

1 Лекция 1.

1.1 Введение.

Где применяется машинное обучение:

- Цены на нефть (есть данные нужно предсказать дальнейшее развитие);
- Кредитный скоринг (есть человек, есть кредит — надо понять давать ему кредит или нет. Предсказываем вероятность того, насколько он хороший заёмщик);
- Предсказание температуры (диапазон значений);
- Таргетированная реклама (на основе предпочтений подбираем наиболее подходящие товары);
- Информационный поиск;
- Персонализация (наиболее интересные новости VK — градиентный бустинг деревьев);
- Чат-боты (NN);
- Призма;
- Генерация изображений (GAN); <https://deepmind.com/blog>
- Боты в играх (reinforcement learning);

Замечание: Анонсирован градиентный бустинг на 2 занятия.

Замечание: Мода на сеточки :D надо доказать людям, что заниматься не нейронками тоже прикольно.

1.2 По сути

Пусть у нас есть тренировочные данные кредитного скоринга:

- Пол;
- Возраст;
- Возвращают кредит или нет;

Хотим предсказать вернёт он нам кредит или нет.

Определение. Дано X — обучающая выборка (множество объектов), где $x \in X = \{x\}$ — прецедент, модель ML, после чего **модель** обучается — получается алгоритм машинного обучения. Y — множество допустимых ответов (target).

Данные необходимо проверять — модель хочет понять на сколько хорошо модель «понимает» данные, которые она никогда не видела. Проверяем обобщающую способность данных.

Определение. $y^* : X \rightarrow Y$ — **целевая функция**. Имеем значения только на конечном наборе данных $x_i \in X : y^*(x_i) = y_i$. Сопоставляем входным данным соответствующие ответы.

Определение. a — **решающая функция (алгоритм)** — обобщение функции y^* на всё множество объектов.

$$X = X_{\text{train}} \cup X_{\text{test}}$$

Делим выборку на обучающую + тестовую.

Замечание **Kaggle**. Титаник.

Определение. **Признак** (feature) объекта x — это результаты измерения характеристики объекта. Для $x_j = \{x_j^1, \dots, x_j^n\}$ — признаковое описание объекта (вектор).

Можно генерировать новые признаки различными алгоритмами (feature engineering).

Очень круто выяснить какие features наиболее важны для модели.

Формализуем: по выборке X_{train} построить решающую функцию которая хорошо приближает y^* как на X_{train} , так и на всём множестве X .

Чтобы решать задачу необходимо ввести функционал качества. Для этого необходимо ввести метрику и функцию ошибки. Решаем задачу оптимизации: ищем набор параметров в пространстве параметров такой, чтобы Максимизировать функционал качества, минимизировать функцию ошибок (*Loss*).

1.3 Метрики качества

$$MAE = \frac{1}{N} \sum_1^N |y_i - y_i^*|$$

$$MSE = \frac{1}{N} \sum_1^N (y_i - y_i^*)^2$$

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

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

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

$$F = 2 \cdot \frac{PRE \cdot REC}{PRE + REC}$$

- TP — 1 и алгоритм говорит, что 1;
- TN — 0 и алгоритм говорит, что 0;
- FP — 0 алгоритм говорит, что 1;
- FN — 1 алгоритм говорит, что 0;

Для оценки качества работы алгоритма на каждом из классов по отдельности введем метрики precision (точность) и recall (полнота).

Precision можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными, а recall показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм.

Почему нужны эти метрики: Если данных одного класса сильно больше, чем другого, то алгоритму проще обучиться всегда выводить первый класс (вероятность ошибки ниже при accuracy). А precision и recall смещают данные.

2 Машинное обучение

- Классическое обучение (с учителем — без);
- Ансамблевые методы (много моделей);

Определение. Классификация. есть множество объектов и множество классов (конечное). Обучаем модель, разделяющую данные на группы.

Определение. Регрессия. отличается тем, что допустимым ответом является число или числовой вектор (пространство непрерывно, бесконечное множество ответов).

Без учителя:

- Кластеризация;
- Поиск правил;
- Уменьшение размерности;

Определение. **Переобучение** — модель находит зависимости там, где их нет (например у всех плохих заёмщиков были красные ботинки); Причины:

- Слишком много степеней свободы (слишком большая модель);
- Мало данных (генерал зависимости не находим, а находим локальные зависимости);

Определение. Валидационная выборка — знаем на ней ответы (на тестовой не знаем);

Определение. Кросс-валидация — ошибка на валидационных данных не отображает реальной картины мира. Делим N раз на валидационную и тестовую, N раз тренируем алгоритм, считаем среднюю ошибку.

Методы борьбы

Регуляризация (добавляем функцию от весов в loss, сдерживаем их рост)