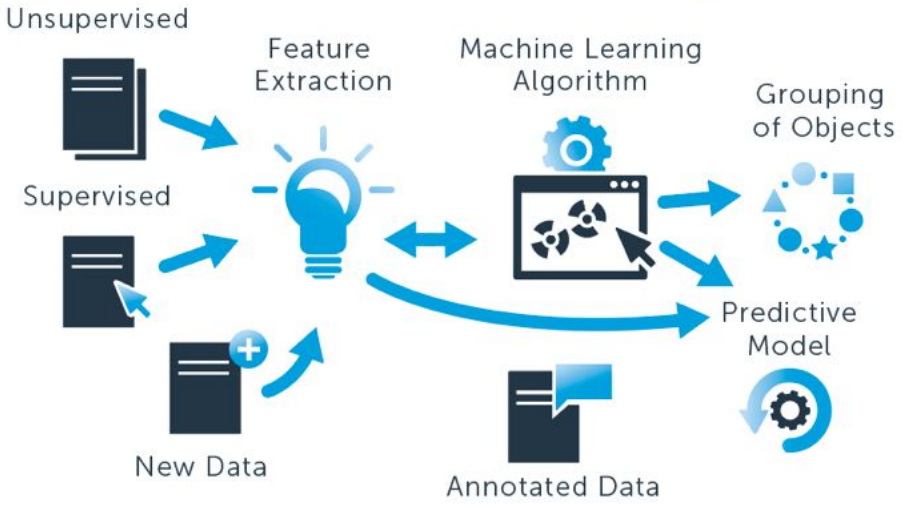
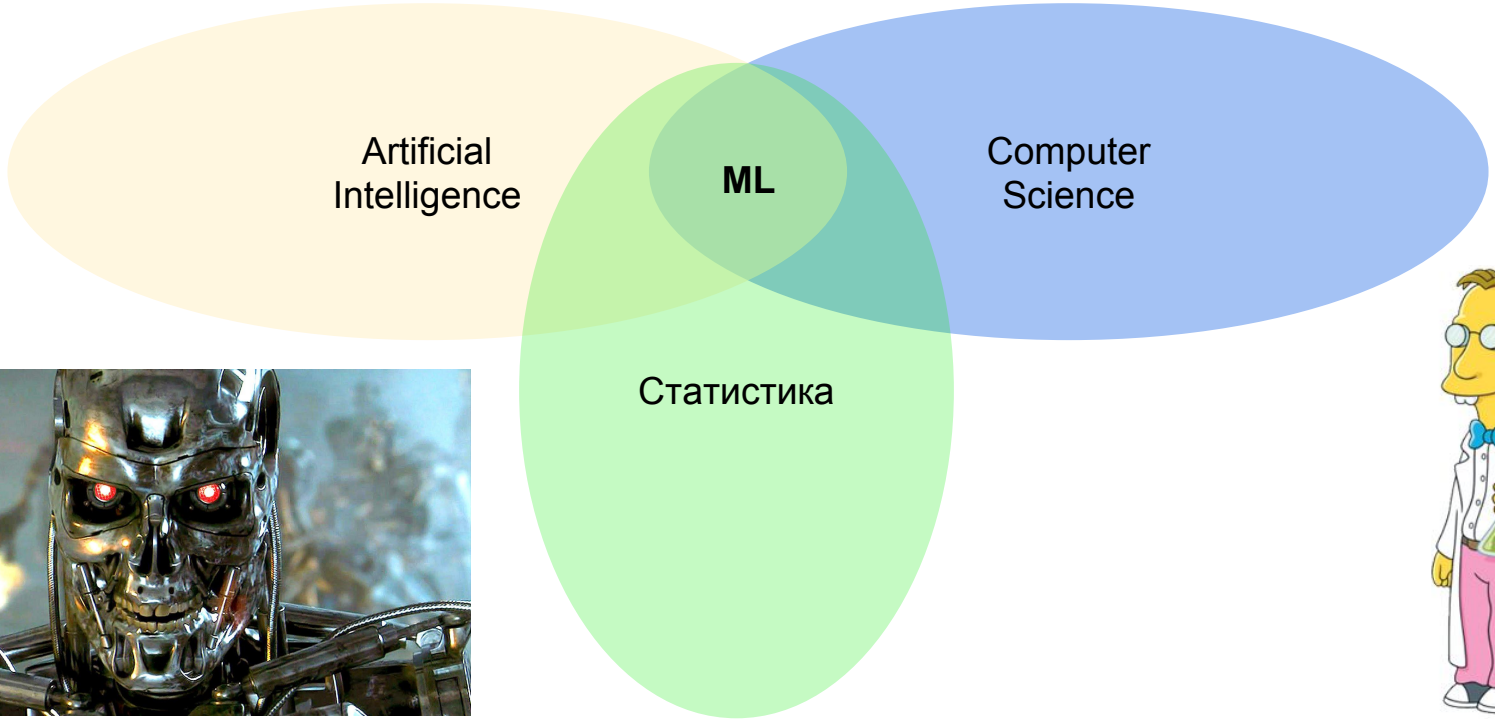


Введение в Machine Learning

Антон Труханёнок | atrukhanyonok@objectstyle.com

План

Machine Learning vs. Artificial Intelligence



Что такое Machine Learning?

Решение задачи находится путём обучения **модели** на данных.
Модель инициализируется случайными значениями, затем обучается циклически.

1. Предсказать.
2. Посчитать функцию ошибки.
3. Изменить параметры модели, чтобы уменьшить ошибку в следующей итерации.

Обучение завершается, когда ошибка перестаёт уменьшаться от итерации к итерации.

В противовес “классическому” программированию, алгоритм решения не задан явно.

Категории алгоритмов Machine Learning

- **Supervised Learning**

  Модель учится на размеченных данных, затем предсказывает на неразмеченных.
  - **Регрессия** – исходящие данные непрерывны (числа).
  - **Классификация** – исходящие данные дискретны (классы).
- **Unsupervised Learning**

  Данные не размечены “правильными” ответами. Алгоритм пытается скомпоновать входящие параметры самостоятельно. Нет правильного ответа.
  - **Кластеризация**.
  - **Детектор аномалий**.
  - **Dimensionality Reduction** (уменьшение количества измерений).
- **Reinforcement Learning**

  Алгоритм получает на входе сигналы и выдаёт ответ, затем получает “награду” в виде числа. Чем больше число, тем “лучше” ответ. Процесс повторяется циклически. В конце считается суммарная награда.

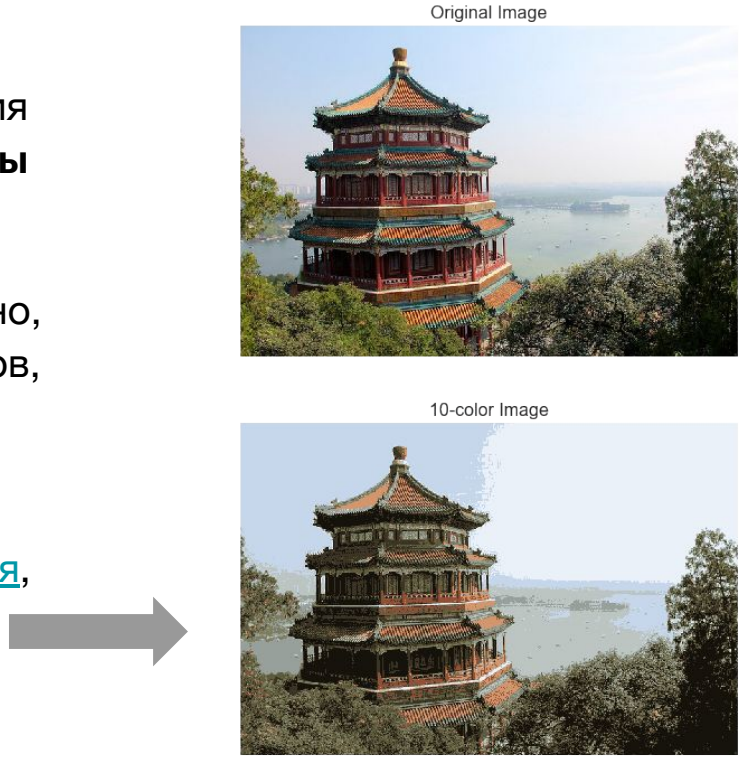
Примеры задач Supervised Learning

Предсказать на основе имеющихся правильных значений:

- **Регрессия**
  - Сколько стоит недвижимость в заданном районе Бостона, зная уровень преступности, количество промышленных предприятий, расстояние до центра и т.д.
- **Классификация**
  - Выжил ли пассажир Титаника, зная его пол, класс билета, возраст и т.д.
  - Что на фото: кот или собака?
  - Является ли данный текст отзыва на tripadvisor положительным или отрицательным?
- **Временные ряды**
  - Предсказать продажи шампанского в течение следующих 12 месяцев, зная продажи в каждый месяц в течение предыдущих 3-х лет.

Примеры задач Unsupervised Learning

- Выделить **целевые группы** для таргетирования рекламы. Для каждой группы найти **центроиды** (наиболее репрезентативные **кластеры**).
- Разделить набор картинок на 5 групп. Известно, что на картинках изображены 5 типов объектов, но нет размеченных картинок с этими объектами.
- Произвести **компрессию цветов** изображения, уменьшив палитру с 16 миллионов до 10 наиболее значимых цветов.



Пример задачи Reinforcement Learning

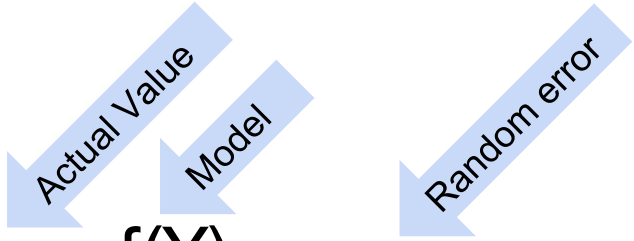
Написать бота, который проведёт мышь по неизвестному лабиринту с максимальными итоговыми **очками**.

За свои **действия** мышь получает очки:

- движение: -1,
- молния: -100,
- вода: +10,
- сыр в конце лабиринта: +1000.

Источник:
[Machine Learning for Humans by Vishal Maini]

Данные в задаче регрессии



$$y = f(X) + err$$

$X = (x_1, x_2, \ldots, x_p)$ — p-мерный вектор features / predictors

$X^1, X^2, \ldots, X^n$ с известными $y^1, y^1, \ldots, y^n$ — Training Dataset размера n

$X^1, X^2, \ldots, X^m$ с неизвестными y — Testing Dataset размера m

Модель линейной регрессии

$y = \theta_0 + \theta_1 * x_1^i + \theta_2 * x_2 + \ldots + \theta_p * x_p$

Функция ошибки (cost / error / loss function):
Среднеквадратическое отклонение (Mean Squared Error, MSE)

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{m} (f_\theta(x^{(i)}) - y^{(i)})^2$$

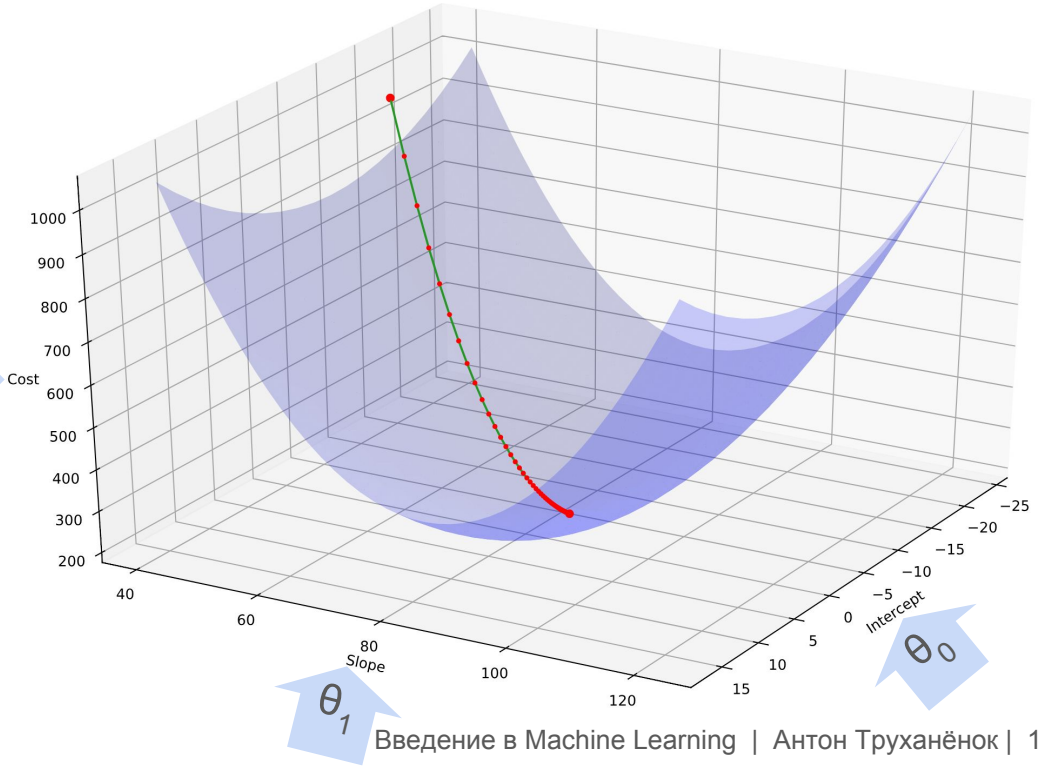
Цель: подобрать коэффициенты $\theta = (\theta_0, \theta_1, \theta_2, \ldots, \theta_p)$, чтобы минимизировать ошибку на тренировочном датасете.

Градиентный спуск в линейной регрессии

Рассмотрим случай одномерного X:
$y = \theta_0 + \theta_1 * x_1$

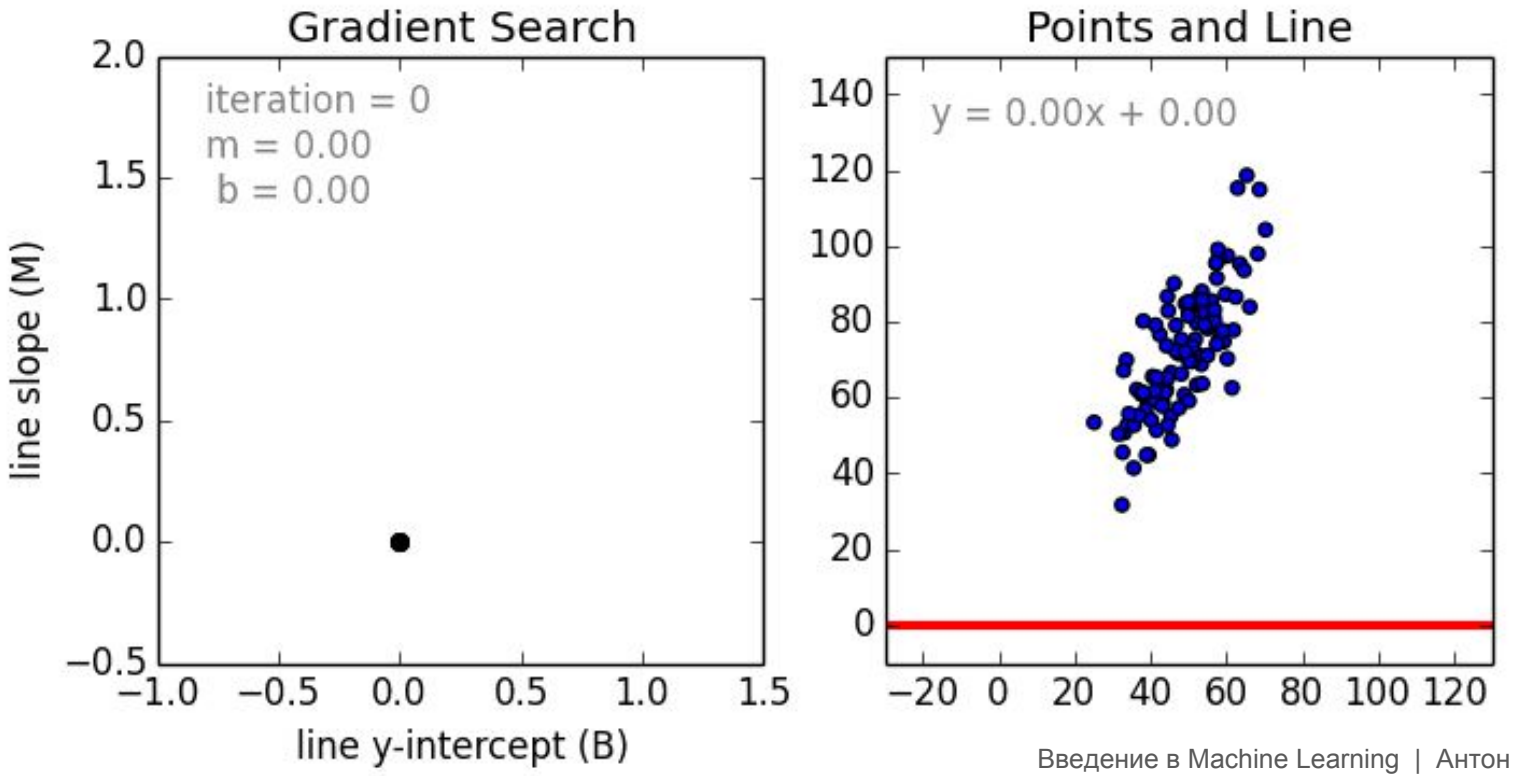
Будем пошагово изменять вектор θ направлении, противоположном его градиенту, пропорционально гиперпараметру η (скорость обучения).

$$\theta := \theta - \eta \nabla_\theta J(\theta) = \theta - \eta \sum_{i=1}^{m} \nabla J_i(\theta)$$



Источник: AM207 Course

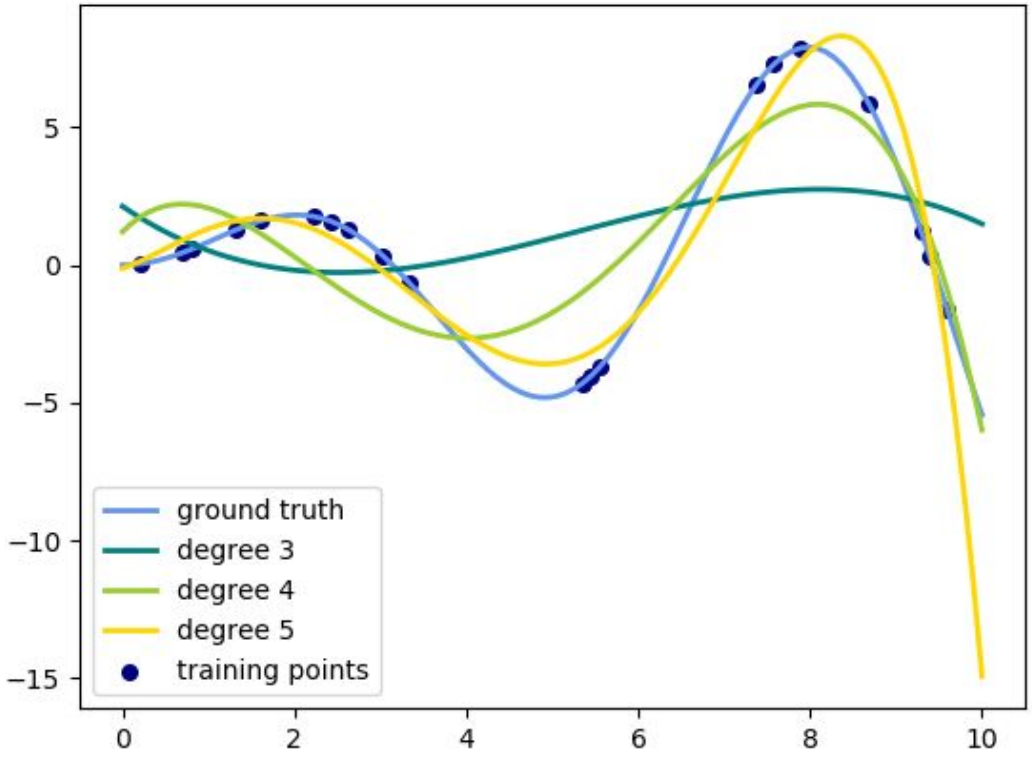
Пример градиентного спуска



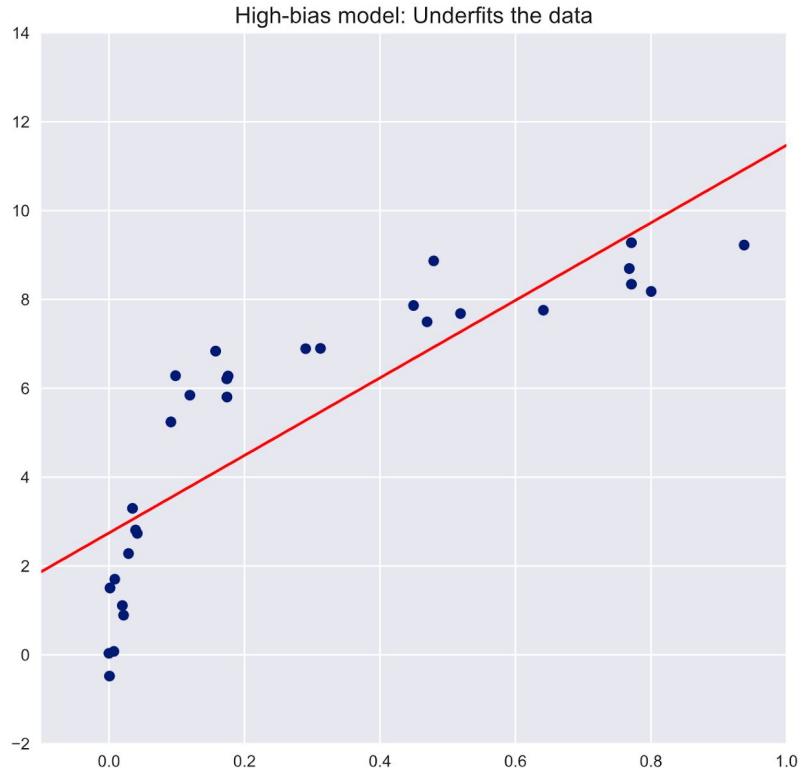
Полиномиальные признаки (Polynomial Features)

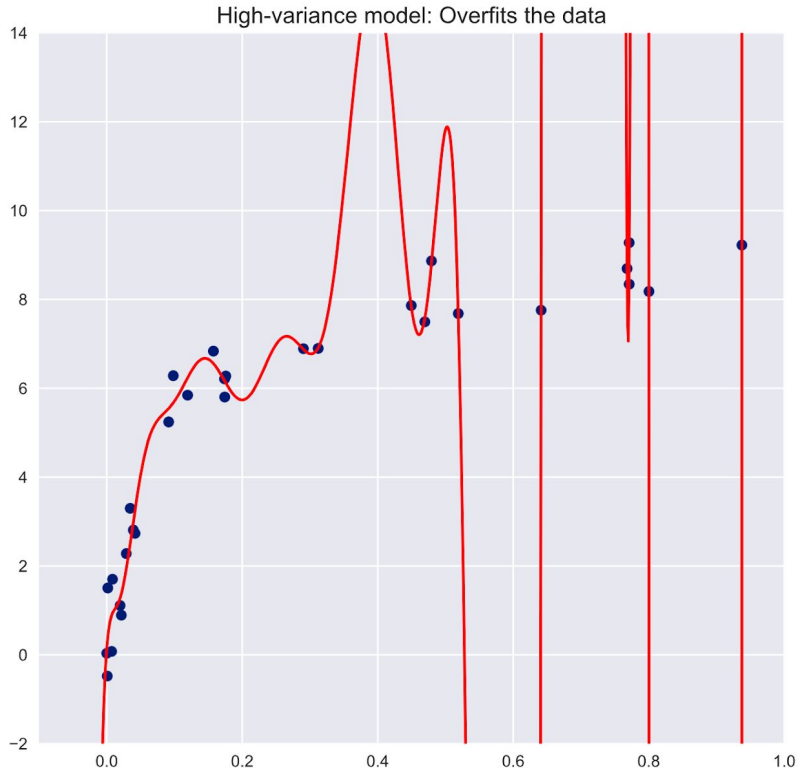
Добавим новые признаки, представляющие собой полиномы второй степени от имеющихся признаков:

$$y = \theta_0 + \theta_1 * x_1 + \theta_2 * x_2 + \theta_3 * x_1 * x_2 + \theta_4 * x_1 * x_1 + \theta_5 * x_2 * x_2$$



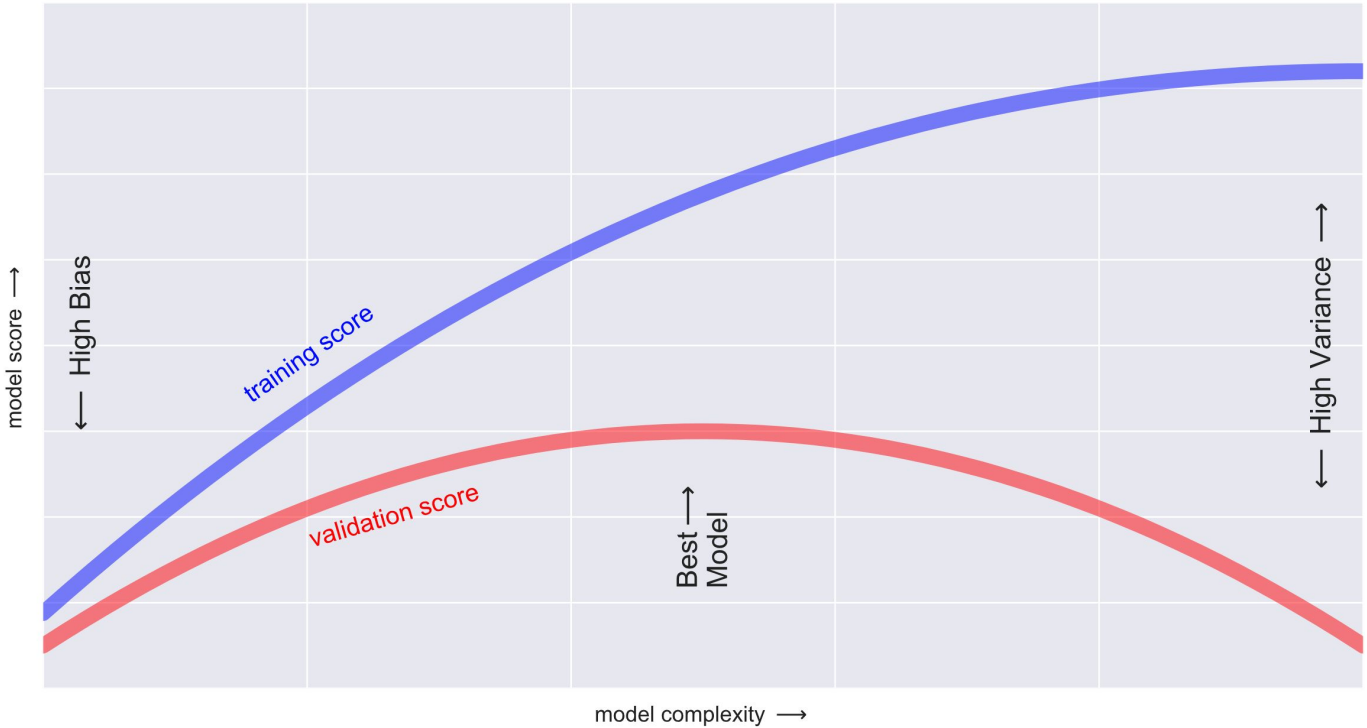
Bias – Variance Tradeoff, Underfitting vs. Overfitting





Источник: [Python Data Science Handbook by Jake VanderPlas]

Bias – Variance on Validation Curve

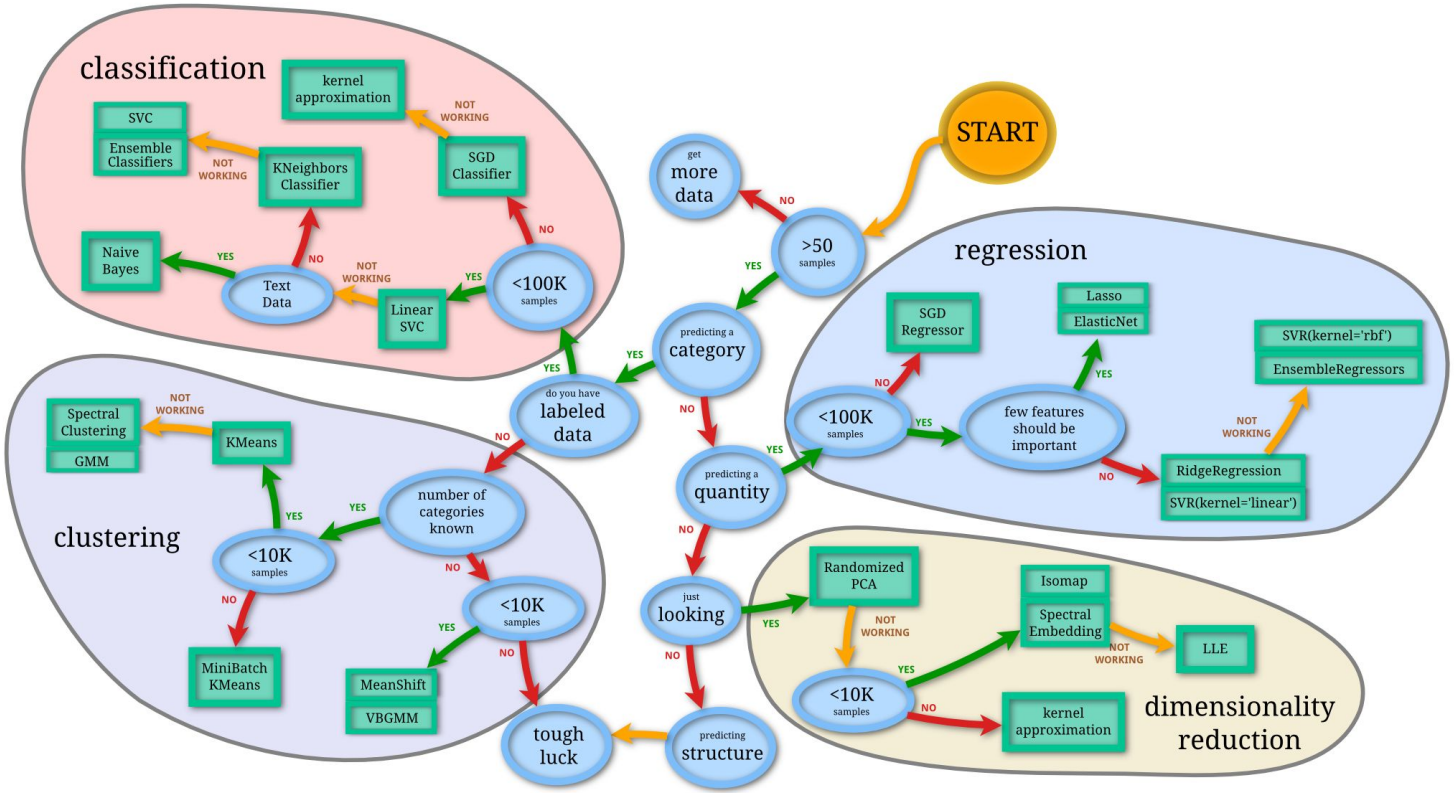


Источник: [Python Data Science Handbook by Jake VanderPlas]

Supervised ML процесс

1. Подготовить данные. Feature engineering.
2. Выбрать несколько классов моделей и инициализировать их какими-нибудь значениями гиперпараметров.
3. Разделить данные на тренировочные, валидационные и тестовые.
4. Обучить модель на тренировочных данных.
5. Посчитать функцию ошибки на тренировочных и валидационных данных.
6. Изменить гиперпараметры модели. Пройти пункты 4 - 5 с разными гиперпараметрами и разными классами моделей.
7. Выбрать лучшую модель и её гиперпараметры.
8. Проверить выбранную модель на тестовых данных.

SciKit Learn Algorithm Cheat-sheet



Источник: SciKit Learn

Инструменты

Python + Conda package manager.

Jupyter Notebook – среда разработки.

Numpy, Pandas – работа с данными.

SciKit Learn – готовые ML алгоритмы и вспомогательные инструменты.

Matplotlib – визуализации.

Рекомендации

OCDevel Machine Learning Guide podcast
ocdevel.com/mlg

Python Data Science Handbook by Jake VanderPlas
github.com/jakevdp/PythonDataScienceHandbook

Курс от сообщества OpenDataScience
habr.com/company/ods/blog/322626

Добавляйтесь в OpenDataScience Slack
ods.ai

Участвуйте в Kaggle Competitions
kaggle.com/challenge-yourself