

Машинное обучение (ML)  
где применяется цели и за-  
дачи, с помощью чего учат-  
ся машины



Мы живем в эпоху научно технического прогресса, люди желая облегчить свой труд используют машины, различные компьютерные программы. Также изобретён Искусственный Интеллект (ИИ) который помогает во многих областях жизни.

Для того чтобы машины нам помогали необходимо машинное обучение.  
Что это такое?

Машинное обучение (ML) — это область искусственного интеллекта, которая занимается разработкой алгоритмов и статистических моделей, позволяющих компьютерам выполнять конкретные задачи без явного программирования на выполнение этих задач. Основная идея состоит в том, чтобы создать модель, способную "учиться" на данных и делать прогнозы или принимать решения на основе этого обучения.

*Области в кото-  
рых применяется  
искусственный  
интеллект*



Финтех: кредитный скоринг, защита клиентов, поиск мошенников







Медицина:  
диагностика  
болезней на ранней  
стадии





## Промышленность: контроль качества







Автомобили и машины без  
водителей



# Фармакология: проектирование лекарств





# Информационная безопасность



# Безопасность и охрана людей



Еще можно упомянуть

- Системы умный дом
- Распознавание голоса и голосовые помощники
- Словари типа T9
- Аграрно промышленный комплекс то не остается без внимания.

Можно автоматизировать производство или сделать жизнь комфортнее .



Для любой машины нужна программа для машинного обучения тоже. Для того чтобы облегчить процесс написания программы были созданы библиотеки: scikit-learn, TensorFlow и PyTorch

# **Сравнение различных библиотек для машинного обучения:**

**scikit-learn, TensorFlow и PyTorch:**

Реализовать задачи классификации и регрессии с использованием scikit-learn, TensorFlow и PyTorch, сравнить их производительность и удобство использования.



Scikit-learn (sklearn) — это один из наиболее широко используемых пакетов Python для Data Science и Machine Learning. Он содержит функции и алгоритмы для машинного обучения: классификации, прогнозирования или разбивки данных на группы. Sklearn написана на языках Python, C, C++ и Cython.





TensorFlow — это библиотека для машинного обучения, группы технологий, которая позволяет обучать искусственный интеллект решению разных задач. Библиотека изначально разработана для Python и чаще всего используется с ним.



PyTorch — это фреймворк для языка программирования Python, предназначенный для машинного обучения. Он включает в себя набор инструментов для работы с моделями, используется в обработке естественного языка, компьютерном зрении и других похожих направлениях.

## Scikit-learn против TensorFlow

Scikit-learn — широко используемая библиотека машинного обучения с открытым исходным кодом для Python. Она построена на основе и интегрируется с такими часто используемыми библиотеками, как NumPy, SciPy, Matplotlib и pandas, что делает ее доступной и универсальной.

TensorFlow , также библиотека машинного обучения с открытым исходным кодом, специализируется на глубоком обучении и нейронных сетях. TensorFlow поддерживает несколько языков программирования, таких как Python, C/C++, Java и Javascript, и другие.

Рассмотрите scikit-learn, если вы новичок в машинном обучении или разрабатываете что-то с использованием алгоритмов, не основанных на нейронных сетях.

Рассмотрите TensorFlow, если вы хотите использовать подход глубокого обучения в сочетании с аппаратным ускорением с помощью графических процессоров и TPU или на кластере компьютеров (что scikit-learn изначально не поддерживает).



## PyTorch против scikit-learn

PyTorch — это библиотека программного обеспечения глубокого обучения для Python, C++ и Julia. PyTorch в основном используется для сквозного построения и обучения глубоких нейронных сетей с возможностью создания пользовательских моделей и алгоритмов обучения.

Scikit-learn — это библиотека для традиционных алгоритмов машинного обучения, используемых для кластеризации, классификации, регрессии и т. д. Природа Scikit-learn как «черного ящика» делает ее более доступной для тех, кто относительно недавно начал заниматься машинным обучением.

Рассмотрите PyTorch, если вы разрабатываете приложения, в которых есть задачи, требующие больших вычислительных затрат, такие как обработка естественного языка, компьютерное зрение и т. д. С Pytorch вы также можете использовать ускорение графического процессора в своих интересах.

Рассмотрите scikit-learn, если вы разрабатываете небольшой исследовательский проект, не требующий значительного объема данных. С scikit-learn вы будете фокусироваться не на настройке, а на скорости и удобстве использования алгоритмов машинного обучения.

## PyTorch против TensorFlow

PyTorch — это фреймворк глубокого обучения с питоническим и объектно-ориентированным подходом. PyTorch имеет больше возможностей отладки и тестирования, чем TensorFlow.

TensorFlow — это низкоуровневая библиотека глубокого обучения, которая обеспечивает рабочие процессы для высокоуровневых API, таких как Keras, хотя и с меньшей вычислительной мощностью. TensorFlow в настоящее время используется более широко, чем PyTorch.

Если Python играет центральную роль в вашей разработке, рассмотрите PyTorch с его многочисленными возможностями отладки.

Рассмотрите TensorFlow, если вам нужна библиотека, совместимая с различными языками программирования, такими как C/C++, Java, JavaScript, Go и т. д. TensorFlow также имеет обширную поддержку нескольких платформ.

## 1. Производительность:

- Все модели могут показать конкурентоспособные результаты на простых задачах, но более сложные модели и большие наборы данных могут требовать больше времени для обучения в TensorFlow и PyTorch по сравнению с scikit-learn.
- TensorFlow и PyTorch предоставляют более мощные инструменты для настройки архитектуры модели, что позволяет достигать лучшей производительности на сложных задачах.



## 2. Удобство использования:

- scikit-learn: Отличается простым и интуитивно понятным интерфейсом. Идеально подходит для быстрого прототипирования и работы с классическими алгоритмами машинного обучения.
- TensorFlow: Обеспечивает высокий уровень гибкости для создания сложных моделей глубокого обучения, но требует больше кода и времени для настройки.
- PyTorch: Более удобен в использовании, особенно для исследования, благодаря динамической вычислительной графике и гибкости. Он также упрощает отладку.

## Заключение

Сценарий выбора библиотеки зависит от задачи. Если вы ищете простоту и скорость, `scikit-learn` может быть лучшим выбором для традиционных методов. Если ваши задачи связаны с глубоким обучением, `TensorFlow` и `PyTorch` предпочтительнее, причём `PyTorch` может быть удобнее на этапе prototyping и исследований.