Машинное обучение • Основные алгоритмы

Сегодня

Эльбрус буткемп

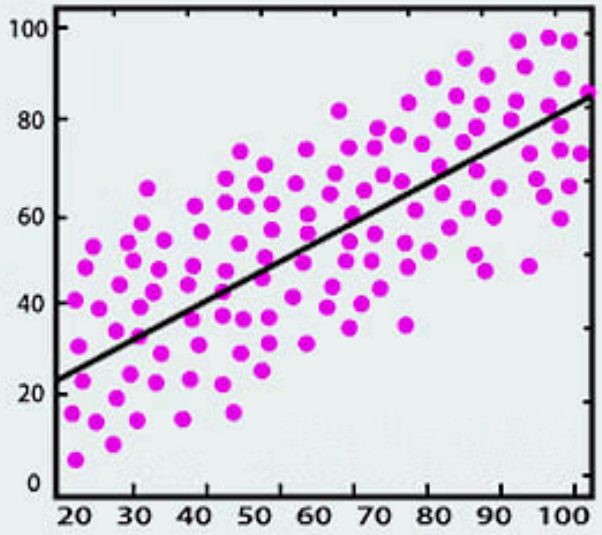
- Основные задачи в машинном обучении
- разбезрем основные алгоритмы
- ответим на вопрос "какой признак самый важный?"
- визуализируем построенное алгоритмы

Типы задач


Классическое Обучение
Данные заранее категоризированы или численные
Данные никак не размечены
С учителем
Без учителя
Предсказать категорию
Предсказать значение
Разделить по схожести
Найти зависимости
Выявить последовательности
Классификация
«Разложи носки по цвету»
Регрессия
«Разложи галстуки по длине»
Кластеризация
«Разложи похожие вещи по кучкам»
Ассоциация
«Найди какие вещи я часто ношу вместе»
Уменьшение Размерности (обобщение)
«Собери из вещей лучшие наряды»

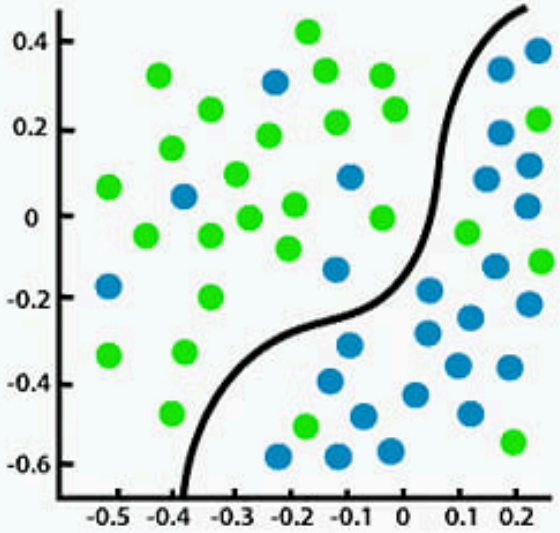
Обучение с Учителем. Регрессия и классификация

Эльбрус буткемп



Regression

versus



Classification

Примеры регрессии и классификации

Эльбрус буткемп

- Примеры задач **регрессии**:

1. Прогнозирование продаж на следующий месяц.
2. Оценка стоимости недвижимости.
3. Прогнозирование доходов компании.

- Примеры задач **классификации**:

1. Определение, будет ли клиент покупать продукт.
2. Классификация писем как спам или не спам.
3. Определение кредитоспособности клиента.

Линейная регрессия

Эльбрус буткемп

Задача: оценить стоимость квартиры по одному признаку (например, площади).

Дана таблица с обучающей выборкой:

Площадь, м2	Цена, млн
50	12
33	8
...	...
76	120

Линейная регрессия

Эльбрус буткемп

Решаем уравнение:

$$y = x_1 w_1 + w_0$$

где x_1 - значение площади, w - "вес" признака, w_0 - свободный параметр.

Если помимо площади есть еще признаки, то уравнение просто увеличивается:

$$y = x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0$$

или короче:

$$y = \sum_{i=1}^{n} x_i w_i + w_0$$

Линейная регрессия: решение

Эльбрус буткемп

- методом градиентного спуска (так обычно и происходит)
  - будем минимизировать *среднеквадратическую ошибку*:

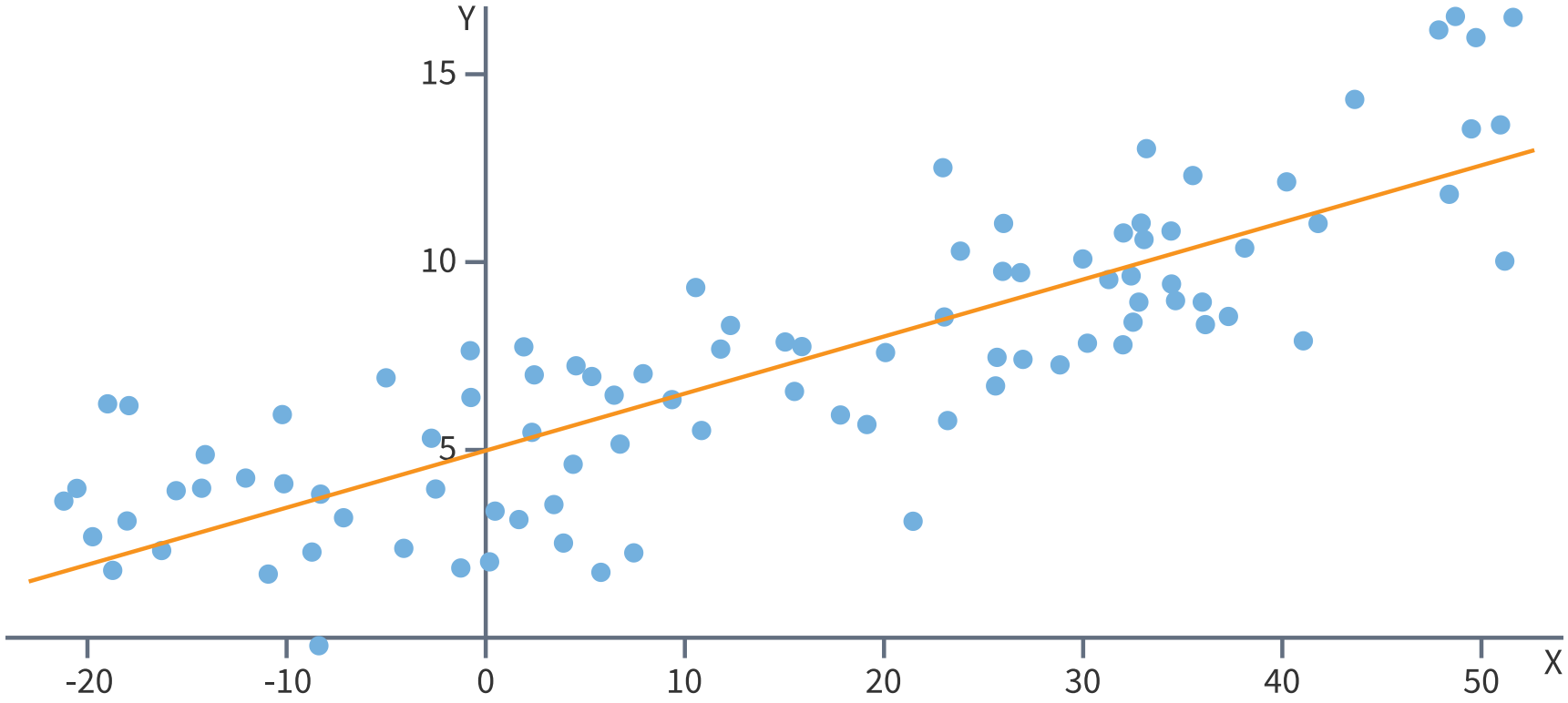
$$L = \frac{1}{K}\sum_{i=1}^{K}(y_i - \hat{y}_i)^2 \to \min$$

$$L = \frac{1}{K}\sum_{i=1}^{K}(y_i - (x_1 \cdot w_1 + x_2 \cdot w_2 + \ldots + x_n \cdot w_n))^2 \to \min$$

y - настоящее значение, $\hat{y}$ - предсказанное моделью значение, K - число объектов в обучающей выборке, $w_1, w_2, \ldots, w_n$ - веса признаков, это и есть наши дифференцируемые парамаметры, частные производные по ним будут составлять наш градиент.

Линейная регрессия: Визуализация

Эльбрус буткемп



Логистическая регрессия

Эльбрус буткемп

- Решает задачу классификации, не смотря на название!

Логистическая регрессия

Эльбрус буткемп

Модель остается линейной:

$$z = \sum_{i=1}^{n} x_i w_i + w_0$$

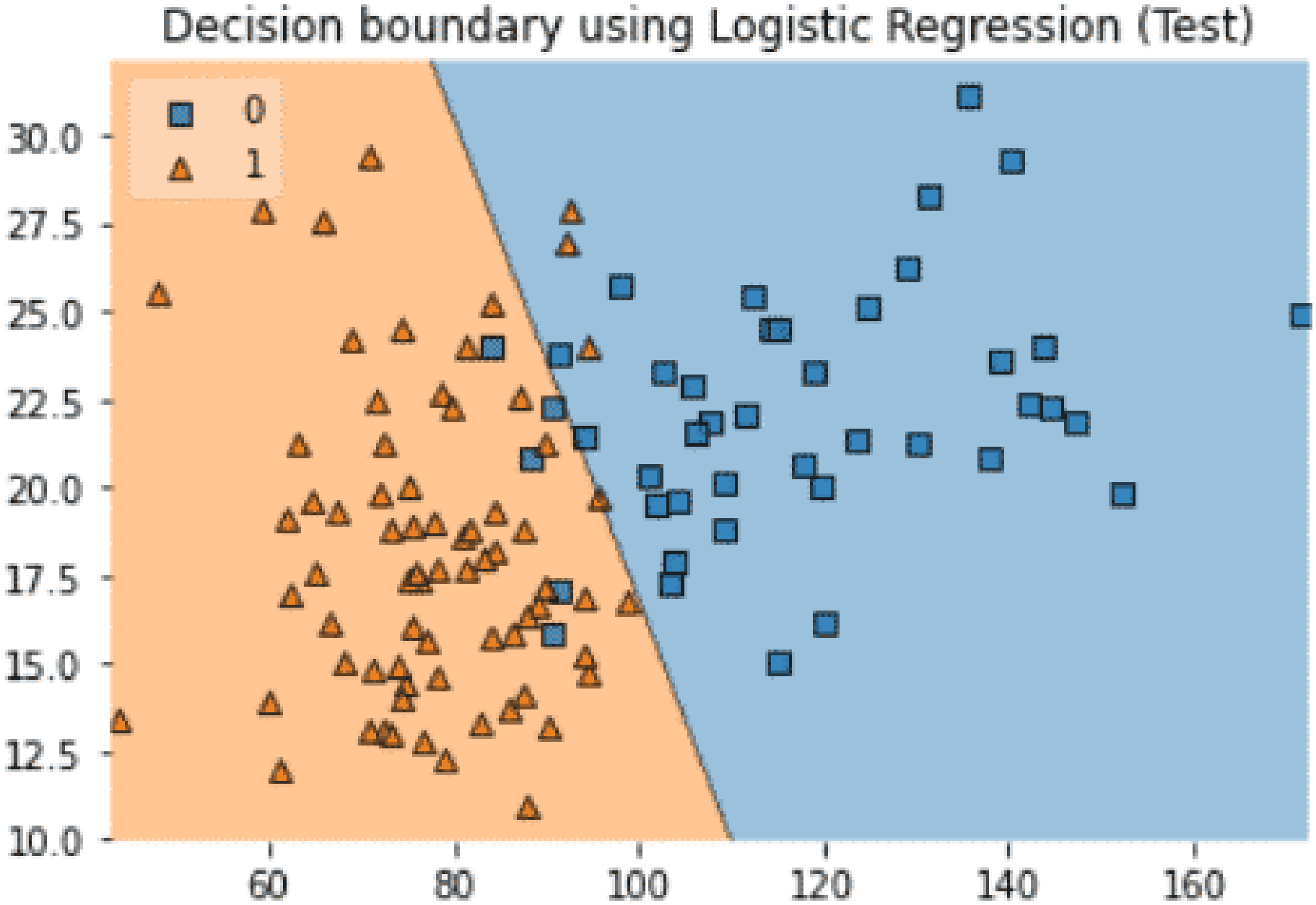
но полученный z подставляем в *логистическую* функцию:

$$y = \frac{1}{1 + e^{-z}}$$

выход y будет лежать в диапазоне от 0 до 1: $y \in [0, 1]$

это значение мы будем интерпретировать как *вероятность* того, что объект относится к классу **1**

Логистическая регрессия: визуализация



Регуляризация линейных моделей

Эльбрус буткемп

- Часто модель может переобучиться на какие-либо выбросы в данных.
- Чтобы этого избежать, можно использовать регуляризацию: регуляризация помогает уменьшить сложность модели и предотвратить переобучение, добавляя штраф за величину коэффициентов в функцию ошибки.
- Три самых распространенных метода регуляризации моделей:
  i. $L1$-регуляризация / LASSO
  ii. $L2$-регуляризация / Ridge
  iii. ElasticNET

α - коэффициент регуляризации

Регуляризация линейных моделей

Эльбрус буткемп

1. $L1$-регуляризация / LASSO: добавляем в функцию ошибки сумму модулей весов:

$$L(y, \hat{y}) = \frac{1}{K}\sum_{i=1}^{K}(y - \hat{y})^2 + \alpha\sum_{j=1}^{n}|w_j|$$

2. $L2$-регуляризация / Ridge: добавляем в функцию ошибки сумму квадратов весов:

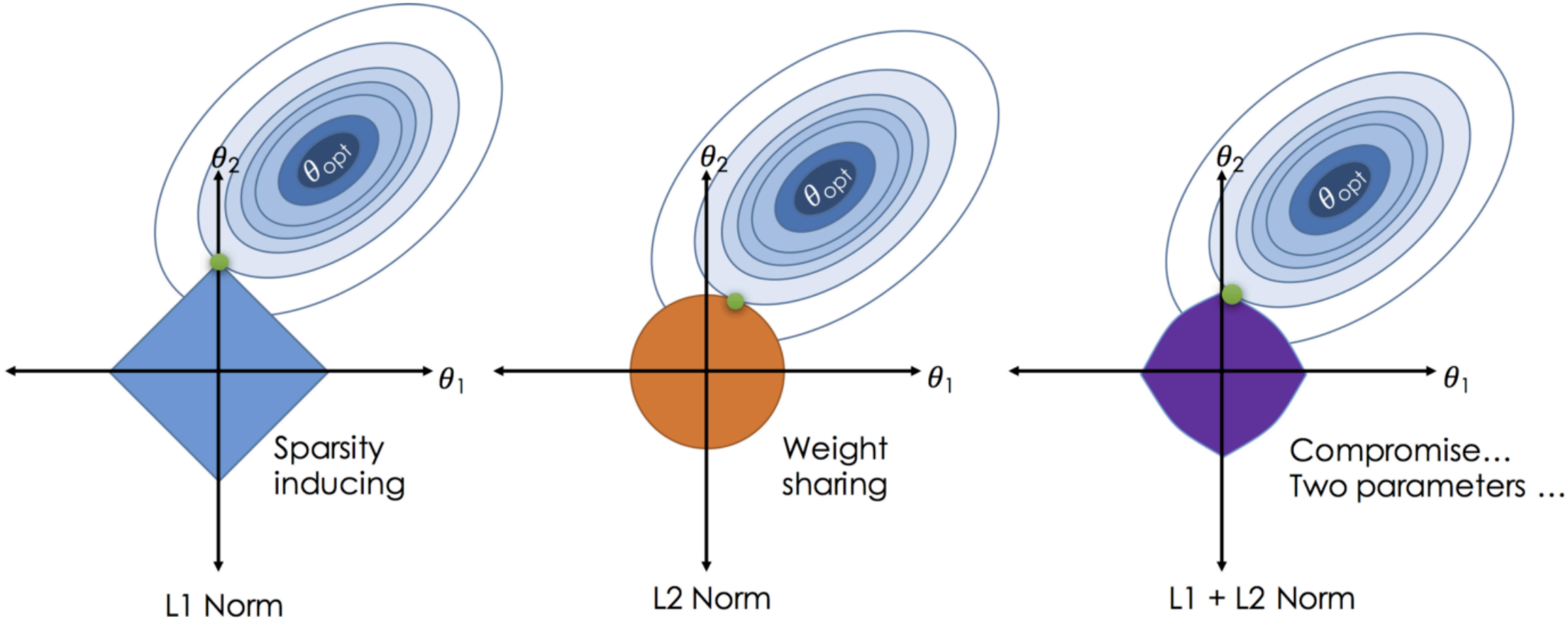
$$L(y, \hat{y}) = \frac{1}{K}\sum_{i=1}^{K}(y - \hat{y})^2 + \alpha\sum_{j=1}^{n}w_j^2$$

3. ElasticNET: добавляем в функцию ошибки сумму квадратов и сумму модулей весов:

$$L(y, \hat{y}) = \frac{1}{K}\sum_{i=1}^{K}(y - \hat{y})^2 + \alpha\sum_{j=1}^{n}w_j^2 + \beta\sum_{j=1}^{n}|w_j|$$

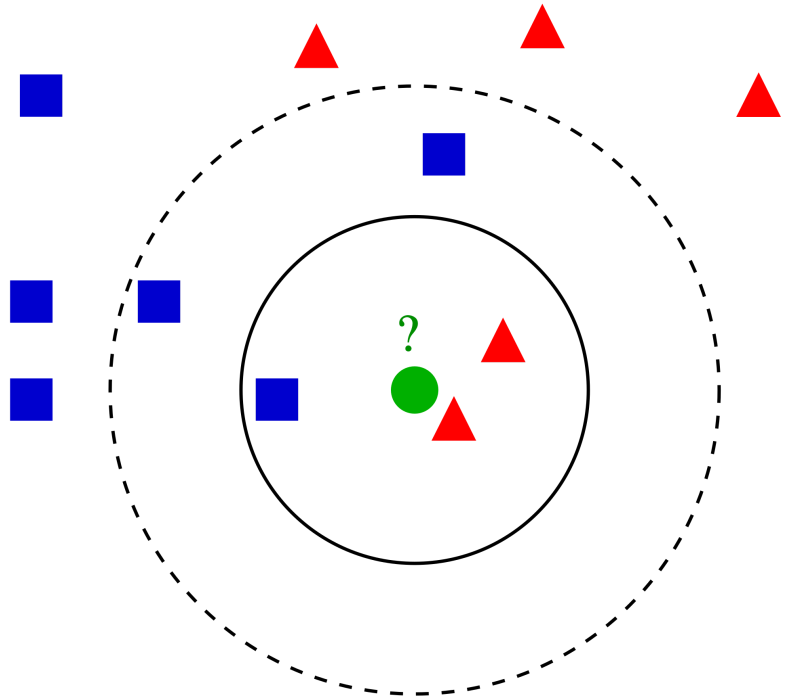
Регуляризация линейных моделей

Эльбрус буткемп



Метод ближайших соседей / KNN

Эльбрус буткемп



Метод ближайших соседей / KNN

Эльбрус буткемп

- классификация:
  - берем k соседей и смотрим, какой класс встречается чаще
- регрессия:
  - берем и вычисляем среднее (можно средневзвешенное) значение для нового объекта

Расстояние

Эльбрус буткемп

- Что значит ближайшие?

Близкие по метрике Минковского:

$$\rho(x, y) = \left(\sum_{i=0}^{d} |x_i - y_i|^p \right)^{1/p}$$

- при $p = 2$ это евклидово расстояние
- при $p = 1$ Манхэттенская метрика
- при $p = \infty$ - метрика Чебышева (наибольшее из всех расстояний)

KNN: гиперпараметры

Эльбрус буткемп

- число соседей / радиус
- метрика
- способ вычисления весов объектов

Decision Tree

Эльбрус буткемп

- Классификация: игра состоится?

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

Какой признак самый информативный?

Эльбрус буткемп

Тот, при использовании которого для классификации, мы получаем наиболее "чистые" подмножества объектов выборки:

- Признак `Windy` имеет два значения: `Weak` и `Strong`
  - Для `Windy=Weak`: `Play=Yes` – 3 объекта, `Play=No` – 1 объект
  - Для `Windy=Strong`: `Play=Yes` – 2 объекта, `Play=No` – 4 объекта
- Признак `Humidity` имеет два значения: `High` и `Normal`:
  - Для `Humidity=Normal`: `Play=Yes` – 2 объекта, `Play=No` – 1 объект
  - Для `Humidity=High`: `Play=Yes` – 3 объекта, `Play=No` – 4 объект

Оценки гомогенности для разбиений

Как измерить гомогенность выборки в полученных разбиениях?

- Коэффициент Джини / Gini impurity:

$$G = 1 - \Sigma_i p_i^2$$

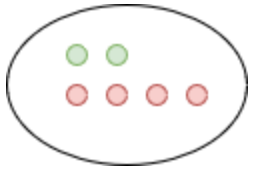
- Энтропия разбиения:

$$H = -\Sigma p_i \log_2 p_i$$

p_i – частота объектов класса i в разбиении

Исходное множество

Эльбрус буткемп

Разбиение	Entropy
Исходное 	$-(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$
`Windy=Weak` 	$-(0.25 \log_2 0.25 + 0.75 \log_2 0.75) = 0.81$
`Windy=Strong` 	$-(0.33 \log_2 0.33 + 0.66 \log_2 0.66) = 0.92$

Information gain

Эльбрус буткемп

$$IG = S_0 - \sum_{i=1}^{q} \frac{N_i}{N} S_i,$$

- q – число листьев (обычно 2),
- N_i – число объектов, попавших в i-ое разбиение,
- N – общее число в родительской веришне объектов,
- S_0 – *impurity metric* (*gini* или *entropy*) для исходного разбиения,
- S_i – *impurity metric* для i-го разбиения.

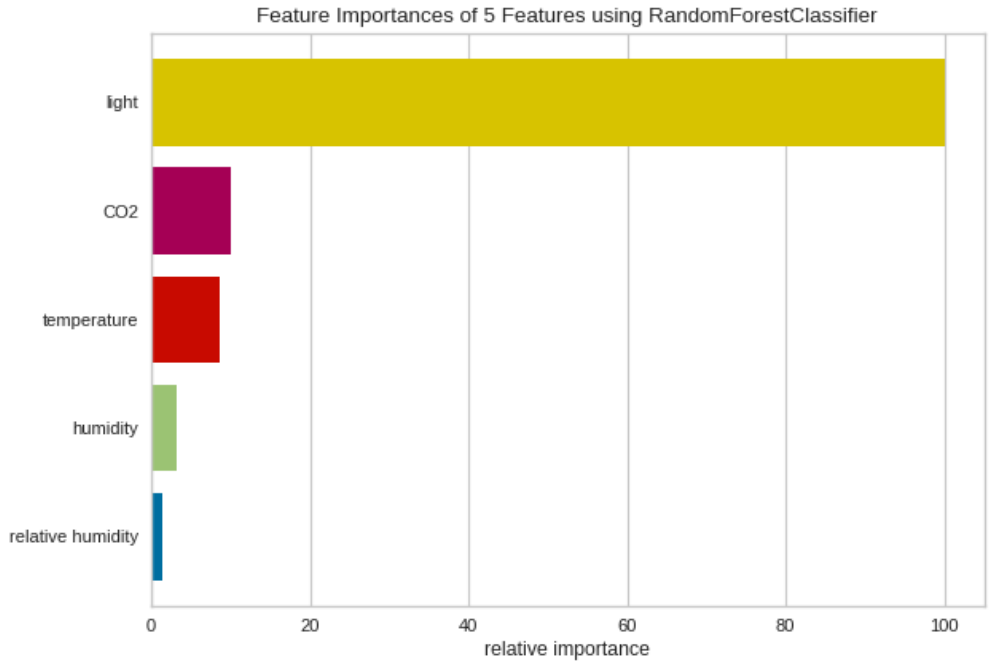
Для разбиения, построоенного по признаку `Windy`:

$$IG = 1 - 0.4 \times 0.81 - 0.6 \times 0.92 = 0.12$$

Важность признаков • Feature importances

Эльбрус буткемп

Суммарный (а в `sklearn` и нормированный) показатель уменьшения гетерогенности выборки используется для оценки важности признаков: если признак выбирался часто и сильно уменьшал энтропию или коэффициент Джини, то он является **информативным**.



Параметры решающих деревьев

Эльбрус буткемп

- Критерий ветвления: `criterion: gini, entropy`
- Максимальная глубина: `max_depth`
- Минимальное число объектов в листе: `min_samples_leaf`
- Минимальное значение уменьшения гетерогенности для осуществления деления: `min_impurity_decrease`

`sklearn.tree`: https://scikit-learn.org/stable/modules/tree.html

Модели машинного обучения

Эльбрус буткемп

Модель	Регрессия	Классификация
Линейные модели	`LinearRegression()`	`LogisticRegression()`
Метод ближайших соседей	`KNeighborsRegressor()`	`KNeighborsClassifier()`
Деревья решений	`DecisionTreeRegressor()`	`DecisionTreeClassifier()`

Итоги

Эльбрус буткемп

Линейные модели

- Простые и интерпретируемые
- Склонны к недообучению на сложных данных
- Примеры: `LinearRegression()`, `LogisticRegression()`

Метод ближайших соседей

- Легко объясним и интуитивно понятен
- Может быть медленным на больших данных
- Примеры: `KNeighborsRegressor()`, `KNeighborsClassifier()`

Итоги

Эльбрус буткемп

Деревья решений

- Мощный алгоритм, склонен к переобучению
- Для уменьшения переобучения можно использовать методы регуляризации
- Может выступать в качестве составного блока для более сложных концепций
- У обученного дерева есть атрибут: `model.feature_importances_`