Титульный лист материалов по дисциплине

(заполняется по каждому виду учебного материала)

ДИСЦИПЛИНА	**Технологии извлечения знаний из больших данных** (полное наименование дисциплины без сокращений)
ИНСТИТУТ	ИКБ
КАФЕДРА	**КБ-4 «Интеллектуальные системы информационной безопасности»** полное наименование кафедры)
ВИД УЧЕБНОГО МАТЕРИАЛА	**Лекция** (в соответствии с пп.1-11)
ПРЕПОДАВАТЕЛЬ	**Никонов В.В.** (фамилия, имя, отчество)
СЕМЕСТР	**3 семестр 2023/2024 уч. года** (указать семестр обучения, учебный год)

Базовая работа с табличными данными.

Методы решения задачи регрессии: линейная регрессия, метод k-ближайших соседей, решающие деревья, ансамбли моделей

Рассмотрим алгоритмы решения задачи регрессии (предсказание численной величины по имеющимся признакам), а именно такие классические подходы как:

- Линейная регрессия;
- Метод k-ближайших соседей;
- Решающие деревья;
- Ансамбли моделей.

Линейная регрессия

Линейная регрессия — один из базовых и самых простых методов решения задачи регрессии. Модель линейной регрессии хорошо находит линейные зависимости данных, поскольку является линейной комбинацией вектора признаков и вектора весов. Модель линейной регрессии выглядит следующим образом:

$$a_{linreg}(\boldsymbol{x}) = \hat{y} = \sum_{i=1}^{d} w_i x_i + w_0,$$

где $\boldsymbol{x} = (x_1, \ldots, x_d)$ — объекты выборки, $\boldsymbol{w} = (w_1, \ldots, w_d)$ — веса модели $a_{linreg}(\boldsymbol{x}), w_0$ — смещение, $\hat{y}$ — предсказание модели. Если $w_0 = 0$, то $a_{linreg}(\boldsymbol{x}) = \hat{y} = \sum_{i=1}^{d} w_i x_i = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$.

Рассмотрим более простой вариант линейной регрессии, когда мы пытаемся восстановить зависимость целевой переменной от одного признака. Пусть целевая переменная зависит от входного признака следующим образом:

$\boldsymbol{y} = w_0 + w_1\boldsymbol{x} + \boldsymbol{\varepsilon}$, где $\boldsymbol{y} = (y_1, \ldots, y_n)$ — целевая переменная, $\boldsymbol{x} = (x_1, \ldots, x_n)$ — входные данные, w_0, w_1 — параметры, которые мы будем оценивать, и $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)$ — шум, нормальное распределение с мат.

ожиданием $\mathbb{E}(\epsilon_i|x_i) = 0$ и дисперсией $\mathbb{V}(\epsilon_i|x_i) = \sigma^2$, т. е. $(\epsilon_i|x_i) \sim \mathcal{N}(0, \sigma^2)$. Тогда $y_i|x_i \sim \mathcal{N}(\mu_i, \sigma^2)$, где $\mu_i = \omega_0 + \omega_1 x_i$. Оценим коэффициенты w_0 и w_1 и с помощью **метода максимального правдоподобия**. В нашем случае правдоподобие будет иметь вид:

$$\mathcal{L} = \prod_{i=1}^{n} f(x_i, y_i) = \prod_{i=1}^{n} f_x(x_i) f_{y|x}(y_i|x_i) = \prod_{i=1}^{n} f_{y|x}(y_i|x_i) = \mathcal{L}_1 \cdot \mathcal{L}_2$$

где $\mathcal{L}_1 = \prod_{i=1}^{n} f_x(x_i)$ и $\mathcal{L}_2 = \prod_{i=1}^{n} f_{y|x}(y_i|x_i)$.

Функция $\mathcal{L}_1$ не содержит параметры w_0 и w_1, поэтому рассмотрим подробнее $\mathcal{L}_2$ — условную функцию правдоподобия:

$$\mathcal{L}_2 \equiv \mathcal{L}(\omega_0, \omega_1, \sigma) = \prod_{i=1}^{n} f_{y|x}(y_i|x_i) \propto \sigma^{-n} \exp\{-\frac{1}{2\sigma^2} \sum_i (y_i - \mu_i)^2\}$$

а логарифм от $\mathcal{L}_2$:

$$l(\omega_0, \omega_1, \sigma) = -n \log \sigma - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - (\omega_0 + \omega_1 x_i))^2.$$

Далее путем максимизации $l(\omega_0, \omega_1, \sigma)$ по σ получаем оценку: $\hat{\sigma}^2 = \frac{1}{n} \sum_i \hat{\epsilon}_i^2$.

Мы можем получить оценки параметров w_0 и w_1:

$$\widehat{\omega_1} = \frac{\sum_{i=1}^{n} (x_i - \overline{x_n})\,(y_i - \overline{y_n})}{\sum_{i=1}^{n} (x_i - \overline{x_n})^2}$$

$$\widehat{\omega_0} = \bar{y}_n - \widehat{\omega_1} \bar{x}_n$$

где $\bar{x}_n$ и $\bar{y}_n$ — средние значения $\boldsymbol{x}$ и $\boldsymbol{y}$ соответственно. Полученные оценки идентичны тем, которые могут быть получены и путем использования среднеквадратичной ошибки. Абсолютные ошибки обычно не используются, поскольку они не дифференцируемы и по ним нельзя понять, насколько модель близка к правильному прогнозу.

Перечислим преимущества и недостатки линейной регрессии.

К **преимуществам** данного метода относят легкую имплементацию и интерпретацию: мы можем сказать, почему модель принимает решение, посмотрев на веса w (чем больше вес w_i, тем более важен $i^{ный}$ признак).

Недостатки:

- Плохо работает в случаях, когда в данных существенно нелинейные зависимости. Это происходит очень часто, особенно при наличии категориальных признаков
- Плохо работает, если в данных есть линейно зависимые или похожие друг на друга признаки
- Лучше работает, если входные данные нормализованы

Теперь рассмотрим, как можно бороться с недостатками метода.

Генерация признаков

Линейные модели все-таки могут восстанавливать нелинейные зависимости, но только после преобразования признакового пространства:

$$\boldsymbol{X} = (x_1, \dots, x_d) \rightarrow \varphi(\boldsymbol{x}) = (\varphi_1(\boldsymbol{x}), \dots, \varphi_m(\boldsymbol{x})).$$

Таким образом можно, например, перейти к квадратичным признакам:

$$\varphi(\boldsymbol{x}) = (x_1, \dots, x_d, x_1^2, \dots, x_d^2, x_1x_2, \dots, x_{d-1}x_d).$$

После перехода к квадратичным признакам линейная модель сможет приближать любые квадратичные закономерности. Также можно работать и с *полиномиальными признаками* более высоких порядков, генерировать новые признаки с помощью радиально-базисной функции и т. д. Существуют и некоторые рекомендации по выбору преобразований:

- Если у значений признака j много знаков после запятой, и они все важны, то $\log(x_j)$
- Если имеется периодическая зависимость: $\sin\left(\frac{x_j}{T}\right)$
- Если важна близость к какой-то точке: $exp\,\frac{||x_j - \mu||^2}{\sigma}$

Нормализация

Признаки бывают разными в плане масштаба. Например, если рассматривать данные о квартирах, то площадь, скорее всего, будет порядка

30–200 кв. м., в то время как количество близлежащих продуктовых магазинов вряд ли будет сильно больше 10, близость до метро — от 2 до 60 мин. Чтобы привести все признаки к единому масштабу, используют *нормализацию*. Наиболее популярными являются следующие типы:

- Z-нормализация:

$x' = \frac{x - \underline{x}}{\sigma}$, где $\underline{x}$ — среднее значение выборки, σ — среднеквадратичное отклонение. В этом случае большинство значений попадет в интервал $(-3\sigma;\ 3\sigma)$.

- Минимакс-нормализация:

$x' = \frac{x - \min[X]}{\max[X] - \min[X]}$, где $\min[X]$ и $\max[X]$ — минимальное и максимальное значения выборки соответственно. В этом случае все значения будут лежать в интервале (0;1), дискретные бинарные значения определяются как 0 и 1.

Если качество модели на обучающей выборке близко к идеальному, а на тестовой выборке гораздо хуже, то это может говорить о том, что модель переобучилась (ее поведение похоже на запоминание ответов из обучающей выборки). Одним из индикаторов переобучения является большая разница в весах признаков. Например, если один признак имеет вес порядка 10^{-3}, а другой 10^{3}. Для предотвращения данного эффекта используют специальный механизм штрафов за разные порядки весов, называемый *регуляризацией*. Наиболее распространенными являются два типа регуляризации:

1) L_1 - регуляризация (Лассо-регрессия). Также используется для отбора признаков $a_{linreg_{lasso}}(\boldsymbol{x}) = <\boldsymbol{w}, \boldsymbol{x}> + \lambda||\boldsymbol{w}||_1$

2) L_2 - регуляризация (гребневая регрессия). Применяется, когда независимые переменные коррелируют друг с другом $a_{linreg_{ridae}}(\boldsymbol{x}) = <\boldsymbol{w}, \boldsymbol{x}> + \lambda||\boldsymbol{w}||_2^2$

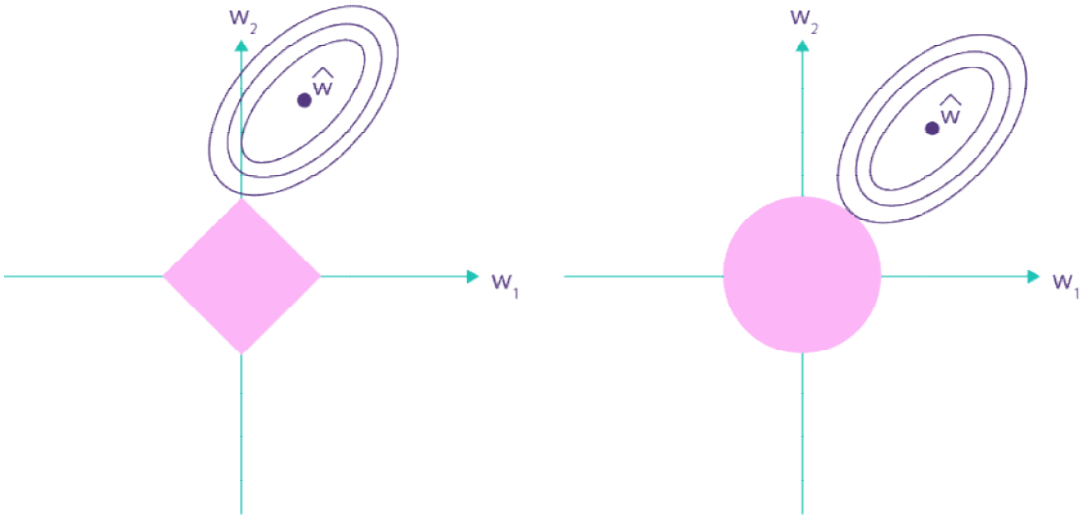

Рис. 1. Сравнение регрессий Лассо (слева) и гребневой (справа), пример для двумерного пространства независимых переменных. Фиолетовые области изображают ограничения на коэффициенты w, эллипсы — некоторые значения функции наименьшей квадратичной ошибки

Рассмотрим подробнее Рис. 1. Поскольку рассматривается функция квадратичной ошибки, которая является выпуклой, задача оптимизации в случае регрессии Лассо сводится к следующей:

$$\begin{cases} \frac{1}{d}\sum_{i=1}^{d}(w_i x_i - y_i)^2 \to \min\limits_{w} \\ \|w\|_1 \le C \end{cases}$$

где C — некоторая константа. Для гребневой регрессии оптимизационная задача выглядит похожим образом, только вместо L_1-нормы используется L_2-норма. На Рис. 1 изображены линии уровня функционала квадратичной ошибки, а также множество, определяемое ограничением $||\boldsymbol{w}||_1 \le C$ для регрессии Лассо и $||\boldsymbol{w}||_2^2 \le C$ для гребневой регрессии.

Решение упомянутой выше оптимизационной задачи определяется точкой пересечения допустимого множества с линией уровня, ближайшей к безусловному минимуму. Из Рис. 1 следует, что эллиптические области могут (в случае регрессии Лассо) касаться углов ромба, и при этом один из коэффициентов будет равен 0 (что невозможно в гребневой регрессии).

Поэтому именно регрессия Лассо может быть использована для отбора признаков: когда из множества всех признаков выбирается такое его подмножество, что признаки в данном подмножестве будут иметь наибольшую важность для обучения и предсказания модели. Также можем сравнить изменения весов для двух типов регуляризации. Из выборки были взяты четыре признака, на которых обучались модели с L_1 и L_2-регуляризацией. Качество модели без регуляризации и с L_1 и L_2 - регуляризацией примерно одинаковое $R^2 \approx 0,62$.

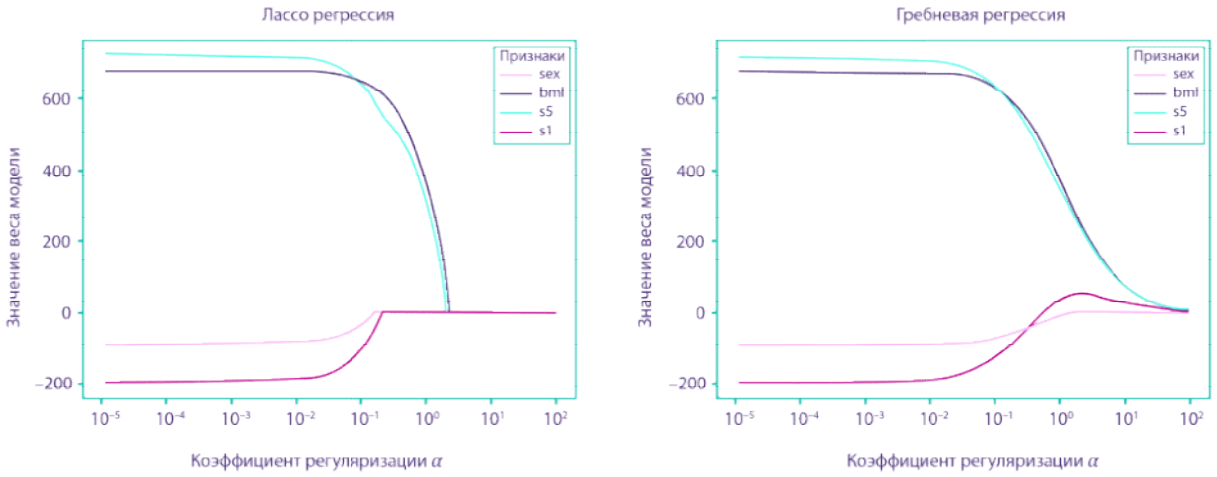

Рис. 2. Зависимость весов модели от силы регуляризации

Метод k-ближайших соседей

Данный метод используется при решении задач классификации и регрессии. Если задано расстояние между объектами, мы можем делать предсказания для какого-то конкретного объекта, используя значения целевой функции его соседей.

Алгоритм прогноза: для нового *объекта x требуется построить прогноз по K-ближайшим к нему объектам* (x_1, ..., x_K). Предсказания модели $a_{knn}(x)$ при этом будут иметь следующий вид:

$$a_{knn}(x) = \hat{y} = \frac{\sum_{k=1}^{K} y_k}{K}.$$ Число ближайших соседей K является гиперпараметром, который нужно подбирать для каждой задачи.

Для выбора числа *k* особенно актуальна дилемма смещения и дисперсии в машинном обучении. Она заключается в следующем:

- Если модель идеально описывает все данные, она переобучилась. Есть вероятность, что она не сохранит предсказательную способность на других данных (обобщающая способность)

- Если модель плохо описывает данные, то она не «переобучилась», но, возможно, и не обучилась совсем

Примеры недообучения и переобучения метода *k*-ближайших соседей представлены на Рис. 3, зависимость ошибки от сложности модели — на Рис. 4.

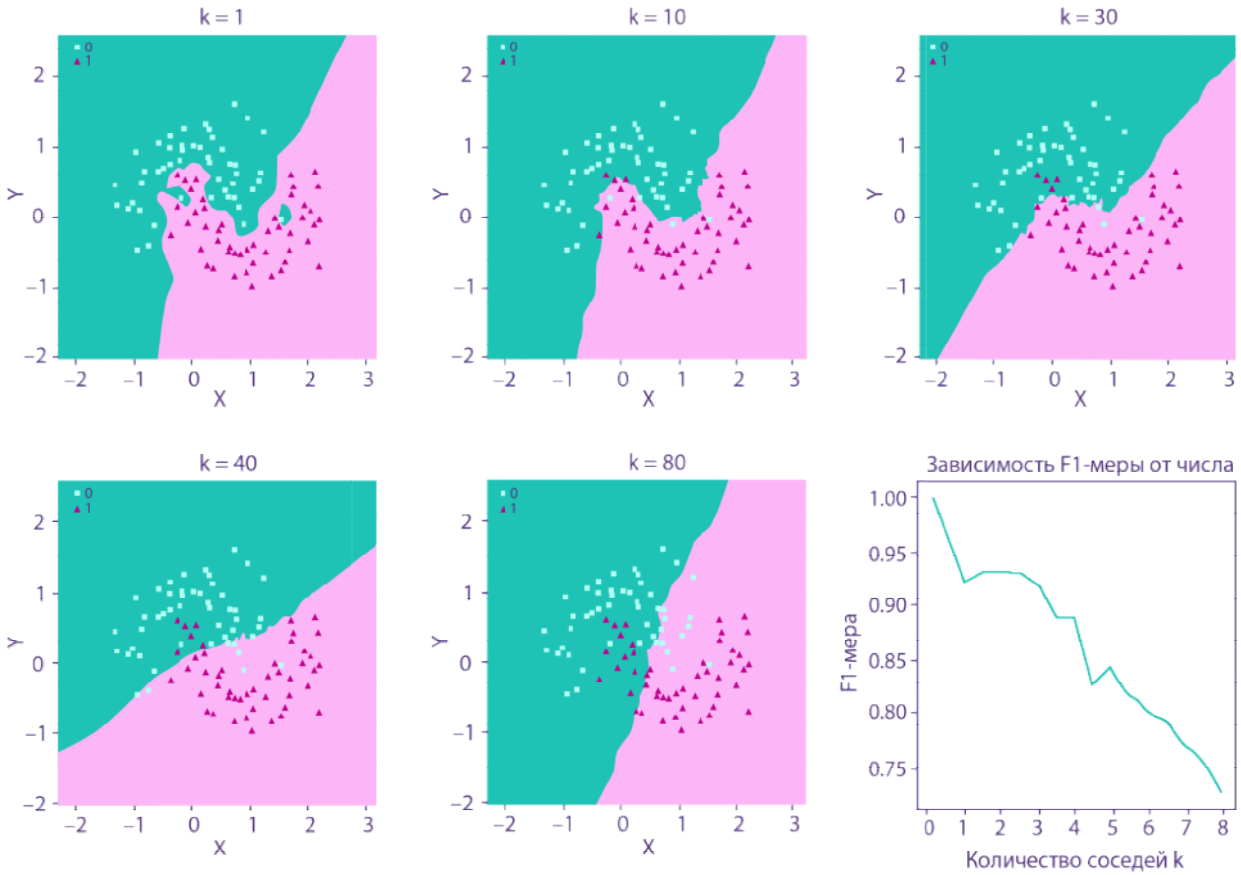

Рис. 3. Зависимость предсказания модели от числа ближайших соседей k. В исходной выборке содержится 100 точек

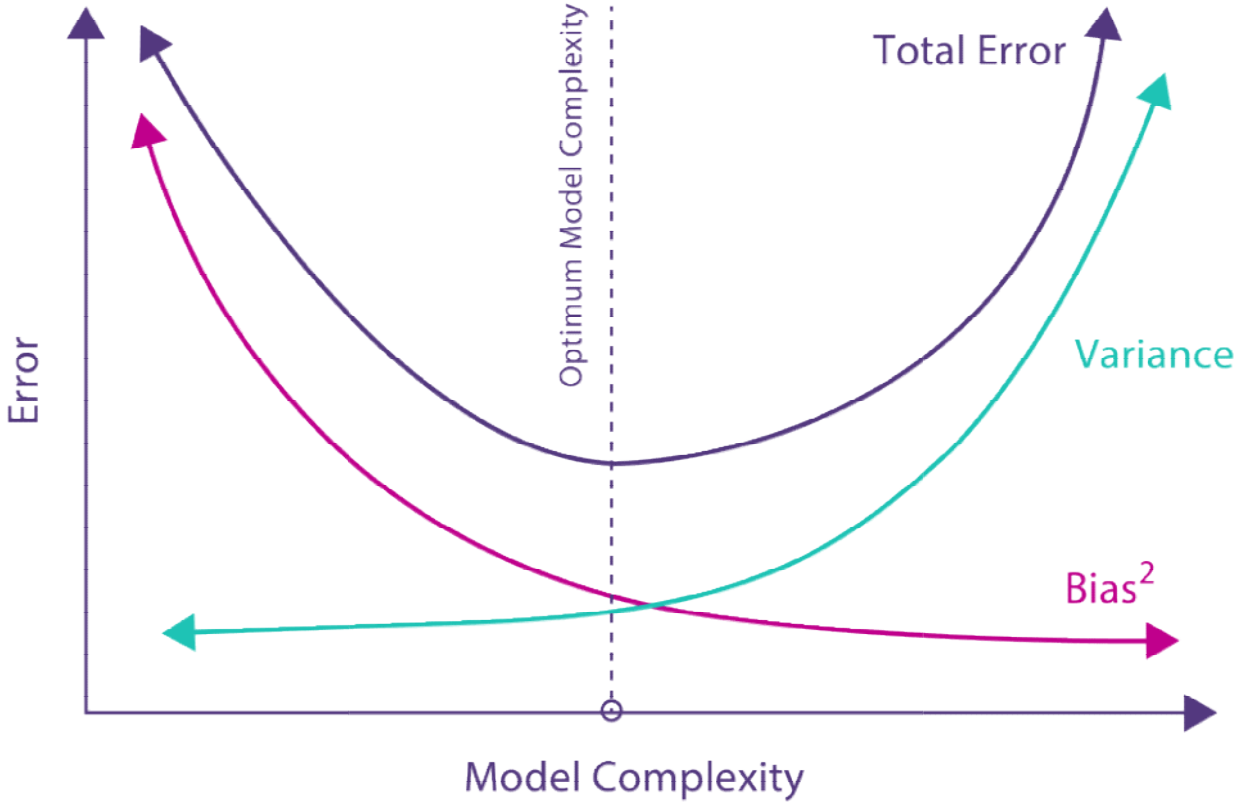

Рис. 4. Зависимость ошибки от сложности модели — демонстрация недообучения модели (левая область), переобучения (правая область) и оптимальной модели (вертикальная пунктирная линия)

Преимуществом данного метода является то, что он интерпретируемый: мы можем сказать, почему модель принимает решение, предъявив похожие объекты из обучающей выборки.

Недостатки:

- Требует задания расстояния между объектами, а также числа соседей, которые используются для принятия решения для нового объекта
- Плохо работает, если входных признаков или объектов много (для каждого объекта выборки необходимо посчитать расстояние между ним и всеми остальными объектами, что требует высокой вычислительной сложности) либо если расстояние между объектами не отражает их близость с точки зрения целевого свойства
- Требует нормализации входных данных

Взвешивание объектов в методе ближайших соседей

Существует вариация метода *k*-ближайших соседей со взвешенным учетом объектов. Пусть обучающая выборка задается объектами $(x_1, ..., x_n)$. Упорядочим их относительно рассматриваемого объекта x: $\rho(x, x_{i_1}) \leq \rho(x, x_{i_2}) \leq \ldots \leq \rho(x, x_{i_n})$.

Обозначим $z_1 = x_{i_1}, \ldots, z_K = x_{i_K}$.

Тогда модель для решения задачи регрессии будет иметь вид:

$$a_{knn_{weighted}}(x) = \hat{y} = \frac{\sum_{k=1}^{K} y_i w(k, \rho(x, x_k))}{\sum_{k=1}^{K} w(k, \rho(x, x_k))}.$$

Примеры весов:

1. Веса, зависящие от индекса:

a) $w_k = \alpha^k$, $\alpha \in (0, 1)$

b) $w_k = \frac{K+1-k}{K}$

2. Веса, зависящие от расстояния:

a) $w_k = \frac{\rho(z_K, x) - \rho(z_k, x)}{\rho(z_K, x) - \rho(z_1, x)}$, если $\rho(z_K, x) \neq \rho(z_1, x)$; иначе 1

b) $w_k = \frac{1}{\rho(x, z_k)}$

Решающие деревья

Решающие деревья используются при решении задач классификации и регрессии. В отличие от линейной регрессии не ищут линейные закономерности в данных. Решающее дерево чаще всего строится по принципу наилучшего разделения или, в математических терминах, минимизации энтропии. Процесс построения **решающих деревьев** заключается в последовательном, рекурсивном разбиении обучающего множества на подмножества с применением решающих правил в узлах. Процесс разбиения продолжается до тех пор, пока все узлы в конце всех ветвей не будут объявлены листьями (либо пока не будет достигнута максимальная заданная глубина дерева). Пример решающего дерева представлен на Рис. 5.

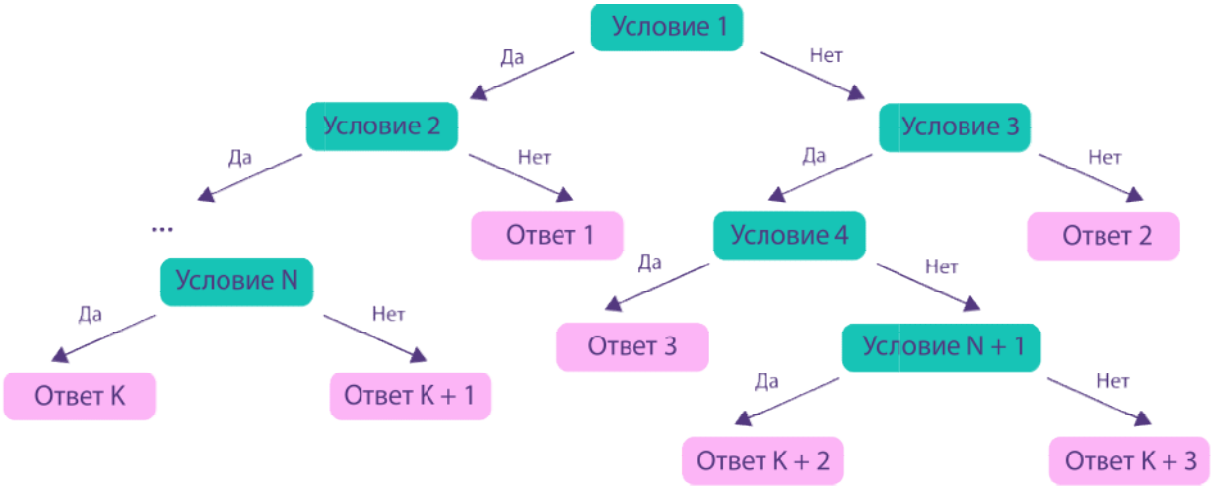

Рис. 5. Решающее дерево.

Преимущества данного метода:

- Интерпретируемый: на каждом шаге алгоритм рекурсивно проверяет, удовлетворяет ли объект тому или иному условию, и в зависимости от результата переходит к новому условию, пока не станет очевиден ответ (т. е. пока объект не окажется в листе дерева)
- Не требует нормализации входных данных
- Позволяет оценить модель при помощи статистических тестов, что дает возможность оценить надежность модели

Недостатки:

- Не могут выявить линейную зависимость
- Проблема получения оптимального дерева является NP-полной задачей — практическое применение алгоритма деревьев решений основано на эвристических алгоритмах, таких как алгоритм «жадности», где единственно оптимальное решение выбирается локально в каждом узле. То есть такие алгоритмы не могут обеспечить оптимальность всего дерева в целом

Чаще всего используется не одно дерево, а несколько, т. е. *ансамбль* (лес решающих деревьев).

Ансамбли моделей

Ансамбли объединяют множество простых моделей, которые по отдельности были бы слабыми и недообученными. Ансамбли комбинируют

решения нескольких моделей с целью улучшения качества обобщающих и предсказательных свойств. Ансамбли можно собирать несколькими способами, которые будут подробнее рассмотрены далее в курсе.

Преимуществом и в какой-то мере недостатком ансамблевых алгоритмов служит эффективность работы только на выборках большого размера, от тысячи объектов. На малых выборках они будут обычно давать слишком сложную модель, склонную к переобучению. Ансамбли также не требуют нормализации входных данных.