Absolute Error" From StatisticsHowTo.com: Elementary Statistics for the rest of us! https://www.statisticshowto.com/relative-absolute-error/
13. "Analyzing genexprotools models statistically," https://www.gepsoft.com/gxpt4kb/Chapter10/Section2/SS15.html
14. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: Sleeman, D.H., Edwards, P. (eds.) Ninth International Workshop on Machine Learning, pp. 249–256. Morgan Kaufmann (1992)
15. Kononenko, I.: Estimating attributes: analysis and extensions of relief. In: Bergadano, F., Raedt, L.D. (eds.) European Conference on Machine Learning, pp. 171–182. Springer (1994)
16. Hirai, M., Kawai, S.: False positive and negative results in allen test. J. Cardiovasc. Surg. **21**(3), 353–360 (1980)
17. Lewis, H., Brown, M.: A generalized confusion matrix for assessing area estimates from remotely sensed data. Int. J. Remote Sens. **22**(16), 3223–3235 (2001)
18. "WEKA The workbench for machine learning". https://www.cs.waikato.ac.nz/ml/weka/