Содержание

1 Леция 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, на сколько 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-ом занятии.
Замечание: Мода на сеточки :D надо доказать людям, что заниматься не нейронками тоже прикольно.

1.2 По сути

Пусь у нас есть тренировачные 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. $y^* : X \to Y$ — **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 результа измерения характеристики объекта. Для $x_j = \{x_j^1, \ldots, x_j^n\}$ — 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 моделек);

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, бескоечное 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 (general 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)

3 Лекция 2. Модели, ансамбли моделей

3.1 Зоопарк моделей

Занимаемся моделями обучения с учителем.
Есть 2 варианта постановки задачи:

- Классификация;
- Регрессия;

3.1.1 Наивный байесовский классификатор

Самый простой способ, как построить себе модель.
Замечание. *Только классификация.*
Для класса определяем вероятность. Есть условная вероятость, как часто встречается фича с фиксированной величиной для каждого класса.
По формуле Байеса найдём вероятность ответа при значении фичей.

$$p(A|B) = \frac{p(A) \cdot p(B|A)}{p(B)}$$

Определение. *Априорная вероятность.* – это вероятность, присвоенная событию при отсутствии знания, поддерживающего его наступление $p(A)$;
Определение. *Апостериорная вероятност.* – это условная вероятность события $p(A|B)$.

- Spam detection;
- Сегментация новостных статей по теме;
- Определение эмоционального окраса текста;

3.1.2 Линейные модели

Определение. *Линейная модель.* выражается линейной функцией.

$$Y_i = \beta_0 + \beta_1 f(X_{i1}) + \cdot + \beta_p f(X_{ip}) + \varepsilon_i$$

Где ε_i – шум (гауссово распределение), $i = 1, \ldots, N$.
Замечание. Функции $f_1, \ldots, f_p$ не обязательно должны быть линейными.

3.1.3 Линейная регрессия

$$y = X \cdot w^T + \varepsilon$$

В данном случае решение будет решением СЛАУ:

$$w^T = \left(W^T W\right)^{-1} W^T y$$

Проблема 1. решение будет «осмысленным» только тогда, когда в данных реально есть линейная зависимость. В реальных данных почти не встречается.
Проблема 2. Линейная зависимость признаков, $\det W^T W$ должен быть отличен от 0.
Решение проблемы: **регуляризация**:

$$w^T = \left(W^T W + \lambda \cdot \mathrm{Id}\right)^{-1} W^T y$$

3.1.4 Градиентный спуск

Веса модели ищутся с помощью градиентного спуска

1. Инициализация W случайными значениями;
2. Итеративно для всех сэмплов $x \in X$:
   (a) Вычислить функцию потерь на x;
   (b) Вычислить производные функции потерь $grad_w$ по каждому из $w \in W$;

(c) Обновляем веса: $w = w - \mathrm{lr} \cdot grad_w$

3. Полученное $W = \{w\}$ — искомое.

Для линейной регрессии функция потерь:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - (Wx_i + b))^2$$