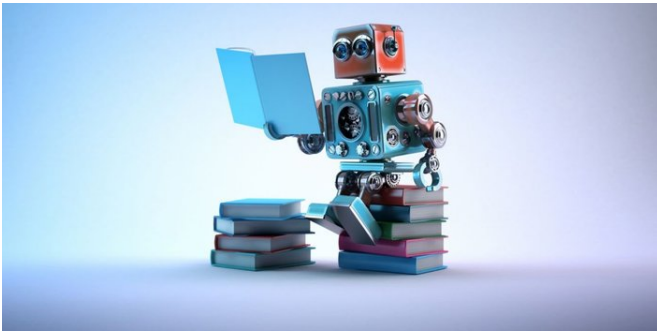
План

1. Recap
2. Терминология машинного обучения
3. Метрические алгоритмы
4. Линейные модели

Recap

1. Что такое машинное обучение? Какие примеры можете привести?
2. Что такое обучение с учителем?
3. Что необходимо знать, чтобы заниматься машинным обучением?
4. Что такое модель? Какие ошибки могут возникать в процессе построения модели?

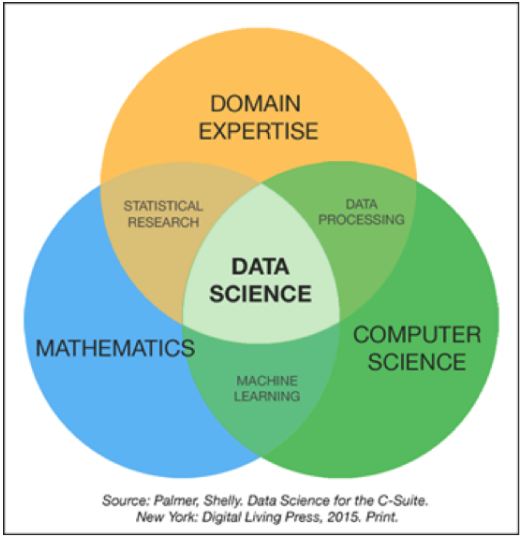


Что такое машинное обучение

Машинное обучение (machine learning, ML) – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач **Что это означает:**

- Есть некоторая задача, например
  - поиск текстов
  - распознавание изображений
  - прогноз цен на акции
  - написание стихов
- Человек умеет решать задачу хорошо, но делает это очень медленно
- Хочется на примере человека научить компьютер решать эти задачи намного быстрее с небольшой потерей качества

Машинное обучение и анализ данных



Задача обучения с учителем

Необходимо найти закономерности в имеющихся прецедентах и обобщить на объекты, для которых ответы неизвестны
Имеются:

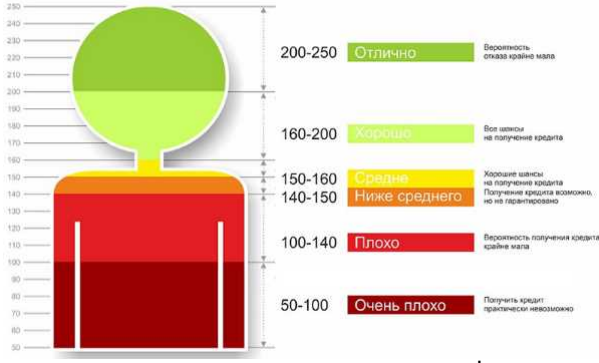
- Множество объектов (ситуаций) X
- Множество возможных ответов (откликов, реакций) Y
- Между X и Y существует зависимость $a : X \to Y$, известная на конечной выборке **прецедентов** (x_i, y_i) – парах "объект-ответ"
- Множество прецедентов называется обучающей выборкой X_{train}. На основе имеющихся прецедентов необходимо построить алгоритм $a : X \to Y$, способный построить достаточно точный ответ для любого допустимого x_i из X

Множество объектов

X – множество объектов и их описаний (признаков)

Пусть решаем задачу **кредитного скорринга** – по описаниям клиентов банка хотим решить, будем ли выдавать кредит. Возможные признаки:

- Пол
- Возраст
- Доход
- Кредитная история



За хорошие качества клиенту начисляются очки, за плохие – штраф.

Чем больше итоговый скор, тем благонадежнее клиент.

Множество ответов

Y – множество допустимых ответов. Для задачи кредитного скоринга это будет $0, 1$. Положим 0, если человек не вернул кредит и 1 иначе. Это как раз то, что нам нужно предсказать для объектов, ответы на которых мы заранее не знаем.

Прецеденты

Прецедент (от лат. praecedens «предшествующий») — случай или событие, имевшее место в прошлом и служащее примером или основанием для последующих действий в настоящем. В нашем случае это пары "объект-ответ" (x_i, y_i).

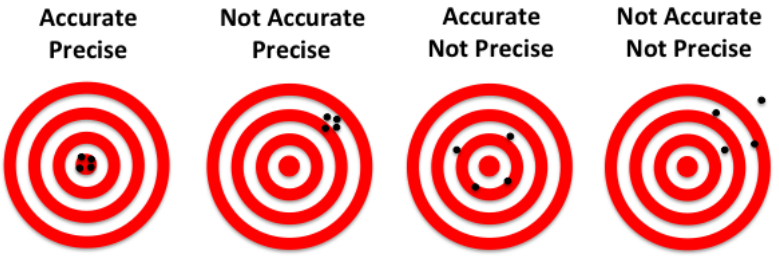
- Отображение $y* : X \to Y$ на прецедентах называется **целевой зависимостью**.
- Хотим построить функцию (алгоритм) $a : X \to Y$, которая приближала бы неизвестную целевую зависимость как на элементах выборки, так и на всём множестве X.
- Говорят также, что алгоритм $a(x)$ должен обладать способностью к обобщению эмпирических фактов, или выводить общее знание (закономерность, зависимость) из частных фактов (наблюдений, прецедентов).

Прецеденты

- Важно, чтобы алгоритм $a(x)$ приближал целевую зависимость хорошо не только на обучающей выборке X_{train}, но и на любых объектах из X. Явление, когда алгоритм хорошо приближает зависимость на обучающей выборке и только на ней, называется **переобучением**

Тестовая выборка

Как можно понять, что модель работает хорошо?
Мы выбираем из множества прецедентов некоторую часть (обычно 20-30 процентов), убираем на них ответы и пробуем построить предсказания. Затем сравниваем с реальными ответами и смотрим, насколько точны наши предсказания.
Существуют различные метрики для оценивания качества алгоритма. Самая простая из них – *accuracy*, которая показывает процент "попаданий".



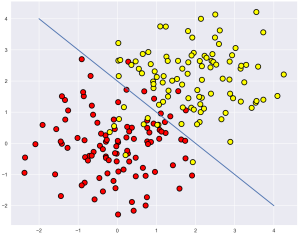
Итог

- Множество объектов (ситуаций) X
- Множество возможных ответов (откликов, реакций) Y
- Между X и Y существует зависимость $a : X \to Y$, известная на конечной выборке **прецедентов** (x_i, y_i) – парах "объект-ответ"
- Множество прецедентов называется обучающей выборкой X_{train}. На основе имеющихся прецедентов необходимо построить алгоритм $a : X \to Y$, способный построить достаточно точный ответ для любого допустимого x_i из X
- Явление, когда алгоритм хорошо приближает зависимость на обучающей выборке и только на ней, называется **переобучением**
- Проверять качество алгоритма можно при помощи X_{test}.
- *Accuracy* (точность) – метрика качества, показывающая процент попаданий.

Классификация

Классификация – задача разделения объектов на классы.

- Делим клиентов банка на тех, кто возвращает кредиты и тех, кто не возвращает
- Определяем, к какой кухне относится блюдо по его ингредиентам



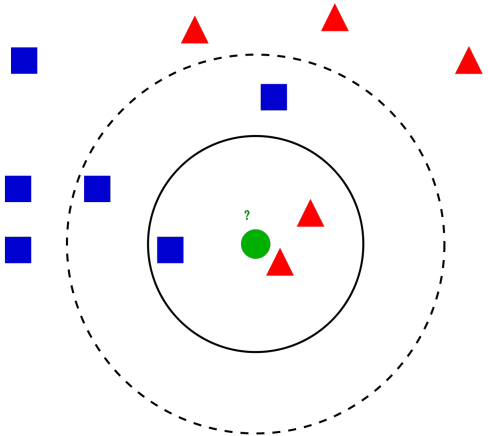
Алгоритм, осуществляющий классификацию, называется классификатор.

Метрический классификатор

Метрический классификатор (similarity-based classifier) — алгоритм классификации, основанный на вычислении оценок сходства между объектами.

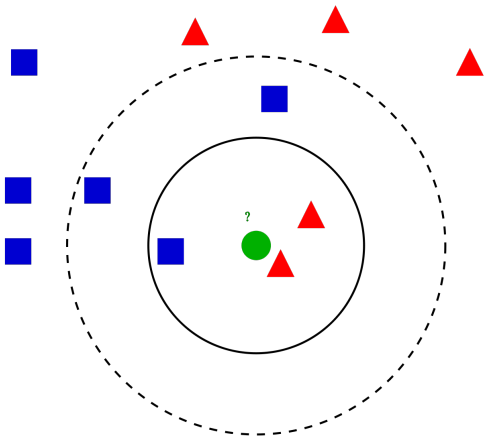
Метод ближайших соседей

Пусть известно, что похожие объекты расположены рядом на плоскости. Когда будем относить к одному классу объекты, расстояние между которыми минимально.



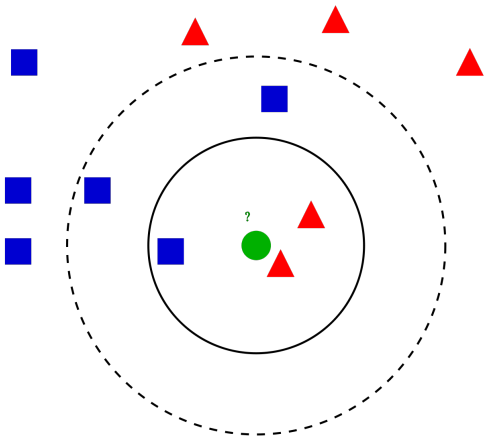
Метод ближайших соседей

Пусть мы решаем задачу классификации для двух классов – на красные треугольники и синие квадраты. Хоти предсказать класс зеленого круга. Тогда выберем тот же класс, что и у объекта с известной меткой, расположенного на минимальном расстоянии от зеленого круга. Этот алгоритм называется **методом ближайшего соседа**



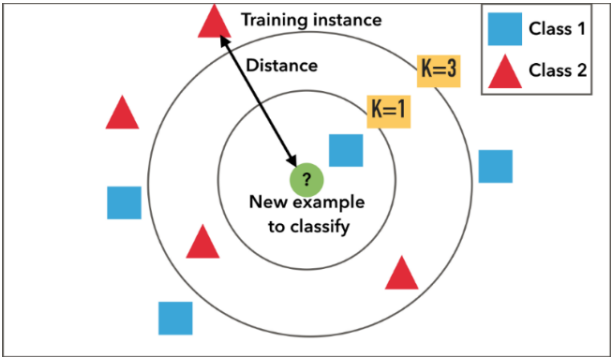
Метод ближайших соседей

Однако что, если ближайший сосед это выброс или мы находимся на границе классов? Тогда стоит рассматривать не только одного ближайшего соседа, но нескольких (k) и отвечать наиболее часто встречающимся классом. Этот алгоритм называется **методом k ближайших соседей** или k-nearest neighbours (knn). k – настраиваемый алгоритм модели



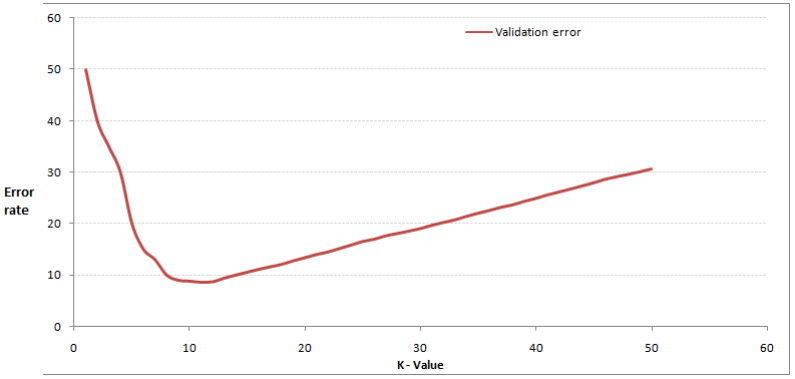
Метод ближайших соседей

Что, если в k-окрестности лежит одинаковое количество классов или больше объектов синего класса, хотя красные лежат явно ближе? Ответ – взвешенный knn! Добавляем – коэффициенты, позволяющие учитывать **важности**. Тогда чем меньше расстояние до объекта, тем больше важность признака. k – настраиваемый параметр модели



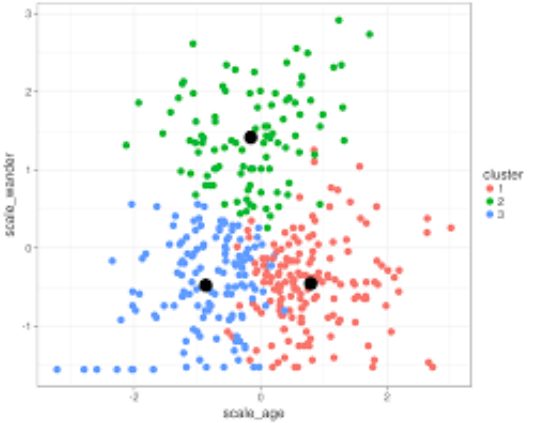
Настройка k

Число соседей можно настраивать путем подстановки в алгоритм и поиска значения, при котором функция ошибок достигает минимума



K-means

Метод **k-средних** – это метод кластерного анализа, целью которого является разделение m наблюдений (из пространства) на k кластеров, при этом каждое наблюдение относится к тому кластеру, к центру (центроиду) которого оно ближе всего.



Линейная регрессия

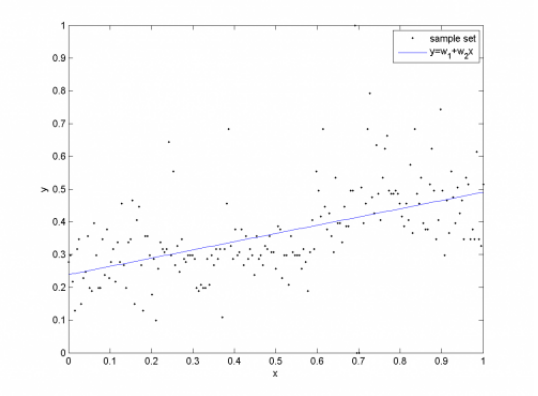
Регрессионная модель зависимости одной (объясняемой, зависимой) переменной y от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) x с линейной функцией зависимости $y(x)$

Одномерная линейная регрессия

(x, y) – пары точек (прецедентов)

Задача: построить предсказания по x для неизвестных y в предположении, что y(x) – линейная функция

$y = Ax + B$



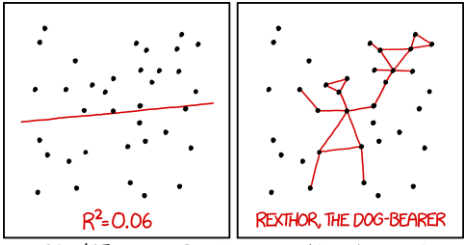
Линейная регрессия

Как выглядит формула для линейной регрессии?

$y = \sum_{i=1}^{n} \omega_i x_i + \omega_0$

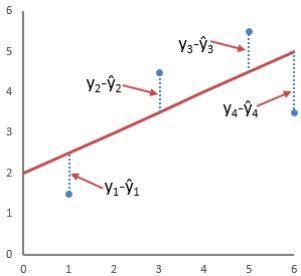
Здесь:

- ω_i – важность i-го признака
- x_i – сам признак



Метод обучения

Для того, чтобы построить решающую функцию $a(x) : X$, нам необходимо подобрать оптимальные параметры. И в нашей задаче это веса. Пусть рассматриваем одномерную регрессию, тогда наша модель имеет вид:
$y = ax + b$
. Настраивать мы можем параметры a, x.
алее, как мы уже делали это раньше, найдем параметры, которые приводят к минимуму ошибки. Как можно эту ошибку задать?
Метод наименьших квадратов!

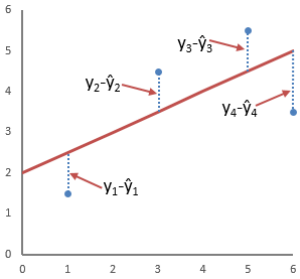


Метод наименьших квадратов (MSE)

- $L_i(a(x_i, \omega_i), y*_i) = (a(x_i, \omega_i) - y*_i)^2$ – ошибка на i-м объекте
- $Q(\omega) = \sum_{i=1}^{n} L_i \to min$ – сумма ошибок по всем объектам (функционал качества).
  Минимизируя функционал качестве путем настройки весов мы добиваемся минимизации квадратичной ошибки.

Существуют и другие функции ошибок. Например – MAE:

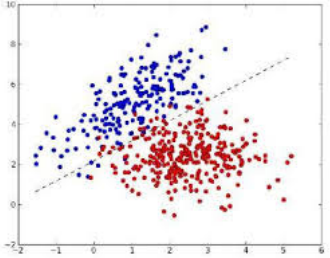
- $L_i(a(x_i, \omega_i), y*_i) = |a(x_i, \omega_i) - y*_i|$ – ошибка на i-м объекте
- $Q(\omega) = \sum_{i=1}^{n} L_i \to min$ – сумма ошибок по всем объектам (функционал качества).



Линейный классификатор

Пусть теперь мы решаем задачу не регрессии, а классификации. Пусть для простоты задача классификации – бинарна и мы находимся на плоскости. Тогда хотим провести прямую (разделяющую гиперплоскость) так, чтобы сверху от прямой лежало как можно больше объектов одного класса, а ниже прямой – другого класса. Самый простой вариант – использовать функцию $sgn(x)$, которая возвращает 1, если $x > 0$ и -1 иначе:

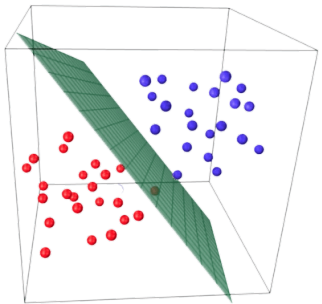
$$y = sgn\left(\sum_{i=1}^{n} \omega_i x_i + \omega_0\right)$$



Обучение линейного классификатора

Самая простая метрика – *accuracy*, которая показывет, в каком проценте случаев мы построили верное предсказание

- $L_i(a(x_i, \omega_i), y*_i) = I[a(x_i, \omega_i) = y*_i]$ – ошибка на i-м объекте
- $Q(\omega) = \sum_{i=1}^{n} L_i \to min$ – сумма ошибок по всем объектам (функционал качества).



Итоги занятия

- Бывают задачи, которые можно решать unsupervised (без обучающей выборки)
- Самые простые классификаторы – метрические, основанные на близости объектов
- Если нужно предсказать вещественное число, то вы решаете задачу регрессии
- Если нужно предсказать метку класса – решаете задачу классификации
- Линейные модели – очень важный и обширный класс методов, который впоследствии используется в нейронных сетях

Успехов!:)