Решающие деревья

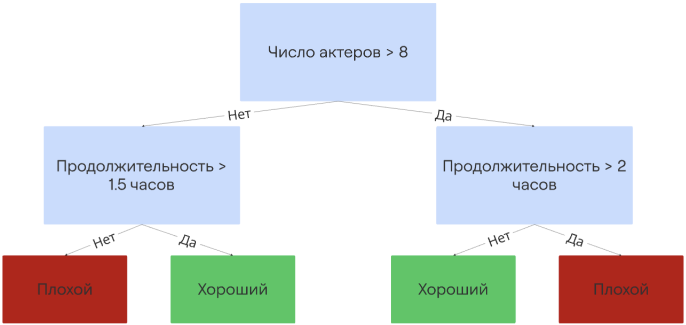
Арам Аветисян

3 апреля 2025

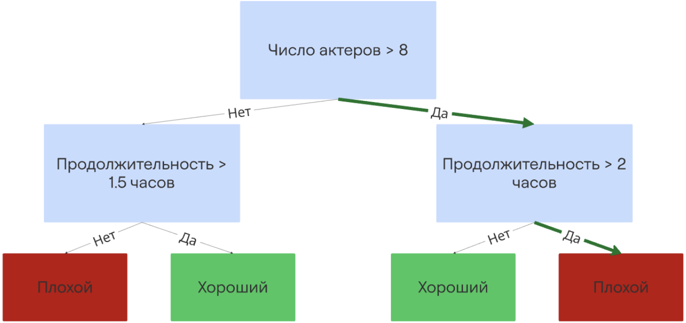
Решающие деревья

Деревья решений делают предсказания, рекурсивно разделяя различные признаки в соответствии с древовидной структурой.

Пример Классификация фильма как хорошего или плохого на основе продолжительности и количества актеров



Тестовый пример Крестный отец (2 ч 55 мин, актеров много)



Определение решающего дерева

Данные:

- X: Пространство признаков (входные данные).
- Y: Пространство меток (выходные данные).

Бинарное дерево:

- Внутренние вершины: предикат $Q_i : X \to \{0, 1\}$
- Листовые вершины: прогноз $\hat{y}_i \in Y$

Определение решающего дерева

Данные:

- X: Пространство признаков (входные данные).
- Y: Пространство меток (выходные данные).

Бинарное дерево:

- Внутренние вершины: предикат $Q_i : X \to \{0, 1\}$
- Листовые вершины: прогноз $\hat{y}_i \in Y$

Процесс предсказания:

- Движение от корня
- Вправо, если $Q_i(x) = 1$, влево, если $Q_i(x) = 0$
- Ответ — прогноз листа $\hat{y}_i$

Почему это сложная задача?

- Пусть есть датасет (X, y), где X — матрица признаков, y — вектор таргетов.
- Цель: минимизировать некоторую функцию потерь $L(f, X, y)$.

Построение оптимального решающего дерева

Почему это сложная задача?

- Пусть есть датасет (X, y), где X — матрица признаков, y — вектор таргетов.
- Цель: минимизировать некоторую функцию потерь $L(f, X, y)$.
- Оптимизация структуры дерева градиентным спуском невозможна (почему?)

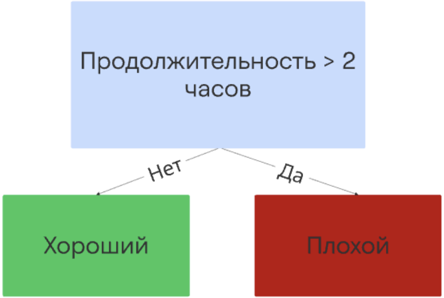
Построение оптимального решающего дерева

Почему это сложная задача?

- Пусть есть датасет (X, y) размера N, где X — матрица M признаков, y — вектор таргетов
- Цель: минимизировать некоторую функцию потерь $L(f, X, y)$.
- Оптимизация структуры дерева градиентным спуском невозможна (почему?). Функция для построенного дерева кусочно-постоянная -> производная равна нулю везде, где задана

Решающий пень

Решающий пень - простое дерево решений с 1 правилом разделения

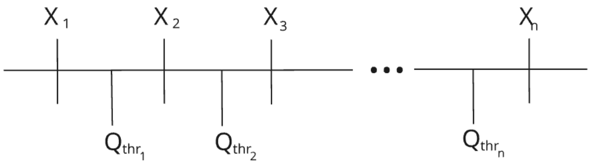


Построим решающий пень

Алгоритм:

- Будем решать задачу минимизацию функции потерь полным перебором

$$(feature_{\text{best}}, threshold_{\text{best}}) = \arg\min_{f,t} L(Q_{f,t}, X, y)$$



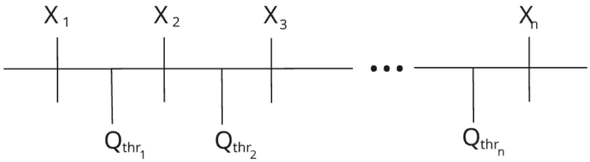
Построим решающий пень

Алгоритм:

- Будем решать задачу минимизацию функции потерь полным перебором

$$(feature_{\text{best}}, threshold_{\text{best}}) = \arg\min_{f,t} L(Q_{f,t}, X, y)$$

- Всего не более $(N-1) * M$ предикатов.
- Для каждого предиката нужно посчитать функцию потерь, пройдясь по всему датасету



Построим решающий пень

Алгоритм:

- Будем решать задачу минимизацию функции потерь полным перебором

$$(feature_{\text{best}}, threshold_{\text{best}}) = \arg\min_{f,t} L(Q_{f,t}, X, y)$$

- Всего не более $(N-1) * M$ предикатов.
- Для каждого предиката нужно посчитать функцию потерь, пройдясь по всему датасету

Сложность алгоритма: $O(N^2 M)$

Обобщение для дерева произвольной глубины

Рекурсивный алгоритм:

- Вызываем функцию для всех возможных разбиений.
- Проблема: так можно построить дерево, идеально запоминающее всю выборку, однако на тестовых данных такой алгоритм вряд ли покажет высокое качество.

Предложение:

- Построить оптимальное с точки зрения качества на обучающей выборке дерево минимальной глубины

Обобщение для дерева произвольной глубины

Рекурсивный алгоритм:

- Вызываем функцию для всех возможных разбиений.
- Проблема: так можно построить дерево, идеально запоминающее всю выборку, однако на тестовых данных такой алгоритм вряд ли покажет высокое качество.

Предложение:

- Построить оптимальное с точки зрения качества на обучающей выборке дерево минимальной глубины
- Проблема: построение идеального дерева — NP-полная задача.

Как будем решать?

Решение: будем искать не оптимальное, а хорошее решение
Жадный алгоритм: строим дерево по уровням
Ключевые идеи:

- Разбиваем выборку на каждом уровне
- Используем эвристики для улучшения качества

Жадный алгоритм построения дерева

Пусть X — исходное множество объектов обучающей выборки, а X_m — множество объектов, попавших в текущий лист (в самом начале $X_m = X$).

Основные шаги:

- Создаём вершину v.

Жадный алгоритм построения дерева

Пусть X — исходное множество объектов обучающей выборки, а X_m — множество объектов, попавших в текущий лист (в самом начале $X_m = X$).

Основные шаги:

- Создаём вершину v.
- Если выполнен **критерий остановки** $S(X_m)$, объявляем её листом и определяем выходное значение $A(X_m)$

Критерии остановки и выходные значения в листах

Когда прекращать разбиение?

- Достигнута минимальная возможная ошибка.
- Меньше определённого числа объектов в листе.
- Достигнута заданная глубина дерева.

Как назначать прогноз в листе?

- Для задачи классификации — самый частый класс или распределение вероятностей.
- Для регрессии — среднее, медиана или другая статистика.
- Листы могут содержать небольшие модели, например, линейную регрессию.

Жадный алгоритм построения дерева

Основные шаги:

- Создаём вершину v.
- Если выполнен критерий остановки $S(X_m)$, объявляем её листом и назначаем ответ $A(X_m)$.
- Иначе определяем критерий ветвления: находим предикат $Q_{i,t}$, который даёт наилучшее разбиение множества X_m.

Жадный алгоритм построения дерева

Основные шаги:

- Создаём вершину v.
- Если выполнен критерий остановки $S(X_m)$, объявляем её листом и назначаем ответ $A(X_m)$.
- Иначе определяем критерий ветвления: находим предикат $Q_{i,t}$, который даёт наилучшее разбиение множества X_m.
  - Оцениваем улучшение выбранной метрики качества при разбиении.
  - Выбираем предикат, дающий максимальное улучшение.

Жадный алгоритм построения дерева

Основные шаги:

- Создаём вершину v.
- Если выполнен критерий остановки $S(X_m)$, объявляем её листом и назначаем ответ $A(X_m)$.
- Иначе определяем критерий ветвления: находим предикат $Q_{i,t}$, который даёт наилучшее разбиение множества X_m.
- Для образовавшихся подвыборок рекурсивно повторяем процедуру.

Формализуем критерий ветвления

1. Определим ответы дерева:
   - $\hat{y} \in \mathbb{R}$ — для регрессии и меток класса.
   - $\hat{y} \in \mathbb{R}^K$ — вектор вероятностей для дискретного распределения:

$$\hat{y} = (\hat{y}_1, \ldots, \hat{y}_K), \quad \sum_{i=1}^{K} \hat{y}_i = 1$$

2. Зададим функцию потерь $L(y_i, \hat{y})$, которая определяет качество предсказания.
3. Наша задача — найти оптимальное разделение выборки X_m: $X_m = X_l \cup X_r$.

Формализуем критерий ветвления

1. Определим ответы дерева:
   - $\hat{y} \in \mathbb{R}$ — для регрессии и меток класса.
   - $\hat{y} \in \mathbb{R}^K$ — вектор вероятностей для дискретного распределения:

$$\hat{y} = (\hat{y}_1, \dots, \hat{y}_K), \quad \sum_{i=1}^{K} \hat{y}_i = 1$$

2. Зададим функцию потерь $L(y_i, \hat{y})$, которая определяет качество предсказания.
3. Наша задача найти оптимальное разделение выборки X_m: $X_m = X_l \cup X_r$
4. Попробуем найти константу $\hat{y}$, которое предсказало бы дерево, если бы мы дошли до листовой вершины (ответа)

Формализуем критерий ветвления

1. Определим ответы дерева:
   - $\hat{y} \in \mathbb{R}$ — для регрессии и меток класса.
   - $\hat{y} \in \mathbb{R}^K$ — вектор вероятностей для дискретного распределения:

$$\hat{y} = (\hat{y}_1, \ldots, \hat{y}_K), \quad \sum_{i=1}^{K} \hat{y}_i = 1$$

2. Зададим функцию потерь $L(y_i, \hat{y})$, которая определяет качество предсказания.
3. Наша задача найти оптимальное разделение выборки X_m: $X_m = X_l \cup X_r$
4. Попробуем найти константу $\hat{y}$, которое предсказало бы дерево, если бы мы дошли до листовой вершины (ответа), т.е. минимизировала среднее значение функции потерь:

$$\frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} L(y_i, \hat{y})$$

Формализация критерия ветвления

Попробуем найти константу $\hat{y}$, которое предсказало бы дерево, если бы мы дошли до листовой вершины (ответа), т.е. минимизировала среднее значение функции потерь:

$$\frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} L(y_i, \hat{y})$$

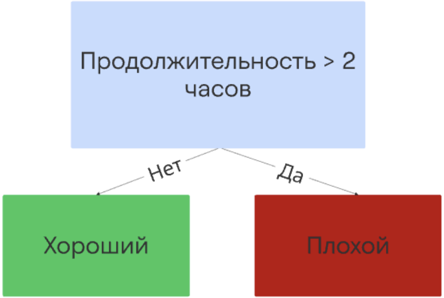
Информативность (impurity) - оптимальное значение этой величины:

$$In(X_m) = \min_{\hat{y} \in Y} \frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} L(y_i, \hat{y})$$

Чем ниже информативность, тем лучше приближение константой

Решающий пень

Решающий пень - простое дерево решений с 1 правилом разделения



Информативность решающего пня

Определим информативность решающего пня.
Пусть:

- X_l — множество объектов, попавших в левую вершину.
- X_r — множество объектов, попавших в правую вершину.
- $\hat{y}_l$ и $\hat{y}_r$ — константы, которые предсказываются в этих вершинах.

Информативность решающего пня

Определим информативность решающего пня.

Пусть:

- X_l — множество объектов, попавших в левую вершину.
- X_r — множество объектов, попавших в правую вершину.
- $\hat{y}_l$ и $\hat{y}_r$ — константы, которые предсказываются в этих вершинах.

Тогда функция потерь для всего пня в целом будет равна:

$$\frac{1}{|X_m|}\left(\sum_{x_i \in X_l} L(y_i, \hat{y}_l) + \sum_{x_i \in X_r} L(y_i, \hat{y}_r)\right)$$

Связь информативностей

Вопрос Как информативность решающего пня связана с информативностью его двух листьев?

Связь информативностей

Преобразуем выражение:

$$\frac{1}{|X_m|}\left(\sum_{x_i \in X_l} L(y_i, \hat{y}_l) + \sum_{x_i \in X_r} L(y_i, \hat{y}_r)\right)$$

Связь информативностей

Преобразуем выражение:

$$\frac{1}{|X_m|}\left(\sum_{x_i \in X_l} L(y_i, \hat{y}_l) + \sum_{x_i \in X_r} L(y_i, \hat{y}_r)\right)$$

$=$

$$\frac{1}{|X_m|}\left(|X_l| \cdot \frac{1}{|X_l|}\sum_{x_i \in X_l} L(y_i, \hat{y}_l) + |X_r| \cdot \frac{1}{|X_r|}\sum_{x_i \in X_r} L(y_i, \hat{y}_r)\right)$$

Связь информативностей

Преобразуем выражение:

$$\frac{1}{|X_m|}\left(\sum_{x_i\in X_l} L(y_i,\hat{y}_l)+\sum_{x_i\in X_r} L(y_i,\hat{y}_r)\right)$$

$=$

$$\frac{1}{|X_m|}\left(|X_l|\cdot\frac{1}{|X_l|}\sum_{x_i\in X_l} L(y_i,\hat{y}_l)+|X_r|\cdot\frac{1}{|X_r|}\sum_{x_i\in X_r} L(y_i,\hat{y}_r)\right)$$

$=$

$$\frac{|X_l|}{|X_m|}In(X_l)+\frac{|X_r|}{|X_m|}In(X_r)$$

информативность решающего пня при оптимальном выборе констант $\hat{y}_l$ и $\hat{y}_r$

Критерий ветвления

Для принятия решения о разделении сравниваем информативность исходного листа и решающего пня.

Разность информативности исходной вершины и решающего пня:

$$In(X_m) - \frac{|X_l|}{|X_m|} In(X_l) - \frac{|X_r|}{|X_m|} In(X_r)$$

Критерий ветвления

Для принятия решения о разделении сравниваем информативность исходного листа и решающего пня.
Разность информативности исходной вершины и решающего пня:

$$In(X_m) - \frac{|X_l|}{|X_m|} In(X_l) - \frac{|X_r|}{|X_m|} In(X_r)$$

Умножим на $|X_m|$ и получим критерий ветвления:

$$\mathrm{IG}(X_m) = |X_m| \cdot In(X_m) - |X_l| \cdot In(X_l) - |X_r| \cdot In(X_r)$$

Полученная величина неотрицательна и тем больше, чем лучше предлагаемый сплит.

Информативность в задаче регрессии: MSE

Рассмотрим задачу регрессию и выберем в качестве критерия минимизацию среднеквадратичной ошибки (MSE):

$$L(y_i, \hat{y}) = (y_i - \hat{y})^2$$

Информативность листа:

$$In(X_m) = \frac{1}{|X_m|} \min_{\hat{y} \in Y} \sum_{(x_i, y_i) \in X_m} (y_i - \hat{y})^2$$

Информативность в задаче регрессии: MSE

Информативность листа:

$$In(X_m) = \frac{1}{|X_m|} \min_{\hat{y} \in Y} \sum_{(x_i, y_i) \in X_m} (y_i - \hat{y})^2$$

Оптимальным предсказание константного классификатора для задачи минимизации MSE - среднее значение:

$$\hat{y} = \frac{1}{|X_m|} \sum y_i$$

=>

$$In(X_m) = \frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} (y_i - \overline{y})^2, \quad \text{где } \overline{y} = \frac{1}{|X_m|} \sum_i y_i$$

Критерий информативности в задаче классификации: misclassification error

Рассмотим задачу классификации с K классами и выберем в качестве критерия индикатор ошибки:

$$L(y_i, \hat{y}) = I[y_i \neq \hat{y}]$$

Информативность для такой функции потерь:

$$In(X_m) = \min_{\hat{y} \in Y} \frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} I[y_i \neq \hat{y}]$$

Критерий информативности в задаче классификации: misclassification error

Информативность для такой функции потерь:

$$In(X_m) = \min_{\hat{y} \in Y} \frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} I[y_i \neq \hat{y}]$$

Пусть p_k — доля объектов класса k в текущей вершине X_m:

$$p_k = \frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} I[y_i = k]$$

Оптимальным предсказанием в листе будет наиболее частотный класс k^*, информативность можно записать следующим образом:

$$In(X_m) = \frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} I[y_i \neq k^*] = 1 - p_{k^*}$$

Алгоритм ID3 (1986)

- Начинаем с исходного набора X в корневом узле
- На каждой итерации:
  - Перебираем все неиспользуемые признаки
  - Вычисляем энтропию $\mathrm{In}(X)$ (см. дальше) и прирост информации
  - Выбираем признак с наибольшим приростом информации
- Разделяем набор X по выбранному атрибуту, создавая подмножества
- Рекурсивно продолжаем для каждого подмножества, игнорируя ранее выбранные признаки
- Рекурсия останавливается, если:
  - Все элементы подмножества принадлежат одному классу
  - Нет признаков для выбора
  - Нет примеров в подмножестве

Предсказание вероятностного распределения классов

Пусть мы предсказываем вероятностное распределение классов $(\hat{y}_1, \dots, \hat{y}_K)$. Будем подходим к этому, максимизируя правдоподобие этого распределения на обучающей выборке.

Пусть в вершине дерева предсказывается фиксированное распределение $\hat{y}$ (не зависящее от x_i), тогда правдоподобие имеет вид:

$$P(y \mid x, \hat{y}) = P(y \mid \hat{y}) = \prod_{(x_i, y_i) \in X_m} P(y_i \mid \hat{y}) = \prod_{(x_i, y_i) \in X_m} \prod_{k=1}^{K} \hat{y}_k^{I[y_i = k]}$$

Предсказание вероятностного распределения классов

Пусть мы предсказываем вероятностное распределение классов $(\hat{y}_1, \dots, \hat{y}_K)$. Будем подходим к этому, максимизируя правдоподобие этого распределения на обучающей выборке.
Пусть в вершине дерева предсказывается фиксированное распределение $\hat{y}$ (не зависящее от x_i), тогда правдоподобие имеет вид:

$$P(y \mid x, \hat{y}) = P(y \mid \hat{y}) = \prod_{(x_i, y_i) \in X_m} P(y_i \mid \hat{y}) = \prod_{(x_i, y_i) \in X_m} \prod_{k=1}^{K} \hat{y}_k^{I[y_i = k]}$$

Откуда информативность (минимизируем отрицательное правдоподобие, берем логарифм):

$$In(X_m) = \min_{\sum_k \hat{y}_k = 1} \left(-\frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} \sum_{k=1}^{K} I[y_i = k] \log \hat{y}_k \right)$$

Вероятностное распределение классов

Вопрос Чему равны оценки вероятностей $\hat{y}_k$, минимизирующие $In(X_m)$?

Вероятностное распределение классов

Вспомним, что $\sum_k \hat{y}_k = 1$ => добавим множитель Лагранжа и будем минимизировать новую функцию:

$$L(\hat{y}, \lambda) = \min_{\hat{y}, \lambda} \left(-\frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} \sum_{k=1}^{K} I[y_i = k] \log \hat{y}_k + \lambda \sum_{k=1}^{K} \hat{y}_k \right)$$

Ищем минимум функции

Вспомним, что $\sum_k \hat{y}_k = 1$ => добавим множитель Лагранжа и будем минимизировать новую функцию

$$L(\hat{y}, \lambda) = \min_{\hat{y},\lambda} \left(-\frac{1}{|X_m|} \sum_{(x_i,y_i)\in X_m} \sum_{k=1}^{K} I[y_i = k] \log \hat{y}_k + \lambda \sum_{k=1}^{K} \hat{y}_k \right)$$

Возьмём частную производную и решим уравнение:

$$\frac{\partial}{\partial \hat{y}_j} L(c, \lambda) = \left(-\frac{1}{|X_m|} \sum_{(x_i,y_i)\in X_m} I[y_i = j] \frac{1}{\hat{y}_j} \right) + \lambda = -\frac{p_j}{\hat{y}_j} + \lambda = 0$$

=>

$$\hat{y}_j = \frac{p_j}{\lambda}$$

Ищем минимум функции

$$L(\hat{y}, \lambda) = \min_{\hat{y}, \lambda} \left(-\frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} \sum_{k=1}^{K} I[y_i = k] \log \hat{y}_k + \lambda \sum_{k=1}^{K} \hat{y}_k \right)$$

Возьмём частную производную и решим уравнение:

$$\frac{\partial}{\partial \hat{y}_j} L(\hat{y}, \lambda) = \left(-\frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} I[y_i = j] \frac{1}{\hat{y}_j} \right) + \lambda = -\frac{p_j}{\hat{y}_j} + \lambda = 0$$

=>

$$\hat{y}_j = \frac{p_j}{\lambda}$$

Суммируя эти равенства, получим:

$$1 = \sum_{k=1}^{K} \hat{y}_k = \frac{1}{\lambda} \sum_{k=1}^{K} p_k = \frac{1}{\lambda}$$

Ищем минимум функции

$$\hat{y}_j = \frac{p_j}{\lambda}$$

$$1 = \sum_{k=1}^{K} \hat{y}_k = \frac{1}{\lambda} \sum_{k=1}^{K} p_k = \frac{1}{\lambda} =>$$

$$\lambda = 1 => \hat{y}_k = p_k.$$

Информативность и энтропия

Информативность:

$$In(X_m) = \min_{\sum_k \hat{y}_k = 1} \left(-\frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} \sum_{k=1}^{K} I[y_i = k] \log \hat{y}_k \right)$$

Подставим $\hat{y} = (p_1, \dots, p_K)$ в формулу информативности и получим **информационную энтропию Шеннона**:

$$In(X_m) = -\sum_{k=1}^{K} p_k \log p_k$$

Информативность в задаче классификации: критерий Джини

Предсказание модели:

- Распределение вероятностей классов $(\hat{y}_1, \dots, \hat{y}_k)$.
- Вместо логарифма правдоподобия будем использовать метрику Бриера (MSE от вероятностей).

Информативность:

$$In(X_m) = \min_{\sum_k \hat{y}_k = 1} \frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} \sum_{k=1}^{K} (\hat{y}_k - I[y_i = k])^2$$

Информативность в задаче классификации: критерий Джини

Оптимальное значение достигается на векторе $\hat{y}$, состоящем из выборочных оценок частот классов $(p_1, \ldots, p_k)$

Подставим p_k в информативность:

$$In(X_m) = \frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} \sum_{k=1}^{K} (p_k - I[y_i = k])^2$$

Вывод критерия Джини

Подставим p_k в информативность:

$$In(X_m) = \frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} \sum_{k=1}^{K} (p_k - I[y_i = k])^2 =>$$

$$In(X_m) = \sum_{k=1}^{K} p_k (1 - p_k)^2 + \sum_{k=1}^{K} (1 - p_k) p_k^2$$

Вывод критерия Джини

$$In(X_m) = \frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} \sum_{k=1}^{K} (p_k - I[y_i = k])^2 =>$$

$$In(X_m) = \sum_{k=1}^{K} p_k (1 - p_k)^2 + \sum_{k=1}^{K} (1 - p_k) p_k^2 =>$$

$$In(X_m) = \sum_{k=1}^{K} p_k (1 - p_k)$$

Это критерий Джини.

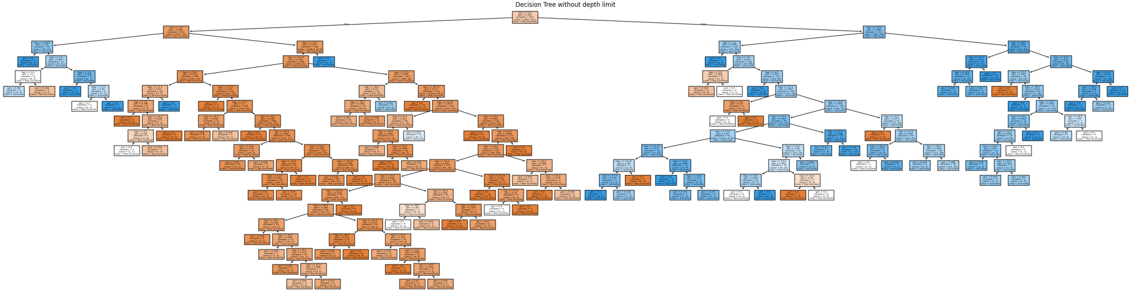
Переобучение решающих деревьев

Решающие деревья склонны к переобучению:

- **Глубокие деревья**: С увеличением глубины дерева модель становится сложнее и может начать запоминать тренировочные данные, включая шум.
- **Идеальная подгонка тренировочных данных**: Без механизмов регуляризации деревья могут создавать слишком специфичные правила, которые идеально подходят для тренировочных данных, но не обобщаются на тестовые данные.

Пример overfitting-а

Решающее дерево, где для каждого примера в обучающей выборке есть отдельный лист



Остановка роста дерева и регуляризация

- Основные критерии остановки:
  - Ограничение по максимальной глубине дерева.
  - Ограничение на минимальное количество объектов в листе.
  - Ограничение на максимальное количество листьев в дереве.
  - Требование, чтобы функционал качества IG улучшался не менее чем на выбранный процент при разбиении.
- Методы остановки:
  - Pre-pruning (early stopping) — проверка критериев во время построения дерева.
  - Pruning — построение полного дерева и последующая стрижка.

Категориальные признаки

Среди признаков, которые мы хотим рассматривать, могут быть категориальные признаки

- Деревья могут работать с категориальными переменными, создавая разбиения по подмножествам значений признака

Категориальные признаки

Среди признаков, которые мы хотим рассматривать, могут быть категориальные признаки

- Деревья могут работать с категориальными переменными, создавая разбиения по подмножествам значений признака
- **Проблема:** при большом количестве значений M число возможных разбиений равно $2^{M-1} - 1$. Это очень много

Категориальные признаки

Среди признаков, которые мы хотим рассматривать, могут быть категориальные признаки

- Деревья могут работать с категориальными переменными, создавая разбиения по подмножествам значений признака
- **Проблема:** при большом количестве значений M число возможных разбиений равно $2^{M-1} - 1$. Это очень много
- **Решение:** упорядочивание значений категориального признака.

Упорядочивание категориальных признаков

Пример

- **Бинарная классификация**: упорядочивание по неубыванию доли объектов класса 1.
- **Регрессия**: упорядочивание по среднему значению целевой переменной.
- Оптимальные сплиты по этим порядкам соответствуют лучшим разбиениям среди всех возможных.

Работа с пропусками

В данных могут быть пропуски, которые нужно обрабатывать. Деревья решений с этим могут бороться.

- При выборе сплитов объекты с пропущенным значением игнорируются.
- После выбора сплита объекты с пропусками в признаке отправляются в оба поддерева

Работа с пропусками

Этап применения:

- Объект с пропущенным значением признака x_i отправляется в обе ветки
- Предсказания усредняются с теми же весами:

$$\hat{y} = \frac{|X_l|}{|X_m|}\hat{y}_l + \frac{|X_r|}{|X_m|}\hat{y}_r$$

- В классификации даёт вероятность класса 1, в регрессии — предсказание целевой переменной
- Можно ввести дополнительное значение «пропущено» и рассматривать его как отдельную категорию

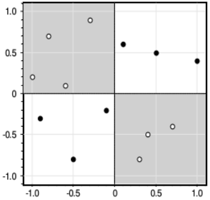
Алгоритм построения дерева решений

Основные шаги:

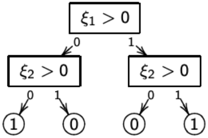
1. Создаётся корневой узел на основе наилучшего разбиения.
2. Тренировочный набор разбивается на 2 поднабора: Всё, что соответствует условию разбиения, отправляется в левый узел, остальное — в правый узел.
3. Рекурсивно повторяются шаги 1-2 для каждого поднабора, пока не будет достигнут один из критериев остановки:
   - Максимальная глубина.
   - Максимальное количество листьев.
   - Минимальное количество наблюдений в листе.
   - Минимальное снижение загрязнения в узле.

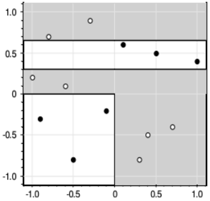
Научились ли мы оптимально строить деревья?

Нет

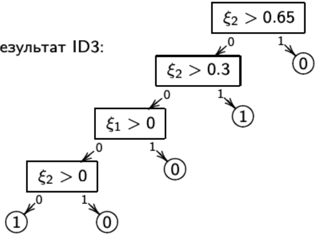


Оптимальное дерево для задачи XOR:





Результат ID3:



http://www.machinelearning.ru/wiki/images/archive/9/97/20140227072517!Voron-ML-Logic-slides.pdf

Итоги

Деревья решений
Преимущества:

- Простая интерпретируемость
- Не требуется особой подготовки тренировочного набора
- Высокая скорость обучения и прогнозирования

Недостатки:

- Поиск оптимального дерева является NP-полной задачей
- Нестабильность работы даже при небольшом изменении данных
- Возможность переобучения из-за чувствительности к шуму и выбросам в данных