

Машинное обучение



Что это?

Определение с сайта machinelearning.ru:

Машинное обучение (Machine Learning) — обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться.

Менее формально: о том, как с помощью статистики учить компьютер самостоятельно делать выводы о данных.



Обобщающая способность

Обучившись на имеющихся данных, наша модель умеет делать предсказания о новых данных.

Пример: понять, является ли спамом новое сообщение на почте



Зачем оно нужно?

- человеческое время -- дорогой ресурс
- есть рутинная работа, с которой справится и компьютер
- на какие-то типы задач человеку просто не хватит времени (найти закономерность, проверив влияние тысяч факторов)
- сложные взаимодействия факторов, которые трудно увидеть невооружённым глазом

Чаще всего МО -- про экономию человеческих ресурсов.




Задачи МО



Виды МО, глобально

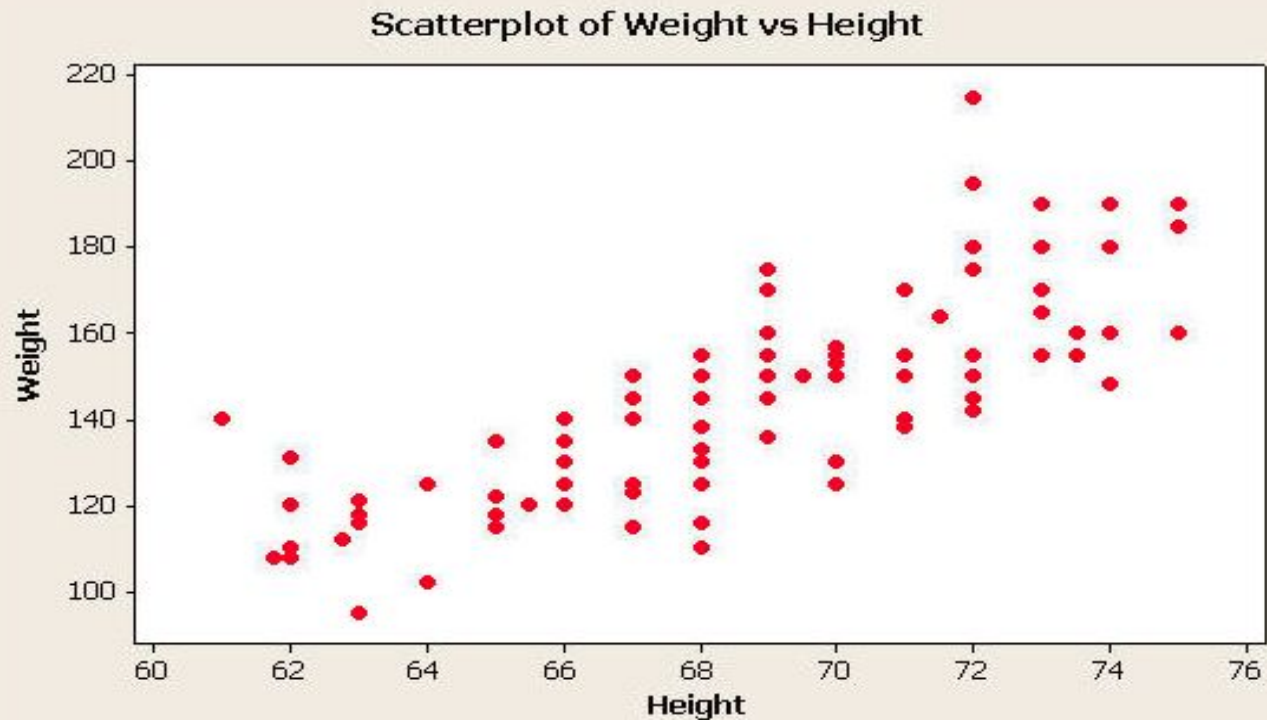
- обучение с учителем (= у нас есть размеченные данные); supervised learning
- обучение без учителя (= есть много неразмеченных данных); unsupervised learning
- некоторые промежуточные виды

Размеченные данные -- дорогой ресурс, потому что требуют работы человека.

A decorative graphic on the left side of the slide. It consists of a blue parallelogram and a light green parallelogram, both tilted at an angle. The blue shape is in the foreground, and the green shape is partially behind it. They are set against a dark blue background with faint, lighter blue diagonal stripes.

Обучение с учителем

Подбор закономерностей





Регрессия

- предсказывает какое-то значение по набору признаков
- допустимым ответом является действительное число или числовой вектор
- модель учится наилучшим образом приближать данные
- примеры:



Регрессия

- предсказывает какое-то значение по набору признаков
- допустимым ответом является действительное число или числовой вектор
- модель учится наилучшим образом приближать данные
- примеры: предсказать сложность текста для восприятия по количеству символов/слов в нём



Классификация

- есть несколько классов
- есть множество объектов, каждый из которых принадлежит к одному классу
- для ряда объектов известно, к какому классу они принадлежат (это обучающая выборка)
- нужно построить алгоритм, способный отнести произвольный (новый) объект из этого множества к тому или иному классу
- примеры: ...

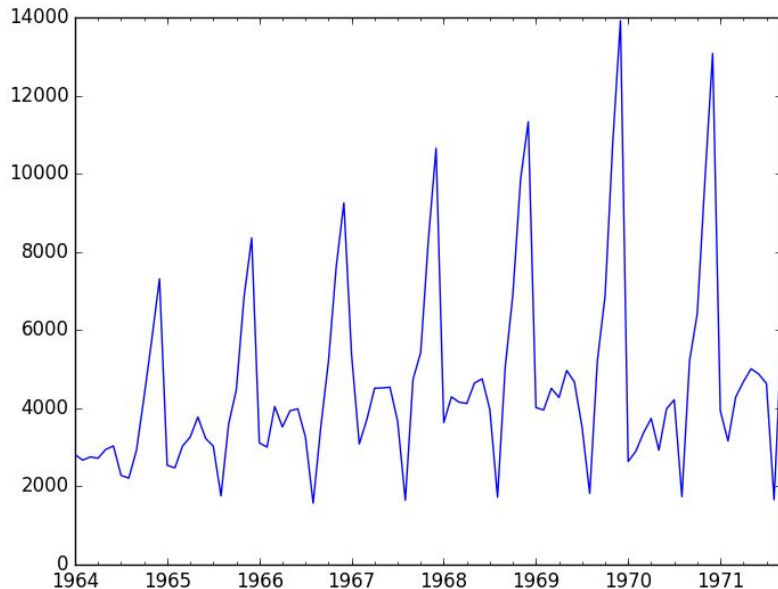



Классификация. Примеры.

- определение части речи (POS-тэггинг)
- определить, является ли э-мэйл спамом
- снятие семантической омонимии (Word Sense Disambiguation)
- определение языка

Другие виды

- ранжирование (например, поисковики)
- прогнозирование временных рядов, например:





Обучение без учителя



Кластеризация

- у нас есть выборка объектов, но нет заданных классов
- мы хотим разбить их на группы так, чтобы объекты в разных группах сильно отличались
- примеры?



Кластеризация

- у нас есть выборка объектов, но нет заданных классов
- мы хотим разбить их на группы так, чтобы объекты в разных группах сильно отличались
- примеры: группировка новостей по темам, группировка запросов (например, к боту)



Снижение размерности

- часто приходится иметь дело с данными больших размерностей
- такие данные сложно хранить и обрабатывать (требуют больших вычислительных мощностей и много RAM)
- есть методы, которые позволяют сильно снижать размерность, выделяя наиболее значимые компоненты
- ещё одно применение -- визуализация в 2D



Что ещё?

Вспомним векторную семантику и word2vec.

Мы научили компьютер понимать, какие слова близки друг к другу по значению, используя неразмеченные данные (просто много собраний текста).



Промежуточные типы

- обучение с подкреплением (= алгоритм получает ответ (подкрепление) от внешней среды); reinforcement learning
- активное обучение (= мы на ходу понимаем, какие данные надо дособрать); active learning
- частичное обучение (= размечена только часть данных); semi-supervised learning
- больше типов -- [здесь](#)



Фичи

(признаки)



Как скормить текст машине [2]

- У нас есть наш прекрасный предобработанный текст
- Но есть проблема – машины понимают только числа
- Задача: преобразовать текстовую информацию в числа
- Как?



Какие бывают переменные

- бинарные (1 или 0)
- численные (любое число)
- категориальные (жёлтый/белый/зелёный)
- порядковые (бакалавриат/магистратура/PhD)



One-hot encoding

Способ кодирования любой категориальной переменной

feature		feature_A	feature_B	feature_C
A	→	1	0	0
B		0	1	0
C		0	0	1
B		0	1	0
C		0	0	1



Векторизаторы

Count Vectorizer – для каждого текста в столбцах слов количество раз, сколько слово встретилось в тексте (+ n-граммы)

feature		feature_A	feature_B	feature_C
A B	→	1	1	0
B C A C		1	1	2
B C B B		0	3	1
B A		1	1	0
C		0	0	1

TF-IDF Vectorizer – вместо абсолютных частот значения TF-IDF



Векторизаторы

Проблемы:

- Умеют только в те слова, что видели (если не видели слово – вектор, где все нули)
- Очень много нулей (в каждом тексте встречается лишь малая часть слов из словаря) -> разреженные матрицы (sparse matrices)
- Размерность (700 тыс фичей на 70 тыс объектов)



Семантика

Вместо one-hot encoding-like векторизаторов взять word2vec (doc2vec)

Плюсы:

- не такая большая размерность (100 или 300)
- обобщение информации на уровне семантики (нет привязки к конкретным словам)

Минусы:

- Очень много весит (300 float-32 для миллиона объектов – смерть и sparse тут не поможет)
- Проблема незнакомых слов на уровне w2v модели



Уменьшение размерности

Актуально, особенно когда у нас 700 тыс фичей

Много фичей:

- шум
- модель долго обучается

Выход:

- выбрать самые важные фичи (feature selection)
- сделать новые фичи на основе старых (feature extraction)



Уменьшение размерности

Feature selection

- Метрики важности
- Модели МО с L1-регуляризацией, обнуляющие неважные признаки

Feature extraction

Перепроецировать признаки на другую систему координат

- PCA
- SVD



Что еще использовать

- POS-тэги
- Синтаксис:
 - какой глагол управляет этим существительным?
 - какой тип связи между глаголом и этим существительным?
- Семантика
 - гиперонимы (более общая сущность, надмножество)
 - семантические отношения




Фичи – это важно

Предобработка – очень важно

Ключ к успеху

Важнее моделей

В первую очередь думайте о фичах и о том, что из них можно
вытащить и что интересного сделать



Оценка качества



Регрессия

MSE - Mean squared error - среднеквадратичная ошибка

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2.$$

MAE - Mean absolute error - средняя абсолютная ошибка

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

Классификация

Accuracy (аккуратность, экьюраси) - доля правильных ответов

Precision (точность) - сколько объектов, отнесенных моделью к классу X, действительно относятся к классу X

Recall (полнота) - сколько объектов, действительно относящихся к классу X, отнесены моделью к классу X.

F1-score (F1-мера) - гармоническое среднее между precision и recall

еще есть **AUC-ROC** - площадь под кривой ROC, основана также на вот этой табличке, чем ближе к 1, тем лучше

		Condition (Gold standard)	
		True	False
Test outcome	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$



Train / validation / test

Нам нужно понимать, насколько качественно работает и будет работать (!) наша машина.

Как оценивать - понимаем (пред. слайды)

А на чем?



Train / validation / test

Чтобы понять, как будет работать модель на данных, которых не было в обучающей выборке - от данных сразу отрезают и откладывают в сторону **тестовую** выборку.

Модели нужно настраивать -> от обучающей выборки отрезают **валидационную** выборку, на которой тестируются разные настройки.

Кроссвалидация:

- делим обучающую выборку на несколько частей (фолдов)
- несколько раз обучаем
- при каждом обучении один фолд валидационный, остальные обучающие
- Результат усредняем