УДК 004.85
DOI https://doi.org/10.32838/TNU-2663-5941/2020.3-1/19

Кравченко С.М.
Державний університет «Житомирська політехніка»

Гришкун Є.О.
Державний університет «Житомирська політехніка»

Власенко О.В.
Державний університет «Житомирська політехніка»

МЕТОДИ КЛАСИФІКАЦІЇ МАШИННОГО НАВЧАННЯ З ВИКОРИСТАННЯМ БІБЛІОТЕКИ SCIKIT-LEARN

Об'єктом дослідження є використання різних алгоритмів класифікації під час групування результатів для моделей машинного навчання в галузі бінарної й багатокласової класифікації. У статті розглянуто вибір такого алгоритму машинного навчання, який залежить від декількох факторів, включаючи розмір даних, їх якість і різноманітність, а також розуміння, які відповіді на основі цих даних потрібні бізнесу. Через це доводиться пробувати багато різних алгоритмів, перевіряючи ефективність кожного на тестовому наборі даних, і потім вибирати кращий варіант. Зважаючи на це, потрібно вибирати серед наявних алгоритмів ті, які відповідають цьому завданню. Автори статті приділили увагу точності, часу на навчання, параметрам, даним. Тому вибір правильного алгоритму – це поєднання бізнес-потреб, специфікацій, експериментальної роботи й обліку доступного часу. Досліджено впровадження методів машинного навчання в різні сфери. Представлено процес машинного навчання, який містить такі етапи: підготовка даних, створення навчальних наборів, створення класифікатора, навчання класифікатора, складання прогнозів, оцінювання продуктивності класифікатора й настройка параметрів. Проаналізовано використання різних алгоритмів класифікації за допомогою бібліотеки Python, Scikit-learn, здійснено аналіз використання методу підбору моделі, обчислення, форматування й підготовки даних, підібрано оптимальні вхідні значення й моделі. Приведено оцінювання декількох варіантів оцінювання класифікатора. Метою роботи є дослідження бібліотеки для ефективності її практичного застосування. Представлено методи класифікації в машинному навчанні за допомогою бібліотеки Scikit-Learn. Здійснюється порівняння різних методів класифікації за допомогою Scikit-learn для моделей машинного навчання.

Ключові слова: *машинне навчання, класифікація, оптимізація, аналіз даних, прогнозування.*

Постановка проблеми. Існує безліч методів класифікації, які використовують різний математичний апарат і різні підходи під час реалізації [1]. Однак ефективність цих методів залежить від конкретного завдання, яке буде вирішуватися. Незважаючи на те що останнє десятиліття комерційні компанії займаються проблемою підвищення якості машинного навчання, натепер не існує методів, які могли б однозначно ефективно вирішити завдання класифікації. Тому необхідно проаналізувати застосування різних алгоритмів класифікації за допомогою бібліотеки Scikit-Learn

Аналіз останніх досліджень і публікацій. Технологія машинного навчання на основі аналізу даних бере початок у 1950 році, коли почали розробляти перші програми для гри в шашки. За минулі десятиліття загальний принцип не змінився. Зате завдяки вибуховому зростанню обчислювальних потужностей комп'ютерів багаторазово ускладнилися закономірності й прогнози, які створюються ними, і розширилося коло проблем і завдань, що вирішуються з використанням машинного навчання.

Останніми роками проведено велику кількість досліджень, що демонструють упровадження методів машинного навчання в різні сфери [2]. Машинне навчання використовується здебільше для вирішення проблем, які є занадто складними та потребують адаптації. Тобто це такий клас завдань, який неможливо вирішити певним чітким алгоритмом, необхідно зважати на вже отримані результати. Аналіз літературних джерел про застосування цих методів обмежується невеликою кількістю інформації.

Постановка завдання. Мета статті – за допомогою бібліотеки Python, Scikit-learn проаналізувати

використання різних алгоритмів класифікації, використання методу підбору моделі, обчислення, форматування й підготовки даних; підібрати оптимальні вхідні значення й моделі; дослідження бібліотеки для ефективності її практичного застосування.

Виклад основного матеріалу дослідження. Завдяки машинному навчанню програміст не зобов'язаний писати інструкції, що враховують усі можливі проблеми й містять усі рішення. Замість цього, у комп'ютер (або окрему програму) закладають алгоритм самостійного знаходження рішень шляхом комплексного використання статистичних даних, із яких виводяться закономірності й на основі яких робляться прогнози.

Натепер машинне навчання дає змогу комп'ютерам навчатися розпізнавати на фотографіях і рисунках не тільки осіб, а і пейзажі, предмети, текст і цифри. Що стосується тексту, то й тут не обійтися без машинного навчання: функція перевірки граматики зараз наявна в будь-якому текстовому редакторі й навіть у телефонах. Причому враховується не тільки написання слів, а й контекст, відтінки сенсу й інші тонкі лінгвістичні аспекти. Більш того, уже існує програмне забезпечення, здатне без участі людини писати новинні статті (на тему економіки або, наприклад, спорту).

Типи завдань машинного навчання.

Усі завдання, які вирішуються за допомогою машинного навчання, належать до такої категорії (таблиця 1).

Приклад використання.

Завдання класифікації і регресії – це завдання навчання з учителем. Як приклад будемо представляти завдання кредитного скорингу: на основі накопичених кредитною організацією даних про своїх клієнтів можна прогнозувати неповернення кредиту. Тут для алгоритму досвід E – це наявна навчальна вибірка: набір об'єктів (людей), кожен із яких характеризується набором ознак (таких як вік, зарплата, тип кредиту, неповернення в минулому тощо), а також цільовою ознакою. Якщо цільова ознака – просто факт неповернення кредиту (1 або 0, тобто банк знає про своїх клієнтів, хто повернув кредит, а хто – ні), то це завдання (бінарної) класифікації. Якщо відомо, на скільки часу клієнт затягнув з поверненням кредиту, хочеться те саме прогнозувати для нових клієнтів, то це буде завданням регресії [2].

Класифікатор (classifier) – це система, яка вводить (як правило) вектор дискретних і/або неперервних функцій і виводить одне дискретне значення класу.

Наприклад, фільтр спаму класифікує повідомлення електронної пошти на «спам» або «не спам», і його вхідними даними може бути вектором булевих значень $x = (x_1, ..., x_j, ..., x_d)$, де $x_j = 1$, якщо j-е слово в словнику з'являється в електронному листі, а $x_j = 0$ в іншому випадку. Учень (learner) уводить навчальний набір (training set) прикладів (x_i, y_i), де $x_i = (x_{i,1}, ..., x_{i,d})$ – це спостережуваний ввід, y_i – відповідний вихід і виводить класифікатор. Тест учня полягає в тому, чи дає цей класифікатор правильний вихід y_t для майбутніх прикладів x_t (наприклад, чи фільтр спаму правильно класифікує раніше невидимі електронні листи як спам чи не спам).

Машинне навчання проводиться за використанням різних алгоритмів, проте для всіх алгоритмів найважливішими є 3 компоненти:

– Представлення. Класифікатор повинен бути представлений за допомогою формальної мови, яку комп'ютер може обробляти. І, навпаки, вибір представлення для учня рівносильний вибору набору класифікаторів, яких він може навчитися. Цей набір називається гіпотезою простору (hypothesis space) учня. Якщо класифікатор не знаходиться в гіпотезі простору, то він не можу бути вивчений.

– Оцінювання. Функція оцінювання (також звана цільова функція (objective function)) або функція оцінювання (scoring function)) потрібна для виділення хороших класифікаторів від поганих. Функція оцінювання, яка використовується всередині алгоритму, може відрізнятися від зовнішньої, яку ми хочемо оптимізувати для класифікатора, для простоти оптимізації.

– Оптимізація. Нарешті, нам потрібен метод пошуку серед класифікаторів того, який буде класифікувати найбільш швидко й правильно. Вибір методу оптимізації є ключовим елементом ефективності учня, а також допомагає визначити вибраний класифікатор, якщо функція оцінки має більше ніж один оптимум. Для нових учнів найкраще почати використовувати загальноприйняті оптимізатори, які пізніше замінюються спеціально розробленими. [4]

Класифікація є дуже великою частиною галузі, зокрема статистика й машинне навчання. Як правило, може бути розбита на 2 частини:

1. Бінарна класифікація – групування результату в одну з двох груп.

2. Багатокласова класифікація – групування результату в одну з декількох (більше двох) груп.

Методи класифікації в машинному навчанні за допомогою бібліотеки Scikit-Learn.

Для машинного навчання на Python написано дуже багато бібліотек. Розглянемо одну з найпопулярніших – Scikit-Learn.

Що таке Scikit-Learn?

Scikit-Learn – це Python-бібліотека, уперше розроблена David Cournapeau в 2007 році. У цій бібліотеці знаходиться велика кількість алгоритмів для задач, пов'язаних із класифікацією й машинним навчанням загалом.

Scikit-Learn базується на бібліотеці SciPy, яку потрібно встановити перед початком роботи. Scikit-Learn спрощує процес створення класифікатора й допомагає більш чітко виділити концепції машинного навчання, реалізуючи їх за допомогою зрозумілої, добре документованої й надійної бібліотеки. Вона містить низку методів, що охоплюють усе, що може знадобитися під час аналітики даних: алгоритми класифікації та регресії, кластеризації, валідації й вибір моделей. Також її можна застосовувати для зменшення розмірності даних і виділення ознак (рис. 2).

У системах машинного навчання існують входи й виходи. Те, що подається на вхід, прийнято називати ознаками. Коли ознаки подаються на входи системи машинного навчання, ця система намагається знайти збіг, помітити закономірність

Таблиця 1

Категорії завдань

Завдання	Призначення
Регресії	Прогнозування на основі вибірки об'єктів з різними ознаками. На виході має вийти дійсне число (2, 35, 76,454 тощо). Наприклад, ціна квартири, вартість цінного паперу після півроку, очікуваний дохід магазину на наступний місяць, якість вина при сліпому тестуванні.
Класифікації	Отримання категоріальної відповіді на основі набору ознак. Має кінцеве кількість відповідей (як правило, у форматі «так» або «ні»): чи є на фотографії кіт, чи є зображення людським обличчям, хворий пацієнт раком. Застосовується в маркетингу під час оцінювання кредитоспроможності позичальників, визначення лояльності клієнтів, розпізнавання образів, медичної діагностики та в багатьох інших сферах.
Кластеризації	Розподіл даних на групи: поділ усіх клієнтів мобільного оператора за рівнем платоспроможності, зарахування космічних об'єктів до тієї чи іншої категорії (планета, зірка, чорна діра тощо).
Зменшення розмірності	Зведення великої кількості ознак до меншого (зазвичай 2–3) для зручності їх подальшої візуалізації (наприклад, стиснення даних).
Виявлення аномалій	Виявлення аномалій від стандартних випадків. На практиці таким завданням є, наприклад, виявлення шахрайських дій з банківськими картами.

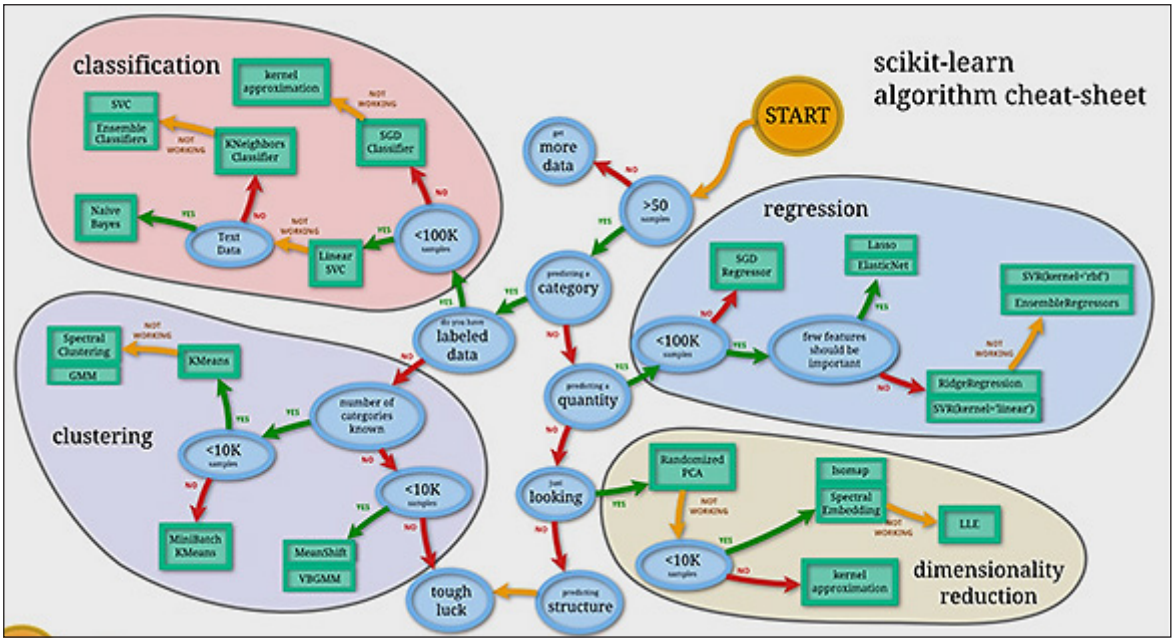

Рис. 1. Scikit-learn. Алгоритми

між ознаками. На виході генерується результат цієї роботи [3].

Цей результат прийнято називати міткою, оскільки у виходів є деяка помітка, яка видається системою, тобто прогнозування того, у яку категорію попадає вихід після класифікації.

Scikit-Learn надає доступ до різних алгоритмів класифікації. Основні з них:

– метод k-найближчих сусідів (K-Nearest Neighbors);

– метод опорних векторів (Support Vector Machines);

– класифікатор дерева рішень (Decision Tree Classifier) / случайный лес (Random Forests);

– наївний байєсівський метод (Naive Bayes);

– лінійний дискримінантний аналіз (Linear Discriminant Analysis);

– логічна регресія (Logistic Regression);

Приклади завдань класифікації

Завдання класифікації – ця будь-яке завдання, де потрібно визначити тип об'єкта з двох і більше наявних класів. Такі завдання можуть бути різними: визначення, кішка на зображенні або собака, або визначення якості вина на основі його кислотності й умісту алкоголю.

Залежно від завдання класифікації можна використовувати різні типи класифікаторів. Наприклад, якщо класифікація містить якусь бінарну логіку, то до неї найкраще підійде логістична регресія.

Процес машинного навчання

Процес містить у собі такі етапи: підготовка даних, створення навчальних наборів, створення класифікатора, навчання класифікатора, складання прогнозів, оцінювання продуктивності класифікатора й настройка параметрів.

По-перше, потрібно підготувати набір даних для класифікатора: перетворити дані в коректну для класифікації форму й обробити будь-які аномалії в цих даних. Відсутність значень у даних або будь-які інші відхилення – усі їх потрібно обробити, інакше вони можуть негативно впливати на продуктивність класифікатора. Цей етап називається попередньою обробкою даних (Data preprocessing).

Наступним кроком буде поділ даних на навчальні й тестові набори. Для цього в Scikit-Learn існує відмінна функція traintestsplit.

Як уже сказано вище, класифікатор повинен бути створений і навчений на тренувальному наборі даних. Після цих кроків модель уже може робити прогнози. Порівнюючи свідчення класифікатора з фактично відомими даними, можна робити висновок про точність класифікатора.

Найімовірніше, потрібно буде «коригувати» параметри класифікатора, поки не буде досягнуто бажаної точності, так як малоймовірно, що класифікатор буде відповідати всім вимогам із першого ж запуску [4].

Оцінювання класифікатора: існує декілька варіантів оцінювання:

– Точність класифікації.

Точність класифікації вимірювати найпростіше, і тому цей параметр найчастіше використовується. Значення точності – це число правильних прогнозів, поділене на число всіх прогнозів або, простіше кажучи, ставлення правильних прогнозів до всіх.

Хоча цей показник і може швидко дати явне уявлення про продуктивність класифікатора, його краще використовувати, коли кожен клас має хоча б приблизно однакову кількість прикладів. Так як таке буде траплятися рідко, рекомендується використовувати інші показники класифікації.

– Логарифмічні втрати.

Значення логарифмічних втрат (логлос) показує, наскільки класифікатор «упевнений» у своєму прогнозі. Логлос повертає ймовірність належності об'єкта до того чи іншого класу, підсумовуючи їх, щоб дати загальне уявлення про «впевненість» класифікатора.

```
In [21]: from sklearn.ensemble import RandomForestClassifier

In [22]: x = known_values[['free', 'super', 'source']]
         y = known_values['phone_type']

In [23]: model = RandomForestClassifier(n_estimators = 100)
         model = model.fit(x, y)

In [25]: sample_user = [1, 6, 0]
         model.predict([sample_user])
```

Рис. 2. Проста класифікація з використанням моделі «випадковий ліс» у Scikit-learn

Цей показник лежить у проміжку від 0 до 1 – «зовсім не впевнений» і «повністю впевнений» відповідно. Логлос сильно падає, коли класифікатор сильно «впевнений» у неправильній відповіді.

– Площа ROC-кривою (AUC).

Такий показник використовується тільки при бінарній класифікації. Площа під ROC-кривої становить здатність класифікатора розрізняти підходящі й не відповідні якому-небудь класу об'єкти.

Значення 1.0: уся область, яка потрапляє під криву, являє собою ідеальний класифікатор. Отже, 0.5 означає, що точність класифікатора відповідає випадковості. Крива розраховується з урахуванням точності й специфічності моделі. Детальніше про розрахунки можна прочитати тут.

– Матриця неточностей.

Матриця неточностей (англ. Confusion Matrix) – це таблиця або діаграма, що показує точність прогнозування класифікатора щодо двох і більше класів. Прогнози класифікатора знаходяться на осі X, а результат (точність) – на осі Y.

У міру накопичення досвіду буде простіше вибирати відповідний тип класифікатора. Однак хорошою практикою є реалізація декількох відповідних класифікаторів і вибір найбільш оптимального й продуктивного.

Scikit-learn становить широкий спектр алгоритмів машинного навчання, як контрольованих, так і неконтрольованих, із використанням узгодженого, орієнтованого на завдання інтерфейсу, що дає змогу легко порівнювати методи для цієї заяви. Оскільки він спирається на наукову екосистему Python, його можна легко інтегрувати в додаток поза традиційним діапазоном статистичного аналізу даних. Важливо відзначити, що алгоритм, реалізований мовою високого рівня, може використовуватися як будівельні блоки для підходів, специфічних для використання, наприклад, у медичній візуалізації (Michel et al., 2011). Майбутня робота включає в себе онлайн навчання, масштабування до великих наборів даних.

Висновки. Машинне навчання в scikit-learn полягає в тому, щоб імпортувати правильні модулі й запустити метод підбору моделі. Складніше вичистити, відформатувати й підготувати дані, а також підібрати оптимальні вхідні значення та моделі. Тому перш ніж узятися за scikit-learn, потрібно, по-перше, відпрацювати навички роботи з Python і Pandas, щоб навчитися якісно готувати дані, а по-друге, освоїти теорію й математичну основу різних моделей прогнозування та класифікації, щоб розуміти, що відбувається з даними під час їх застосування.

Список літератури:

1. Rao C., Govindaraju V. Handbook of Statistics: Machine Learning: Theory and Applications, 2013. 552 с.
2. Введение в машинное обучение с помощью Python и Scikit-Learn. URL: https://habr.com/ru/company/mlclass/blog/247751/ (дата звернення: 20.05.2020).
3. Классификация в Python с Scikit-Learn и Pandas. URL: https://stackabuse.com/classification-in-python-with-scikit-learn-and-pandas/ (дата звернення: 17.05.2020).
4. Обзор методов классификации в машинном обучении с помощью Scikit-Learn. URL: https://tproger.ru/translations/scikit-learn-in-python/s://stackabuse.com/classification-in-python-with-scikit-learn-and-pandas/ (дата звернення: 10.05.2020).

Kravchenko S.N., Grishkun E.O., Vlasenko O.V. CLASSIFICATION METHODS FOR MACHINE LEARNING USING THE SCIKIT-LEARN LIBRARY

The object of the research is to use different classification algorithms when grouping results for machine learning models in the field of binary and multiclass classification. The paper discusses the choice of such a machine learning algorithm, which depends on several factors, including the size of the data, their quality and variety, as well as understanding what answers based on this data the business needs. For this reason, you have to try many different algorithms, checking the effectiveness of each on the test data set, and then choose the best option. Given this, you need to choose among the existing algorithms, those that meet this problem. The authors of the article paid attention to the accuracy, time for training, parameters, data. Therefore, choosing the right algorithm is a combination of business needs, specifications, experimental work and accounting for available time. The introduction of machine learning methods in various fields was studied. The process of machine learning is presented, which includes the following stages: data preparation, creation of training sets, creation of the classifier, training of the classifier, drawing up of forecasts, estimation of productivity of the classifier and adjustment of parameters. Analyze the use of different classification algorithms using the Python library, Scikit-learn, analyze the use of the method of model selection, calculation, formatting and data preparation, select the optimal input values and models. The estimation of several variants of a classifier estimation is resulted. The aim of the work is to study the library for the effectiveness of its practical application. The paper presents methods of classification in machine learning using the Scikit-Learn library. A comparison of different classification methods using Scikit-learn for machine learning models is performed.

Key words: *machine learning, classification, optimization, data analysis, forecasting.*