Методы, библиотеки и среды для анализа данных с использованием машинного обучения

М. С. Загорникова, С. А. Лукашевич
Гомельский государственный университет им. Ф. Скорины, г. Гомель, Беларусь,
mszagornikova@gmail.com

Аннотация. *В статье рассматриваются современные подходы к анализу данных с использованием методов машинного обучения. Основное внимание уделено программным инструментам и библиотекам Python: Anaconda, NumPy, Pandas, Scikit-learn, TensorFlow и PyTorch. Подробно описаны ключевые методы анализа данных: отбор признаков, кросс-валидация и настройка гиперпараметров моделей. Особое внимание уделено методам визуализации результатов, включая линейные графики, тепловые карты и 3D-визуализации.*

I. Введение

Современный анализ данных, особенно в научных исследованиях, всё чаще опирается на методы машинного обучения. Этот процесс требует не только теоретических знаний, но и владения специализированными инструментами. В данной статье рассматриваются ключевые аспекты работы с данными: от выбора программного обеспечения до методов визуализации результатов.Особое место в процессе анализа данных занимает язык программирования Python, который благодаря своей простоте и богатой экосистеме библиотек стал стандартом в области анализа данных. Разработанный в конце 1980-х годов и впервые опубликованный в 1991 году, Python предлагает исследователям мощные средства для автоматизации обработки данных.

Anaconda – дистрибутив Python, который включает в себя множество популярных библиотек для анализа данных и имеет удобный пользовательский интерфейс. Преимущество Anaconda состоит в том, что все библиотеки идут комплектом и не конфликтуют между собой.

Далее рассмотрим библиотеки, которые используют для анализа данных. NumPy – это библиотека с открытым исходным кодом для языка программирования Python. Она предлагает эффективный многомерный контейнер данных и предоставляет возможность определять произвольные типы данных. Является основной математической библиотекой для работы с данными, положенная в основу других библиотек для работы с задачами машинного обучения или анализа данных: Pandas (работа с табличными данными), SciPy (методы оптимизации и научные расчеты), Matplotllib (построение графиков) [1]. Scikit-learn – это популярная библиотека для машинного обучения на языке Python. Она предоставляет простые и эффективные инструменты для анализа данных и построения моделей машинного обучения. Библиотека включает в себя множество алгоритмов для классификации, регрессии и кластеризации, таких как линейные модели, деревья решений, случайные леса, SVM, K-ближайших соседей и многие другие [2]. Scikit-learn предлагает инструменты для предобработки данных, включая нормализацию, стандартизацию, кодирование категориальных переменных и обработку пропущенных значений. Библиотека предоставляет функции для оценки качества моделей. Поддерживает создание пайплайнов, что позволяет объединять несколько шагов обработки данных и обучения модели в один объект, упрощая процесс разработки. Библиотека хорошо интегрируется с другими популярными библиотеками Python, такими как NumPy, SciPy и Matplotlib, что делает её удобной для использования в научных и исследовательских проектах. TensorFlow и PyTorch – это библиотеки для глубокого обучения. Они позволяют создавать и обучать сложные нейронные сети. TensorFlow разработан компанией Google. Ориентирован на графы вычислений. Модель сначала строится в виде графа, а затем выполняется. Широко используется в промышленности и для разработки масштабируемых приложений. Имеет хорошую поддержку для мобильных и веб-приложений через TensorFlow Lite и TensorFlow.js. Предоставляет высокоуровневый API, что упрощает создание и обучение моделей. Имеет обширную документацию и множество обучающих материалов. PyTorch разработан Facebook

(Meta). Ориентирован на динамические вычислительные графы. Позволяет изменять граф во время выполнения, что делает его более гибким.

II. Алгоритмы и реализация

Эффективная обработка физических данных методами машинного обучения требует применения специализированных методик, направленных на построение качественных моделей и оценку их результатов. Одним из ключевых этапов предобработки является отбор информативных признаков, то есть идентификация наиболее значимых признаков из сотен или тысяч доступных переменных. Это позволяет решать проблемы избыточности и шума, сокращает время обучения, а также повышает точность и интерпретируемость моделей. Для отбора признаков используются различные методы, включая фильтрацию (автоматическое удаление признаков с высоким процентом пропусков, низкой вариативностью или сильной корреляцией, например, через библиотеку feature-selector), встроенные методы (интеграция отбора в алгоритмы обучения, например, L1-регуляризация), а также оберточные методы, предполагающие итеративную оценку важности признаков с использованием моделей, таких как LightGBM.

После отбора признаков важным этапом является кросс-валидация, которая служит методом оценки устойчивости модели на независимых данных и предотвращает переобучение. Наиболее распространённой является K-кратная кросс-валидация, при которой исходные данные делятся на k частей (обычно 5–10 сегментов), и последовательно каждая часть используется как тестовая выборка, а остальные — для обучения. Итоговая оценка формируется усреднением метрик, по всем итерациям.

Завершающим критически важным этапом является оптимизация гиперпараметров, то есть настройка конфигурационных параметров, задаваемых до обучения (например, архитектура сети, скорость обучения, регуляризация). В отличие от параметров модели, которые представляют собой внутренние веса, оптимизируемые в процессе обучения (например, коэффициенты нейронных связей), гиперпараметры настраиваются заранее. Поиск оптимальных значений гиперпараметров обычно осуществляется с помощью методов GridSearch или RandomizedSearch для достижения максимальной производительности модели.

Наглядное представление данных -- важнейший этап анализа. Линейные графики идеально подходят для демонстрации изменений во времени, а столбчатые диаграммы помогают сравнивать разные категории. Для выявления взаимосвязей между переменными используют диаграммы рассеяния как показано на рисунке 1.

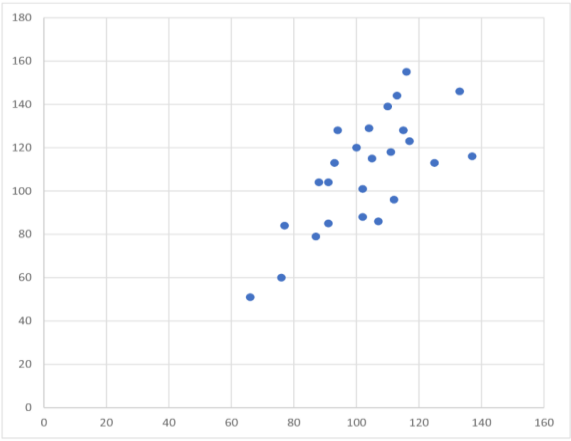

Рис. 1. Диаграмма рассеяния

Тепловые карты отлично показывают структуру матричных данных как показано на рисунке 2.

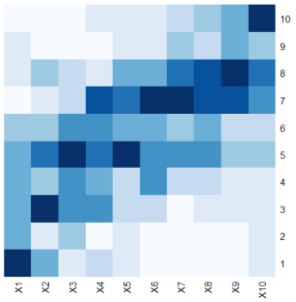

Рис. 2. Тепловая карта

В последние годы всё большую популярность приобретают интерактивные и трёхмерные визуализации (рисунок 3). Они позволяют исследователям буквально "погружаться" в данные, рассматривая их с разных сторон и выявляя скрытые закономерности.

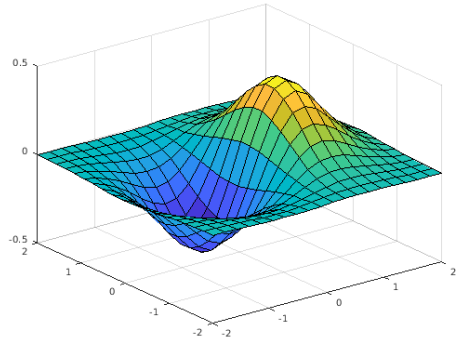

Рис. 3. Трёхмерная визуализация

III. Заключение

Современный анализ данных – это сложный, но увлекательный процесс, требующий комплексного подхода. От выбора инструментов до интерпретации результатов - каждый этап важен для получения достоверных выводов. Использование Python и его библиотек в сочетании с грамотным применением методов машинного обучения открывает широкие возможности для исследователей в самых разных областях науки и практики.

Литература

[1] *И. Шамаев* Язык программирования Python 3. Обзор библиотек принципов modules - Python 3 | Data Science | Нейронные сети | AI - Искусственный Интеллект. Доступно по ссылке: *https://python.ivan-shamaev.ru/overview-python-programming-language-modules-library-principles/*

[2] Skypro. Фреймворк и программа искусственного интеллекта. Доступно по ссылке: *https://sky.pro/wiki/python/frejmvork-i-programma-iskusstvennogo-intellekta/*