DOI 10.31891/2307-5732-2023-323-4-257-261
УДК 004.8

ПИЛИПЕНКО ВЛАДИСЛАВ
Київський національний університет технологій та дизайну
ORCID ID: 0000-0002-2761-4817
e-mail: pylypenko.vi@knutd.edu.ua
СТАЦЕНКО ВОЛОДИМИР
Київський національний університет технологій та дизайну
ORCID ID: 0000-0002-3932-792X
e-mail: statsenko.v@knutd.edu.ua

ПРОГНОЗУВАННЯ АКТИВНОСТІ КОРИСТУВАЧІВ ПЛАТФОРМИ MOODLE НА БАЗІ МЕТОДІВ МАШИННОГО НАВЧАННЯ

В роботі створено модель машинного навчання на базі бібліотеки Scikit-learn. На основі даних про дії користувачів платформи Moodle, модель дозволяє виконати прогнозування їх активності. Перевірено, що використаний метод випадкового лісу має відносно високу точність та низьку тривалість процесу навчання. Розраховано загальну точність розробленої моделі, яка становить 83%. Перевірено, що використання методу машинного навчання "Random Forest" для задач класифікації добре підходить для прогнозування категорії або класу нового зразка активності користувача на основі його характеристик.

Розрахунки точності моделі показують, що вибір бібліотеки Scikit-Learn дозволить створити ефективну модель обробки даних і прогнозування результатів. Використання створеної моделі для прогнозування дозволить оперативно аналізувати активність користувачів і формувати, при необхідності, відповідні рейтинги. Прогноз, отриманий за допомогою моделі, буде корисний як для викладачів, так і навчальних закладів, оскільки дозволить планувати зміни в навчальних програмах та матеріалах, а також освітньому процесі в цілому.

Ключові слова: платформа управління навчанням, Machine Learning, Python, Scikit-learn, Moodle.

PYLYPENKO VLADYSLAV
Kyiv National University of Technologies and Design
STATSENKO VOLODYMYR
Kyiv National University of Technologies and Design

PREDICTION OF USERS ACTIVITY IN THE MOODLE PLATFORM BASED ON MACHINE LEARNING METHODS

The article presents the creation of a machine learning model for activity prediction based on the Scikit-learn library. The model allows to predict activity based on data about the actions of users of the Moodle platform. The program was developed in the Python language in the PyCharm software development environment. The amount of data taken for processing was 1000 samples of users from the Moodle database. Classification was used as the machine learning task, and the random forest method was used as the method. Random forest copes well with overfitting problems and scales well for large data sets. It is also an ensemble method that combines several decision trees to achieve better accuracy and stability compared to single decision trees. It has been verified that the random forest method has relatively high accuracy and low duration of the learning process. The overall accuracy of the developed model was calculated, which is 83%. Increasing the accuracy of the obtained model is possible due to the expansion of the source data, which requires the creation of appropriate applications (plugins) for the Moodle platform. It has been verified that the use of the Random Forest machine learning method for classification tasks is well suited for predicting the category or class of a new sample of user activity based on its characteristics.

The presented information shows that choosing the Scikit-Learn library will allow to create an effective model of data processing and prediction of results. The statement about the feasibility of choosing the Scikit-Learn library also coincides with the result of the analysis of modern libraries used for the development and training of machine learning models. The use of the created model for forecasting will allow to quickly analyze the activity of users and form, if necessary, appropriate ratings.

Keywords: learning management platform; Machine Learning; Python; Scikit-learn; Moodle.

Постановка проблеми

Moodle є однією з найпопулярніших у світі платформ для навчання та викладання, яка використовується в більш ніж 200 країнах [1]. Вона є основою систем управління навчанням багатьох університетів та шкіл, що використовують її для проведення навчання, онлайн-курсів, тестувань, взаємодії між студентами та викладачами. За даними Capterra [2], станом на 2023 рік Moodle входить у топ-20 платформ управління навчанням для академічних та освітніх цілей і випереджає конкурентів: LAMS, Sakai та ATutor [3]. Однак, зі зростанням кількості користувачів на платформі, що використовується освітнім закладом, часто з'являється потреба в аналізі даних користувачів для оптимізації різних освітніх процесів. Зокрема важливим фактором є визначення активності студентів у освітньому процесі.

Завдяки оцінці активності можна зрозуміти, наскільки ефективно студенти працюють над виконанням навчальних завдань, побачити загальний ступінь залучення до освітнього процесу в цілому, прогнозувати успішність. А в подальшому розвиток даних досліджень допоможе визначати, які саме навчальні матеріали користуються найбільшою популярністю у студентів. Для вирішення поставленої задачі доцільно застосувати методи машинного навчання, які дозволяють побудувати моделі на основі інформації про дії користувачів в системі Moodle. Першим етапом є збір та підготовка даних для тренування моделі. Наступні етапи передбачають визначення типу моделі та розрахунок її параметрів (тренування). Після тренування модель може використовуватись для прогнозування результатів на нових даних. В даному дослідженні джерелом даних є база даних Moodle. Прогноз, що отриманий за допомогою моделі буде

корисний як для викладачів так і навчальних закладів. Оскільки дозволить планувати зміни в навчальних програмах та матеріалах, а також освітньому процесі в цілому.

Аналіз досліджень і публікацій

Авторами розглянуто основні методи та задачі машинного навчання, які можна використати для вирішення даної задачі, зокрема: класифікація, кластеризація [4]. Прогнозування у машинному навчанні – це процес створення моделей, які можуть прогнозувати нові дані на основі зразків із навчальних даних. Воно має великий потенціал у багатьох областях, таких як освіта, наука, бізнес, медицина та інші. Зокрема, в освіті та науці це дає можливість для прогнозування успішності студентів, виявлення популярності різних предметів та прогнозування ефективності методів навчання. Також авторами було розглянуто основні бібліотеки для створення та навчання моделей на базі машинного навчання, та обрано бібліотеку Scikit-learn [5].

Формулювання цілей статті

Метою роботи є створення моделі для прогнозування активності користувачів навчальної платформи Moodle на базі методів машинного навчання, та визначення її загальної точності.

Виклад основного матеріалу

На сьогодні існує ряд бібліотек для Python, які широко застосовуються для вирішення подібних задач, до найбільш популярних відносяться: Scikit-learn [5], PyTorch [6] та TensorFlow [7]. Для створення моделі обрано Scikit-learn, оскільки вона містить багато готових алгоритмів для роботи з моделями, якісно оформлену і описану документацію та зручний інтерфейс.

Для побудови моделі прогнозування активності користувачів платформи Moodle у роботі обрано метод випадкового лісу (Random Forest), який широко застосовується у задачах регресії, класифікації та кластеризації. Ідея алгоритму полягає у використанні ансамблю дерев прийняття рішень, для отримання більш точного та стійкого результату. Класифікація здійснюється в два етапи, на першому визначається клас об'єкта по кожному з дерев, що входять до ансамблю. На другому визначається за який клас проголосувала найбільша кількість дерев і цей клас обирається як остаточний. Це дозволяє підвищити точність моделі. Також перевагою методу є відносно висока швидкість тренування, зокрема за рахунок можливості паралелізації обчислень та ефективного використання сучасних багатоядерних процесорів. Таким чином, на основі методу класифікації випадкового лісу (random forest) дозволяє побудувати модель, яка може класифікувати користувачів в залежності від їхньої активності на платформі [8].

Вихідними даними для моделі є записи з бази даних Moodle (експортовані в csv формат), а на виході модель розраховує прогноз активності користувача платформи. Використання класифікації, в даному випадку, зручне для передбачення категорії або класу нового зразка на основі його характеристик [9]. Оскільки класифікаційні моделі побудовані на основі навчання зразків з відомими категоріями. Ці моделі можуть виділяти ті характеристики, які найбільше сприяють віднесенню зразка до певного класу, і використовувати їх для прийняття рішення про класифікацію нового зразка. Що в свою чергу дає можливість визначати, чи є користувачі активними або неактивними на основі їх історії взаємодії з платформою.

Для створення моделі прогнозування активності користувачів з бази даних Moodle були обрані параметри, що тим чи іншим чином пов'язані з діями, які виконують студенти під час роботи з платформою. Перелік цих параметрів наведено у табл. 1. Їх значення були експортовані з бази даних та збережені у csv форматі (файл moodle_data.csv). Всього експортовано записи 1000 студентів. Після цього на основі результатів оцінювання з відповідних дисциплін було проведено оцінювання активності кожного студента. При цьому оцінка «відмінно» відповідала значенню HIGH для поля «Activity», оцінка «добре» – значенню MEDIUM, оцінка «задовільно» – значенню LOW. Студенти, що не отримали позитивну оцінку до вибірки не включались.

Таблиця 1

Вихідні параметри для створення моделі прогнозування активності користувачів платформи Moodle

Назва	UserID	FirstName	LastName	VisitDate	VisitCount	SpendTime	Activity
Тип	Integer	String	String	String	Integer	Integer	String
Значення	3580611	#####	#####	30/03/2023	7	358	HIGH

де UserID – унікальний ідентифікатор користувача на платформі;
FirstName, LastName – ім'я та прізвище користувача;
VisitDate – дата останнього візиту на платформу;
VisitCount – кількість відвідувань;
SpendTime – час проведений на платформі (у хвилинах);
Activity – експертна оцінка активності студента (можливі значення LOW, MEDIUM, HIGH).

Перед виконанням навчання моделі вихідні дані були розділені на тренувальну та тестову вибірки для того, щоб перевірити, наскільки добре модель, навчена на тренувальній вибірці, може передбачати класи нових даних. Обсяг даних взятих для обробки складав 1000 вибірок користувачів із бази даних, з яких тренувальна вибірка містила 800, а тестова вибірка – 200. Ділення даних на тренувальну та тестову вибірки допомагає уникнути перенавчання (overfitting) моделі [10]. Програмний код завантаження і розділення

даних на тренувальну та тестову вибірки представлено у лістингу 1:

Лістинг 1. Програма формування груп даних для тренувальної та тестової вибірок.

```
# Підключення бібліотек
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
# Завантаження даних
data = pd.read_csv('moodle_data.csv')
# Визначення змінної залежної та незалежних змінних
X = data[['VisitCount', 'SpendTime']]
y = data['Activity']
# Розділення даних на тренувальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Операція розділення даних здійснюється за допомогою функції train_test_split [11], що входить до бібліотеки Scikit-learn. Аргумент test_size=0.2 означає, що тестовий набір даних становитиме 20% від загальної кількості даних, а навчальний набір буде складатися з 80%. Це забезпечує баланс між обсягом навчальних та тестових даних для ефективної оцінки ефективності моделі. Таке співвідношення у розподілі даних обрано відповідно до рекомендацій [12]. Аргумент random_state=42 встановлює початкове значення для генератора випадкових чисел, що забезпечує відтворюваність результатів. Це дозволяє забезпечити стабільні результати під час відлагодження моделі та її налаштування. Кожне дерево у випадковому лісі будується на підмножині навчальних даних, що обираються випадковим чином з повного набору даних, та має свої параметри, що також вибираються випадковим чином. Результатом роботи моделі є класифікація користувачів на три класи: "низька активність" (LOW), "середня активність" (MEDIUM) та "висока активність" (HIGH). Даний метод групує користувачів за спільними ознаками. Створення об'єкту класифікатора випадкового лісу за допомогою конструктора класу RandomForestClassifier з бібліотеки scikit-learn, виглядає наступним чином:

```
rfc = RandomForestClassifier(n_estimators=5, random_state=42)
rfc.fit(X_train, y_train)
```

Цей об'єкт використовувався для тренування моделі на вхідних даних та подальшої класифікації. Параметр n_estimators=5 вказує на кількість дерев, що використовувались у моделі, а параметр random_state=42 встановлює початкове значення для генератора випадкових чисел. При цьому збільшення значення параметру n_estimators може допомогти покращити точність моделі, але збільшує час тренування та складність моделі. У даному випадку обране значення n_estimators=5 обумовлене обчислювальною потужністю обладнання та часом тренування моделі. За допомогою методу fit() модель випадкового лісу навчається на тренувальних даних, переданих у масивах X_train та y_train. Результат тренування моделі показано на рис. 1. Порівняльні параметри VisitCount та SpendTime відповідають кількості відвідувань та часу проведеному на платформі відповідно.

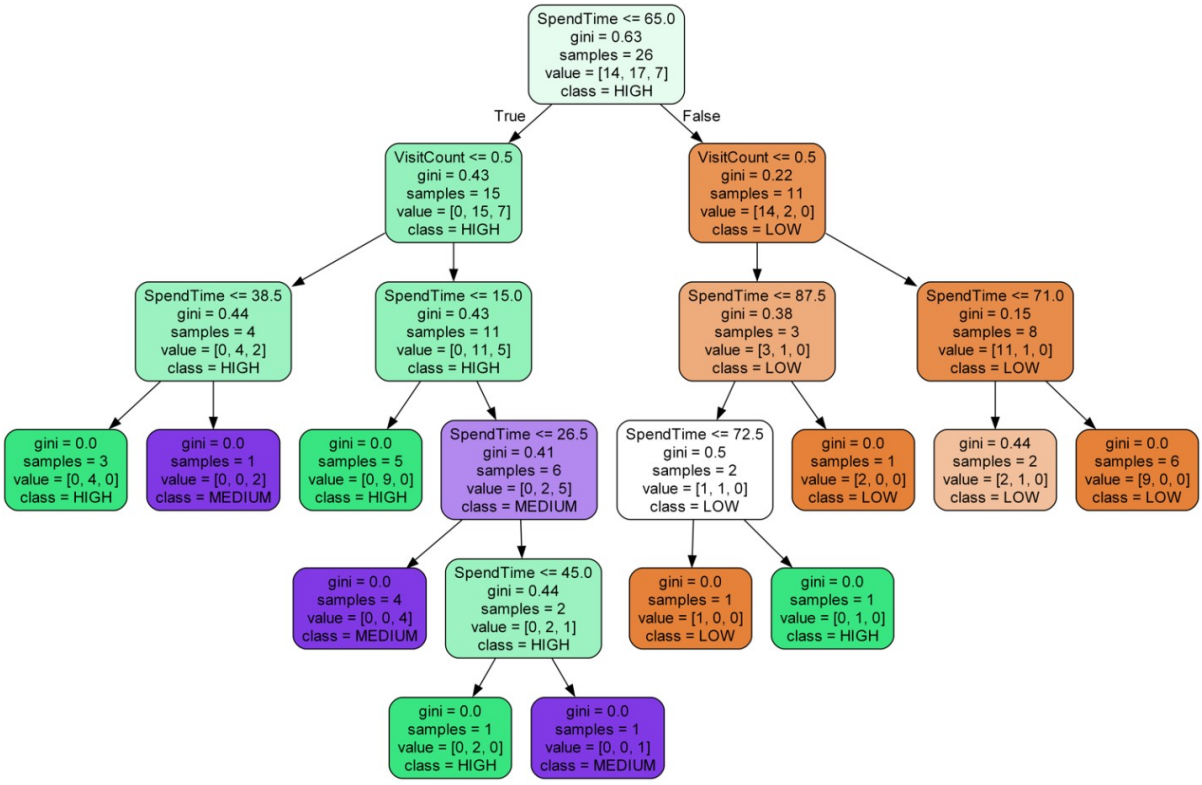

Рис. 1. Модель прогнозування активності користувачів платформи Moodle

gini (індекс) – один із критеріїв оцінки якості поділу вузла;

samples – кількість прикладів даних, які потрапляють до вузла;

value – кількість прикладів в кожному класі, які потрапляють у вузол;

class – клас, якому відповідає вузол дерева, в даному випадку це значення поля Activity, яке прогнозує модель.

Для визначення точності отриманої моделі, проведено розрахунок її ефективності [13]. Точність класифікації визначалась як співвідношення між вірними відповідями та загальною кількістю відповідей. При цьому використовувалась система True positive, True negative, False positive, False negative, для якої вираз для визначення точності можна записати у вигляді наступної формули:

$$Accuracy = (TP+TN)/(TP+TN+FP+FN) \quad (1)$$

де TP (true positives) – кількість правильно передбачених позитивних класів;

TN (true negatives) – кількість правильно передбачених негативних класів;

FP (false positives) – кількість неправильно передбачених позитивних класів;

FN (false negatives) – кількість неправильно передбачених негативних класів.

Під чутливістю бінарної моделі розуміється частка істинно-позитивних класифікацій в загальній кількості позитивних спостережень (TPR – true-positive rate), що є часткою правильно класифікованих позитивних спостережень [14]. Тому, чим вище чутливість, тим надійніше класифікатор розпізнає позитивні приклади. Під специфічністю моделі розуміється частка істинно-негативних класифікацій в загальній кількості негативних спостережень (TNR – true-negative rate). Таким чином, чим вище специфічність, тим надійніше класифікатор розпізнає негативні спостереження.

Перевірка точності моделі, за допомогою класифікатора випадкового лісу, виконана на тестовому наборі даних X_test. Результатом роботи методу predict() є передбачені значення класів, які модель призначила кожному елементу вхідного тестового набору даних. Ці передбачені значення зберігаються в змінній y_pred. Оскільки є набір вхідних даних, для якого ми знаємо правильні класифікації, то відповідно можна порівняти передбачені класифікації з правильними, щоб оцінити точність моделі. У роботі порівнювались передбачені класи y_pred з правильними класами, які зберігаються у змінній y_test. Далі здійснювалось порівняння передбачених значень y_pred з дійсними значеннями y_test і в результаті розраховувалась доля правильних передбачень, результат accuracy. Значення отриманої точності моделі показано на рис. 2. Програмний код для визначення точності моделі:

```
y_pred = rfc.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

Рис. 2. Виведення значення точності моделі в консолі

Отриманий результат складає 0.83, а це означає, що модель розроблена за допомогою Scikit-learn, правильно класифікує 83% тестових даних. Результат може відрізнятися в залежності від даних, що використовуються для навчання та тестування моделі, а також від параметрів, встановлених у моделі. Використання в подальшому даного рішення буде досить корисним у додатках для освітніх та навчальних цілей.

Висновки

1) У роботі створено модель, що дозволяє на основі даних інформації про дії користувачів платформи Moodle виконати прогнозування їх активності.

2) Для побудови моделі використано метод випадкового лісу, який має відносно високу точність та низьку тривалість процесу навчання. Із використанням даного методу отримано структуру дерева прийняття рішень та значення відповідних коефіцієнтів.

3) Розраховано точність розробленої моделі, яка становить 83%, що відповідає поставленим задачам дослідження.

4) Підвищення точності моделі можливе за рахунок розширення вихідних даних, що потребує створення відповідних додатків (плагінів) для платформи Moodle, та є перспективним напрямом розвитку таких систем.

References

1. About Moodle. 2022. https://docs.moodle.org/401/en/About_Moodle

2. Learning Management System Software. 2023. https://www.capterra.com/learning-management-system-software/

3. Statsenko V. V., Pavlenko V. M., Pylypenko V. I. CHOISE PROBLEM IN LEARNING MANAGEMENT SYSTEMS. Digital transformation and technologies for the sustainable development all branches of modern education, science and practice. Łomża: MANS w Łomży, 2023. 125–129.

4. Support Vector Machine – Introduction to Machine Learning Algorithms. 2018. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
5. Scikit-learn. 2023. https://en.wikipedia.org/wiki/Scikit-learn
6. PyTorch documentation. 2023. https://pytorch.org/docs/stable/index.html
7. TensorFlow. 2023. https://www.tensorflow.org/api_docs
8. Random Forests in Scikit-learn. 2023. https://scikit-learn.org/stable/modules/ensemble.html#random-forests
9. Haoyuan T. Machine Learning Algorithm for Classification. 2021. http://doi.org/10.1088/1742-6596/1994/1/012016
10. Model underfitting vs. overfitting. 2021. https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html
11. Train test split in Scikit-learn. 2022. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
12. Train/Test Split and Cross Validation in Python. 2017. https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6
13. Julian L. Analysis of precision and accuracy in a simple model of machine learning. 2017. https://doi.org/10.3938/jkps.71.866
14. Koray K. Values and inductive risk in machine learning modelling: the case of binary classification models. 2021. https://doi.org/10.1007/s13194-021-00405-1