

УДК 37

DOI: 10.34670/AR.2024.96.13.034

Методы машинного обучения при оценке успеваемости студентов

Кричевский Михаил Лейзерович

Доктор технических наук, профессор,
профессор кафедры бизнес-информатики и менеджмента,
Санкт-Петербургский государственный университет
аэрокосмического приборостроения,
190121, Российская Федерация, Санкт-Петербург, ул. Большая Морская, 67;
e-mail: mkrichevsky@mail.ru

Мартынова Юлия Анатольевна

Кандидат экономических наук, доцент,
доцент кафедры бизнес-информатики и менеджмента,
Санкт-Петербургский государственный университет
аэрокосмического приборостроения,
190121, Российская Федерация, Санкт-Петербург, ул. Большая Морская, 67;
e-mail: juli_ko@list.ru

Аннотация

Рассмотрено применение методов машинного обучения в области оценки успеваемости студентов. Выделены основные признаки, от которых зависит успеваемость, и продемонстрировано решение задачи по ее оценке методами машинного обучения. Проведен анализ использования различных подходов в при формировании оценки. Дано краткое описание методов машинного обучения, относящихся, прежде всего, к разряду контролируемого, которые способны найти решение задачи в указанной области. Приведен практический пример решения, выполненного в программном продукте *MatLab* с получением конкретных результатов. Указаны пути дальнейших исследований по привлечению методов искусственного интеллекта в задачу оценки успеваемости.

Для цитирования в научных исследованиях

Кричевский М.Л., Мартынова Ю.А. Методы машинного обучения при оценке успеваемости студентов // Педагогический журнал. 2024. Т. 14. № 1А. С. 393-399. DOI: 10.34670/AR.2024.96.13.034

Ключевые слова

Успеваемость студентов, искусственный интеллект, машинное обучение, формирование оценки.

Введение

Искусственный интеллект – чрезвычайно широкое и трудно формализуемое понятие, которое связано с представлением знаний, их извлечением и последующими манипуляциями над ними. ИИ проникает в различные сферы деятельности человека, дополняя и даже заменяя его естественный интеллект. ИИ возник из информатики и компьютерных наук, но на эту область оказали сильное влияние и другие дисциплины, в частности философия, когнитивная наука, нейробиология и экономика. Учитывая междисциплинарный характер этой области, среди исследователей ИИ мало согласия об общем определении и понимании ИИ.

Внедрение методов ИИ в программы обучения соответствует приоритетному проекту России «Современная цифровая образовательная среда в РФ». Для цифровой экономики нужны компетентные кадры, а для их подготовки необходимо должным образом модернизировать систему образования и профессиональной подготовки, привести образовательные программы в соответствие с нуждами цифровой экономики. Искусственный интеллект – основа цифровой экономики.

Основное содержание

Идея ИИ появилась первой – самая большая область; затем машинное обучение (МО), которое является подмножеством ИИ; наконец, глубокое обучение (ГО), вызвавшее сегодняшнее стремительное развитие ИИ, определяется как часть искусственного интеллекта и машинного обучения.

Трудно дать точное определение ИИ даже экспертам в предметной области по двум основным причинам [Xieling Chen, Naoran Xie, Di Zou, Gwo-Jen Hwang, 2020]. Во-первых, то, что включает в себя ИИ, постоянно развивается. Во-вторых, ИИ сам по себе является междисциплинарной областью с исследователями и экспертами из различных областей, например нейробиологии, психологии и лингвистики, которые постоянно вносят собственный вклад, восприятие, знания и терминологию.

Применение методов МО данных в сфере образования привлекает большое внимание в последние годы. В работе [Fernandes et al., 2019] приводится модель оценок успеваемости, полученных в результате занятий в семестре, с учетом демографических характеристик учащихся. Здесь успеваемость студентов прогнозировалась с помощью классификационных моделей, основанных на *Gradient Boosting Machine*. В [Yağcı, 2022] предлагается модель, основанная на алгоритмах машинного обучения, для прогнозирования итоговых экзаменационных оценок студентов бакалавриата, принимая в качестве исходных данных оценки за промежуточные экзамены. Для прогнозирования оценок студентов на итоговых экзаменах были рассчитаны и сравнены показатели алгоритмов случайных лесов, ближайшего соседа, опорных векторов, логистической регрессии, наивного Байеса и k-ближайшего соседа, которые входят в число алгоритмов машинного обучения. Модель линейной регрессии для задачи прогнозирования оценок учащихся с помощью машинного обучения с использованием Python предлагается в [Student Grades Prediction with Machine Learning, www]. В работе [Бантикова, Новикова, 2023] описываются модели интеллектуального анализа образовательных данных, позволяющие провести сравнительный анализ обучающихся и прогнозирование уровня освоения образовательной программы. Такой анализ проводится с учетом таких факторов, как суммарный балл вступительных испытаний, средний балл академической успеваемости, основа

и форма обучения, курс, уровень подготовки, пол и возраст студента. Даже такой краткий анализ литературных источников показывает серьезное внимание к рассматриваемой теме по оценке успеваемости студентов. Вследствие этого наша работа добавляет в исследовательский ландшафт рассматриваемой темы еще один штрих. Цель работы заключается в демонстрации методов машинного обучения, реализованных в Matlab, при получении зачетной оценки по изучаемой дисциплине.

Машинное обучение – это подмножество алгоритмов ИИ, которые пытаются обнаружить и вывести скрытые законы в соответствии с историческими данными, а затем предсказать или классифицировать неразмеченные наборы данных [Alpaydin, 2010].

Рассмотрим несколько алгоритмов классификации, которые в дальнейшем используем для решения задачи.

Дерево решений (Decision Tree – DT) – это типичный и простой для понимания алгоритм машинного обучения, который представляет собой отношение отображения между различными переменными и может использоваться для регрессии или классификации. Алгоритм *DT* использует древовидную структуру и слой вывода для принятия окончательного решения о результатах моделирования. Процесс применения *DT* обычно включает в себя выбор функций, создание и сокращение структуры дерева. Структура *DT* обычно состоит из трех элементов: корневого узла, внутренних узлов (атрибуты признаков) и конечных узлов (классификация на основе решений). В начале обучения данные, содержащиеся в базе примеров, вводятся в корневой узел. После этого алгоритм *DT* ищет, какое значение переменной разделяет набор исходных данных таким образом, чтобы минимизировать связанную с этим ошибку. К основным преимуществам *DT* относятся простая структура алгоритма, высокая интерпретируемость, простота реализации. Однако *DT* имеет свои недостатки, такие как возможность переобучения и сложная сгенерированная структура *DT*. Кроме того, единственная модель *DT* в сильной степени зависит от помехи в исходных данных.

Линейный дискриминантный анализ (Linear Discriminant – LD) – широко используемый метод математического моделирования, представляющий собой обобщение линейного дискриминанта Фишера, предложенное им еще в 1936 г. Однако, несмотря на долгую историю развития этих методов, эти модели продолжают активно использоваться и сегодня в различных сферах деятельности [Ezghazi, Zahi, Zekoua, 2017]. *LD* определяет оптимальную линейную комбинацию, которая может разделить данные на разные кластеры путем проецирования выборок данных в пространство более низкого размерения. В отличие от принципа главных компонент, *LD* – метод контролируемого обучения, хотя его производительность в некоторых случаях может уступать методу главных компонент.

LD предполагает, что данные нормально распределены, классы имеют идентичные ковариация, признаки независимы и одинаково распределены. Для проведения классификации *LD* измеряет расстояние между прогнозируемыми средними значениями и использует матрицу рассеяния, чтобы максимизировать отношение расстояния выборки между классами к расстоянию выборки внутри класса.

Метод опорных векторов (Support Vector Machine – SVM) – это контролируемый алгоритм машинного обучения, который разделяет неразмеченные наборы данных на две категории. Процесс применения *SVM* обычно включает два этапа: во-первых, модель *SVM* может быть построена путем обучения размеченного набора данных (т.е. результат классификации каждой точки данных известен). После этого обученная модель *SVM* становится бинарным линейным классификатором. Во-вторых, обученную модель *SVM* можно использовать для классификации

неразмеченных наборов данных, которая сопоставляет новый экземпляр с тем же пространством и прогнозирует его категорию. Преимущество *SVM* по сравнению с другими алгоритмами заключается в быстром процессе классификации и более высокой точности при ограниченном количестве выборок. В дополнение к классификации линейных наборов данных метод *SVM* также может использоваться для классификации нелинейных наборов данных посредством изменения размерности с помощью определенных ядер.

Нейронная сеть (Neural Network – NN) – математическая модель, имитирующая функции биологической нейронной сети. Обучившись на примерах, сеть может выполнять сложные задачи нелинейного моделирования и классифицировать объекты. Каждый нейрон (узел) сети содержит лишь небольшую часть информации, которая распределяется между различными нейронами через их взаимосвязи и передается посредством электрических импульсов. Точно так же алгоритм *NN* опирается на взаимосвязанные нейроны, каждый из которых хранит только малый фрагмент глобальной информации. Узлы разделены на слои, которые можно считать как входные, выходные или скрытые. Количество узлов во входном слое равно количеству входов модели, число узлов выходного слоя определяется количеством классов (в задаче классификации). Скрытый слой (слои) состоит из n узлов, а число скрытых слоев и узлов для каждого слоя определяются гиперпараметрами, которые необходимо оптимизировать для конкретного случая.

Здесь рассматривается смоделированный набор данных (*Toy Dataset*), поскольку в машинном обучении важно научиться правильно применять игрушечные наборы данных, так как обучение алгоритма на реальных данных сопряжено с трудностями и может закончиться неудачей [Рамсундар, Заде, 2019]. Разыгранные наборы данных играют решающую роль для понимания работы алгоритмов. При наличии простой синтетической выборки данных достаточно просто оценить, обучился алгоритм нужному правилу или нет. На реальных данных получить такую оценку сложно. В качестве входных признаков используем такой набор: посещение занятий, участие в семинарах, выступления на конференциях, итоговый проект по дисциплине. Все входные признаки оцениваются по десятибалльной шкале. Выходная переменная имеет два дискретных значения: зачет – незачет.

Обучение на разыгранных данных с использованием вышеприведенных методов машинного обучения привело к следующим результатам (рис. 1).

Models		
Sort by:	Model Number	
☆ 1	Tree	Accuracy (Validation): 87.5%
	Last change: Fine Tree	4/4 features
☆ 2	Linear Discrimi...	Accuracy (Validation): 92.5%
	Last change: Linear Discriminant	4/4 features
☆ 3	SVM	Accuracy (Validation): 92.5%
	Last change: Linear SVM	4/4 features
☆ 4	Neural Network	Accuracy (Validation): 80.0%
	Last change: Narrow Neural Network	4/4 features

Рисунок 1 – Результаты классификации

Как видно из рис. 1, лучшую точность классификации, равную 92,5%, показали линейный дискриминантный анализ и метод опорных векторов. Последний метод используем в дальнейшем для оценки успеваемости.

На рис. 2 приведена кривая операционной характеристики, которая показывает долю верных положительных классификаций (TPR) в функции от доли ложных положительных классификаций ($false\ positive\ rate - FPR$) для выбранного классификатора.

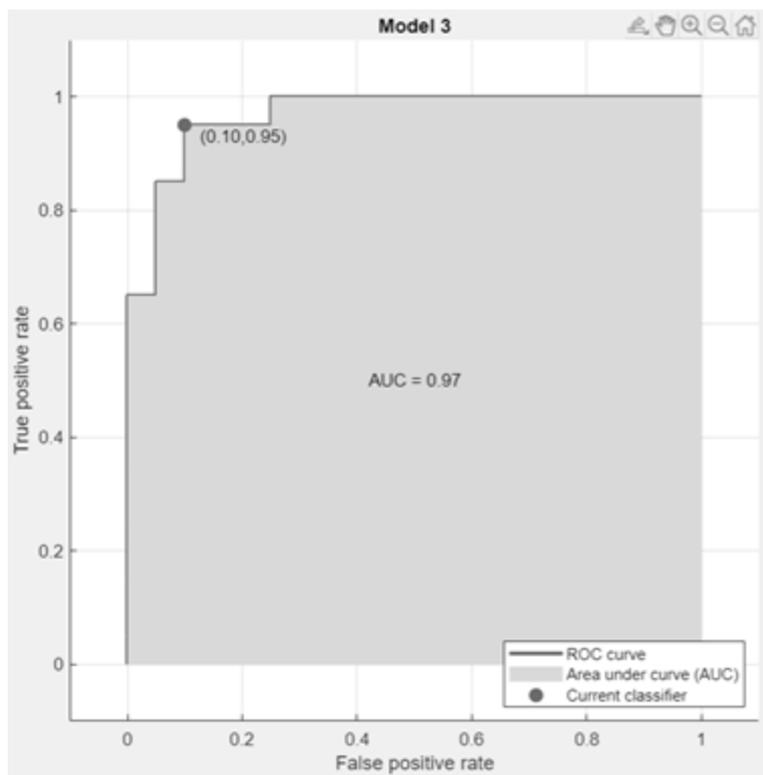


Рисунок 2 – Кривая операционной характеристики

Маркер показывает значения FPR и TPR для выбранного в данный момент классификатора. Параметр AUC в данном случае практически равен 0,97, что говорит о высоком качестве классификатора.

Сформированная структура `trainModel` используется для прогнозирования оценок с использованием новых данных. В качестве примера ниже приведен результат классификации нового вектора, характеризующего учебную деятельность студента:

```
Z1 = [5.1 4.2 3.3 2.4];
>> yfit = trainedModel.predictFcn(Z1)
yfit = 2
```

Как видно, студент с таким отношением к учебе зачета не получит (класс 2 – незачет).

Заключение

Приведенный подход может быть расширен путем как увеличения перечня входных признаков и количества выходных градаций оценок (отлично, хорошо, удовлетворительно, неудовлетворительно), так и использования реальных оценок по результатам сессии.

Библиография

1. Алексейчева Е.Ю. Гуманизация образования: антропоцентризм и видимое обучение. В сборнике: Гуманизация образования: принципиальные позиции и положения. Сборник статей. Ярославль, 2021. С. 6-16.
2. Алексейчева Е.Ю. Новые тренды в управлении образовательными системами // Цифровая гуманитаристика: человек в «прозрачном» обществе: Коллективная монография. М.: Книгодел, 2021. С. 68-97.
3. Боброва В.В., Бантикова О.И., Новикова В.А. Моделирование академической успеваемости студентов на основе интеллектуального анализа образовательных данных// Экономический анализ: теория и практика. 2023. № 2(533). С. 235-253.
4. Рамсундар Б., Заде Р. TensorFlow для глубокого обучения. СПб.: БХВ-Петербург, 2019. 256 с.
5. Alpaydin E. Introduction to machine learning. Massachusetts Institute of Technology, 2010. 579 p.
6. Ezghazi S., Zahi A., Zekoua K. A new nearest neighbor classification method based on fuzzy set theory and aggregation operators // Expert Systems with Applications. 2017. Vol. 80. No. 1. P. 58-74.
7. Fernandes E. et al. Educational data mining: Predictive analysis of academic performance of public school students in the capital of Brazil // Journal of Business Research. 2019. No. 94. P. 335-343.
8. Student Grades Prediction with Machine Learning // Aman Kharwal. URL: thecleverprogrammer.com.
9. Xieling Chen, Haoran Xie, Di Zou, Gwo-Jen Hwang. Application and theory gaps during the rise of Artificial Intelligence in Education // Computers and Education: Artificial Intelligence. 2020. No. 1.
10. Yağcı M. Educational data mining: prediction of students' academic performance using machine learning algorithms // Smart Learn. Environ. 2022. No. 9 (11).

Machine learning methods for evaluating student performance

Mikhail L. Krichevskii

Doctor of Technical Sciences, Professor,
Professor of the Department of business informatics and management,
Saint Petersburg State University of Aerospace Instrumentation,
190121, 67 Bol'shaya Morskaya str., Saint Petersburg, Russian Federation;
e-mail: mkrichevsky@mail.ru

Yuliya A. Martynova

PhD in Economics, Associate Professor,
Associate Professor of the Department of business informatics and management,
Saint Petersburg State University of Aerospace Instrumentation,
190121, 67 Bol'shaya Morskaya str., Saint Petersburg, Russian Federation;
e-mail: juli_ko@list.ru

Abstract

The authors consider the application of machine learning methods in the field of student performance assessment. The main features on which academic performance depends are highlighted, and the solution of the problem of its assessment by machine learning methods is demonstrated. The analysis of the use of various approaches in the formation of the assessment is carried out. A brief description of machine learning methods is given, primarily related to the category of controlled learning, which can find a solution to the problem in this area. A practical example of a solution implemented in the MatLab software product with specific results is given. The ways of further research on the involvement of artificial intelligence methods in the task of assessing academic performance are indicated.

For citation

Krichevskii M.L., Martynova Yu.A. (2024) Metody mashinnogo obucheniya pri otsenke uspevaemosti studentov [Machine learning methods for evaluating student performance]. *Pedagogicheskii zhurnal* [Pedagogical Journal], 14 (1A), pp. 393-399. DOI: 10.34670/AR.2024.96.13.034

Keywords

Student academic performance, artificial intelligence, machine learning, assessment formation.

References

1. Alekseicheva E.Yu. (2021) Gumanizaciya obrazovaniya: antropocentrizm i vidimoe obuchenie. [The humanization of education: Anthropocentrism and visible learning] V sbornike: Gumanizaciya obrazovaniya: principial'nye pozicii i polozheniya. Sbornik statej. YAroslav' [In the collection: Humanization of education: fundamental positions and positions. Collection of articles. Yaroslavl], pp. 6-16.
2. Alekseicheva E.Yu. (2021) Novye trendy v upravlenii obrazovatel'nymi sistemami [New trends in the management of educational systems] Cifrovaya gumanitaristika: chelovek v «prozrachnom» obshchestve: Kollektivnaya monografiya. M.: Knigodel [Digital humanities: a person in a "transparent" society: Collective monograph. M.: Knigodel], pp. 68-97.
3. Alpaydin E. (2010) *Introduction to machine learning*. Massachusetts Institute of Technology.
4. Bobrova V.V., Bantikova O.I., Novikova V.A. (2023) Modelirovanie akademicheskoi uspevaemosti studentov na osnove intellektual'nogo analiza obrazovatel'nykh dannykh [Modeling of students' academic performance based on intellectual analysis of educational data]. *Ekonomicheskii analiz: teoriya i praktika*, 2(533), pp. 235-253.
5. Ezghazi S., Zahi A., Zekoua K. (2017) A new nearest neighbor classification method based on fuzzy set theory and aggregation operators. *Expert Systems with Applications*, 80 (1), pp. 58-74.
6. Fernandes E. et al. (2019) Educational data mining: Predictive analysis of academic performance of public school students in the capital of Brazil. *Journal of Business Research*, 94, pp. 335-343.
7. Ramsundar B., Zade R. (2019) *TensorFlow dlya glubokogo obucheniya* [TensorFlow for deep learning]. Saint Petersburg: BKhV-Peterburg Moscow
8. Student Grades Prediction with Machine Learning // *Aman Kharwal*. Available at: thecleverprogrammer.com.
9. Xieling Chen, Haoran Xie, Di Zou, Gwo-Jen Hwang (2020) Application and theory gaps during the rise of Artificial Intelligence in Education. *Computers and Education: Artificial Intelligence*, 1.
10. Yağcı M. (2022) Educational data mining: prediction of students' academic performance using machine learning algorithms. *Smart Learn. Environ*, 9 (11).