МЕТОД ПЕРЕХРЕСНОЇ ПЕРЕВІРКИ У МАШИННОМУ НАВЧАННІ

Юрченко Ігор Валерійович
кандидат фіз.-мат. наук, доцент
доцент кафедри математичного моделювання
Чернівецький національний університет імені Юрія Федьковича, Україна

Гуцуляк Іван Васильович
студент 6-го курсу факультету математики та інформатики
Чернівецький національний університет імені Юрія Федьковича, Україна

Машинне навчання займається побудовою математичних моделей для дослідження даних. Задачі "навчання" починаються з появою у цих моделей параметрів, які можна налаштовувати та пристосовувати для відображення спостережуваних даних. Цей процес можна уявити як навчання програми на наявних даних. Після навчання моделі на наявних спостереженнях, її можна буде використовувати для передбачення та розуміння різноманітних даних нових спостережень. Математичне, засноване на моделях, навчання можна деякою мірою порівняти із навчанням людського мозку.

На базовому рівні машинне навчання можна поділити на два типи [1]:

1) машинне навчання з учителем (supervised learning) – містить моделювання ознак даних та відповідних до даних міток; після вибору моделі її можна використовувати для присвоєння міток новим, невідомим раніше даним; воно поділяється далі на задачу класифікації та задачу регресії. При класифікації мітки оперують дискретними категоріями, а при регресії використовують неперервні величини.
2) машинне навчання без учителя (unsupervised learning) – моделювання ознак набору даних без будь-яких міток. Ці моделі включають такі задачі, як кластеризація (clustering) та зниження розмірності (dimensionality reduction). Алгоритми кластеризації використовують для виділення окремих груп даних, тоді як алгоритми зниження розмірності призначені для пошуку найстисліших представлень даних.

Крім того, існують так звані методи часткового навчання (semi-supervised learning), що розташовані приблизно посередині між машинним навчанням з учителем та машинним навчанням без учителя. Методи часткового навчання бувають корисні у випадку неповних міток.

Методи машинного навчання зараз широко застосовуються у задачах розпізнавання образів, класифікації, кластеризації, функціонуванні нейромереж, систем штучного інтелекту та прогнозуванні [1, 2].

У машинному навчанні часто доводиться вибирати з різних моделей. Кожна модель має різні експлуатаційні характеристики. Використовуючи метод перехресної перевірки, можна отримати оцінку того, наскільки точною може бути кожна модель для нових невідомих даних.

Коли досліджується новий набір даних, рекомендується візуалізувати дані, використовуючи різні методи, щоб дивитися на дані з різних точок зору. Така сама ідея відноситься і до вибору моделі. Необхідно використовувати різні способи оцінки передбачуваної точності використовуваних алгоритмів машинного навчання, щоб вибрати один або два найкращих.

Один зі способів зробити це – використовувати різні методи візуалізації, щоб показати середню точність, дисперсію та інші властивості розподілу точності моделей.

Вивчення параметрів функції прогнозування і тестування на одних і тих самих даних є методологічною помилкою: модель, яка просто повторить мітки зразків, які вона щойно “бачила”, матиме ідеальну оцінку, але не зможе передбачити нічого корисного на ще “небачених” даних. Така ситуація називається переналаштуванням. Щоб уникнути цього, звичайною практикою при виконанні експерименту машинного навчання “з учителем” є використання частини доступних даних як тестовий набір X_test, Y_test.

У бібліотеці scikit-learn [2, 3] випадковий розподіл на навчальні та тестові набори можна швидко провести за допомогою функції train_test_split. Найкращі параметри можна визначити за методом grid_search.

При оцінці різних налаштувань ("гіперпараметрів") для оцінювачів все ще існує ризик переналаштування на тестовому наборі, оскільки параметри можуть бути змінені, поки оцінювач не стане оптимальним. Таким чином, знання про тестовий набір можуть «просочуватися» у модель, а показники оцінки більше не свідчать про ефективність узагальнення. Щоб вирішити цю проблему, ще одна частина набору даних може бути проведена як т.зв. “набір валідації”: навчання триває на навчальному наборі, після чого оцінка проводиться на наборі валідації, а коли експеримент здається успішним, остаточна оцінка може бути зроблена на тестовому наборі.

Однак, розділивши наявні дані на три набори, ми різко зменшуємо кількість зразків, які можуть бути використані для вивчення моделі, і результати можуть залежати від конкретного випадкового вибору для пари (навчання, валідація) наборів.

Розв’язанням цієї проблеми є процедура, яка називається перехресною валідацією (cross validation, CV) [4]. Тестовий набір все ще має бути використаний для остаточної оцінки, але набір перевірки більше не потрібен при виконанні CV. У базовому підході, так званому *k*-складковому CV, навчальний набір ділиться на *k* менших наборів. Для кожної з *k* “складок” дотримується наступна процедура:

- модель тренується з використанням *(k−1)* “складок” в якості тренувальних даних;
- отримана модель перевіряється на іншій частині даних (тобто вона використовується як тестовий набір для обчислення міри продуктивності, такої як точність).

Показник продуктивності, про який повідомляє *k*-складкова перехресна перевірка, є середнім значенням, обчисленим у циклі. Цей підхід може бути обчислювально витратним, але він не витрачає занадто багато даних (як у

випадку з виправленням довільного набору валідації), що є основною перевагою в таких задачах, як “обернений вплив”, де кількість зразків дуже мала.

Задача полягає у дослідженні стандартного набору даних бінарної (0 або 1) класифікації з репозиторію машинного навчання (Pima Indians Diabetes Dataset), який ілюструє початок захворювання діабетом у індіанців південноамериканського племені Піма [5]. Дані мають вісім вхідних показників: Pregnancies (перенесена вагітність), Glucose (рівень глюкози), BloodPressure (кров’яний тиск), SkinThickness (товщина шкіри), Insulin (рівень інсуліну), Body Mass Index (індекс маси тіла), Diabetes Pedigree (генеалогія діабету у родичів), Age (вік). Результуючий прогнозний показник – Outcome (0 або 1 – відсутність або наявність діабету).

Процедура 10-кратної перехресної перевірки використовується для оцінки кожного з оцінюваних алгоритмів (*суміш гауссових розподілів*, *логістична регресія, гауссовий наївний байєс, бернуллів наївний байєс, мультиноміальний наївний байєс*), сконфігурованого з одним і тим самим випадковим початковим числом, щоб гарантувати, що виконуються однакові розбиття для навчальних даних і що кожний алгоритм оцінюється точно так само [2,4].

Для візуалізації результатів краще використовувати засіб побудови графіків з бібліотеки matplotlib.pyplot, який називається boxplot. Він створений для відображення результуючого набору значень для даних, що мають такі властивості, як мінімальне значення, перший квартиль, медіана, третій квартиль і максимальне значення. На Рис. 1 створюється прямокутник (box) від першого квартиля до третього квартиля, є “вуса”, які показують мінімальне та максимальне значення, також є лінія, яка проходить через прямокутник в медіані. Вісь *x* позначає дані, які потрібно побудувати, вісь *y* показує розподіл частоти (у даному випадку точності – accuracy).

Обчислення проводилися з використанням бібліотек Pandas, Scikit-learn, Matplotlib мови Python [2–4]. Результати наведено на Рис. 1.

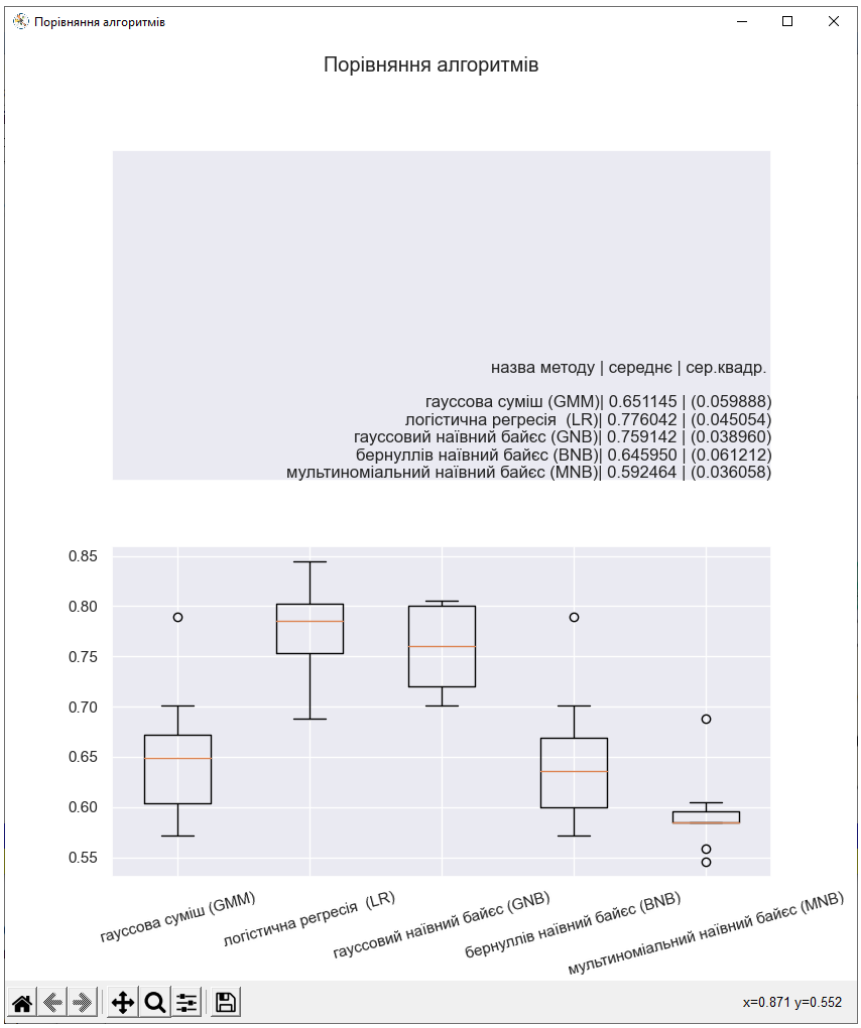

Рис.1. Візуалізація результатів процедури перехресної перевірки

Аналіз графіку boxplot показує, що алгоритми *логістична регресія* та *гауссовий наївний байєс* заслуговують на вибір серед низки інших запропонованих методів.

Список літератури

1. Jake VanderPlas. Python Data Science Handbook. Essential Tools for Working with Data.– Beijing, Boston, Farnham, Tokyo: O'Reilly Media, Inc, 2016.– 576 p.– ISBN: 9-781-491-912-058.
2. Бібліотека Scikit-learn мови Python [Електронний ресурс].– Джерело доступу: https://scikit-learn.org/stable/
3. Scikit-learn–Вікі [Електронний ресурс].– Джерело доступу: https://uk.upwiki.one/wiki/Scikit-learn
4. Scikit-learn – Cross-validation [Електронний ресурс]. Джерело доступу: https://scikit-learn.org/stable/modules/cross_validation.html
5. Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus // *Proceedings of the Symposium on Computer Applications and Medical Care*.– IEEE Computer Society Press, 1988.– 261–265 pp.