УДК 004.89 10.56564/27825264_2025_3_71

НЕКОТОРЫЕ МЕТОДЫ ОЦЕНКИ ВАЖНОСТИ ПЕРЕМЕННЫХ В СТАТИСТИКЕ. II. ДЕРЕВЬЯ КЛАССИФИКАЦИИ

Е. А. Кабакова, В. Г. Панов

Институт промышленной экологии Уральского отделения Российской академии наук, г. Екатеринбург, Россия

Рассмотрены некоторые методы машинного обучения (Machine Learning, или ML), применяемые для оценки значимости переменных. В работе приводится обзор нескольких методов, основанных на деревьях классификации (CART, AdaBoost, Gradient boosted trees, Random Forest, Extra Trees, Rotation Forest), а также сравнение особенностей их применения на примере анализа медико-биологических данных. Дополнительно предлагаются варианты адаптации алгоритмов для обработки баз данных с несбалансированными классами.

Ключевые слова: факторы риска; методы математической статистики; машинное обучение; методы добычи данных; деревья классификации; важность переменных.

1. Введение

Статья посвящена исследованию методов машинного обучения (*Machine Learning,* или *ML*) в контексте оценки значимости переменных. Рассматриваемые ниже подходы базируются на методах математической статистики и алгоритмах искусственного интеллекта, реализующих контролируемое (*supervised learning*) и неконтролируемое (*unsupervised learning*) обучение моделей (т. е. с учителем и без). Эти методы сочетают в себе точность классических математических моделей и эффективность новейших вычислительных технологий.

Методы машинного обучения уже зарекомендовали себя в качестве надежного инструмента обработки графических данных, в том числе медицинских изображений, таких как компьютерная томография, магнитно-резонансная томография и ультразвуковые исследования [1–3]. Такой подход обеспечивает высокую точность анализа и минимизирует влияние субъективных ошибок, связанных с человеческим фактором (невнимательность оператора и т. д.).

Не менее перспективное направление – применение методов машинного обучения для анализа табличных данных. Благодаря высокой вычислительной мощности современных компьютеров, разработке эффективных алгоритмов, гибкости настройки параметров и возможности обработки больших массивов данных такие методы активно развиваются и находят все новые приложения.

Преимущество методов машинного обучения заключается, в частности, в способности алгоритмов выявлять сложные нелинейные зависимости между переменными. Это дает возможность строить более точные модели (как классификации, так и регрессии), ранжировать признаки по степени их значимости, прогнозировать отклик исследуемой системы на основании известных значений независимых переменных. Подробнее об особенностях применения алгоритмов будет рассказано далее.

2. Некоторые меры важности переменных на основе деревьев классификации

В данном разделе рассматриваются методы машинного обучения, основанные на построении деревьев решения – классического инструмента интеллектуального анализа данных (*Data Mining*) и одного из ключевых подходов в машинном обучении.

Основная концепция данного метода заключается в создании иерархической структуры, визуализирующей правила классификации наблюдений на основе логических условий, заданных в узлах этой структуры [4]. Алгоритм построения дерева решений реализует рекурсивное разбиение исходной выборки данных по значениям признаков (атрибутов). В каждом узле алгоритм выбирает наиболее подходящий признак для разделения данных, максимизирующий критерий качества (например, прирост информации или уменьшение неопределенности Джини). Разделение продолжается до достижения условий остановки, таких как достижение максимальной глубины дерева или минимально допустимого количества объектов в терминальном узле. На базе таких деревьев часто создаются более сложные ансамблевые модели.

В нашей статье приводится обзор нескольких методов, основанных на деревьях классификации: *CART*, *AdaBoost, Gradient boosted trees*, *Random Forest*, *Extra Trees, Rotation Forest*. В каждом из них применяются различные алгоритмы для оценки важности переменных, поэтому получение примерно одинаковых списков наиболее важных переменных можно понимать как свидетельство того, что эти списки отражают объективную иерархию данных переменных по важности.

CART

Метод классификации и регрессии с помощью деревьев принятия решений (*Classification and regression trees,* или *CART*) – это базовый алгоритм построения деревьев, на основе которого получили развитие более сложные ансамблевые методы. Он был предложен Брейманом в 1984 г. для построения бинарного дерева, в каждом узле которого обучающая выборка разбивается на две части. Этот метод применяется для классификационных моделей, когда для каждого наблюдения из обучающей выборки заранее известен класс [5].

На вход подается обучающая выборка в виде матрицы предикторов X с элементами $x_{j,i}$, где индекс j характеризует некий признак (j = 1, …, M), индекс i (i = 1, …, N) – i-е значение данного признака, а также вектора Y, состоящего из значений классов y_i. Далее будет использоваться обозначение x_j, подразумевающее набор всех значений j-го предиктора. Рассмотрим стартовый родительский узел дерева t_p и его правый и левый дочерние узлы t_r, t_l соответственно (см. рис. 1). Пусть x_j^R – пороговое значение переменной, обеспечивающей наилучшее разделение данных.

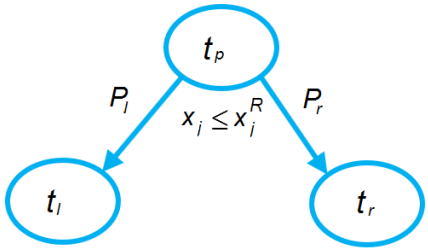

Рис. 1. Алгоритм разделения *CART*

Однородность дочерних узлов определяется так называемой функцией загрязненности $f(t)$, где t – узел дерева. Максимальная однородность дочерних узлов достигается в результате максимизации изменения функции примеси (1):

$$\Delta f(t) = f(t_p) - P_l f(t_l) - P_r f(t_r), \quad (1)$$

где $P_{l(r)}$ – вероятность принадлежности наблюдения к левому (правому) дочернему узлу.

Таким образом, поиск оптимального разделения для каждого узла осуществляется по правилу (2):

$$\hat{x}_j^R = \underset{x_j \le x_j^R,\, j=1,\ldots,M}{\arg\max} \left[f(t_p) - P_l f(t_l) - P_r f(t_r) \right], \quad (2)$$

где x_j^R – возможное пороговое значение переменной;

$\hat{x}_j^R$ – итоговое (экспериментальное) пороговое значение переменной.

В качестве функции «загрязнения» в данной работе по умолчанию предпочтение отдается неопределенности Джини. Пусть база данных (БД) *D* образована *k* классами и рассматривается в некотором фиксированном узле дерева решений. Пусть p_i – вероятность того, что точка данных (наблюдение) принадлежит классу *i.* Тогда неопределенность Джини вычисляется по формуле (3). Интерпретировать ее можно следующим образом: *Gini(D)* – это математическое ожидание доли ложно классифицированных элементов с приписанными к ним метками p_i.

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2 = \sum_{i=1}^{k} p_i (1 - p_i) \quad (3)$$

Если БД *D* некоторым условием (атрибутом) *A* разделяется на части D_1 и D_2 размеров n_1 и n_2 соответственно, $n_1 + n_2 = n$ – объем БД *D,* то неопределенность Джини по этому атрибуту определяется формулой (4):

$$Gini_A(D) = \frac{n_1}{n} Gini(D_1) + \frac{n_2}{n} Gini(D_2) \quad (4)$$

При определении атрибута, который надо выбрать в узле дерева решений, выбирают такой из них, для которого $Gini_A$ минимально. Это необходимо для получения наиболее эффективного разделения выборки в данном узле.

С точки зрения определения важности переменной алгоритмы построения дерева классификации предписывают считать ту из них более важной, которая находится ближе к корню дерева, т. е. находится выше в обычном изображении дерева решений, при котором оно растет вниз.

Иными словами, на первой итерации строятся всевозможные плоскости, разбивающие пространство исследуемых обучающих данных на две части. Далее из них выбирается плоскость, обеспечивающая наиболее полное выделение какого-либо класса данных. На последующих итерациях выбирается наихудший в плане выделения класса узел, требующий разбиения. Данный процесс продолжается до момента достижения предела по количеству узлов или пока качество модели не перестанет улучшаться. В этом случае имеет место риск переобучения (когда модель «заучивает» выявленные закономерности, шумовые составляющие и случайные артефакты), а потому и проводят упомянутые выше тестовые перекрестные проверки, или кросс-валидацию, с последующим возможным уменьшением глубины дерева. Из преимуществ данного метода можно выделить малые временные затраты на построение деревьев и их удобную интерпретацию. Однако зачастую на выходе получается не самая точная классификационная модель, поскольку каждое разбиение оптимизируется только в рамках текущего узла без учета общего влияния на структуру дерева, следовательно, алгоритм часто находит субоптимальные решения.

AdaBoost

Метод адаптивного бустинга (*AdaBoost*), разработанный Фрейндом и Шапирой в 1995 г. [4], – один из представителей семейства бустинговых алгоритмов (*boosting*). Он стал востребованным благодаря своей гибкости, высокой точности и надежности, а также широкому спектру применений [6]. Основная идея метода заключается в построении ансамбля классификационных деревьев, где каждое последующее дерево представляет собой преобразование более слабого дерева-предшественника и исходного классификатора. Для этого преобразования применяется метод повторной выборки – бутстреп (*bootstrap*). Под бутстрепом подразумевается статистический метод, использующий создание повторных выборок для оценки характеристик

выборки: он позволяет оценить стандартную ошибку, построить доверительные интервалы, проверить статистические гипотезы и многое другое [7].

Последовательность объединения простых моделей основана на следующем правиле: выбираются те модели, функциональные зависимости которых от входных данных *x* и выходных значений *y* максимизируют снижение функции потерь. Под функцией потерь подразумевается функция, отражающая соответствие прогнозных результатов модели истинным значениям, т. е. мера несоответствия предсказанных моделью значений реальным данным. В *AdaBoost* используется экспоненциальная функция потерь, учитывающая на каждом этапе обучения изменение веса примеров. В общем виде эта функция записывается как (5):

$$\psi(y, f) = \exp(-\bar{y}f), \tag{5}$$

где y – это значения индикаторной переменной, $y \in \{0,1\}$;

$\bar{y}$ – это преобразованные значения y, $\bar{y} = 2y - 1$;

f – это функциональная зависимость между входными данными.

Пусть имеется набор данных, состоящий из *n* точек, где предикторы $x_i \in R$, а искомая индикаторная переменная $y_i \in \{-1,1\}$, тогда $w_t(x_i, y_i)$ – вес каждой *i*-й точки на итерации *t*. Инициализация процесса состоит в задании равномерного распределения на данных, т. е. каждому значению x_i приписывается вес, равный 1/*n*, но на каждой итерации веса неправильно классифицированных объектов увеличиваются, а потому базовый классификатор h_t вынужден концентрироваться на «проблемных» примерах из тренировочного набора [8]. Таким образом, построение классификатора на любой итерации *t* состоит в определении такой функции h_t, которая минимизирует взвешенную ошибку (6), считая, что $\{w_{i,t}\}_{i=1}^{n}$ – система весов на итерации *t*. Стоит отметить, что функция h_t имеет некоторый подгоночный параметр и может применяться к любому значению выборки (на независимых переменных).

$$\varepsilon_t = \sum_{i=1}^{t} w_i \left[1_{y \neq f(x)}\right], \tag{6}$$

где *t* – номер итерации, *t* = 1, 2, …, *M*;

$\left[1_{y \neq f(x)}\right]$ – скобка Айверсона.

После получения базового классификатора h_t можно определить его общую важность или вес (7), характеризующий точность.

$$\theta_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \tag{7}$$

После выполнения заданного числа итераций формируется окончательный классификатор (8):

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{M} \theta_t h_t(x)\right) \tag{8}$$

Иными словами, на каждой итерации *AdaBoost* подбирает h_t и рассчитывает θ_t с последующим добавлением нового элемента к накапливающейся взвешенной сумме базовых классификаторов так, чтобы сумма приведенных выше показателей была наименьшей. Схематично псевдокод алгоритма представлен на рис. 2.

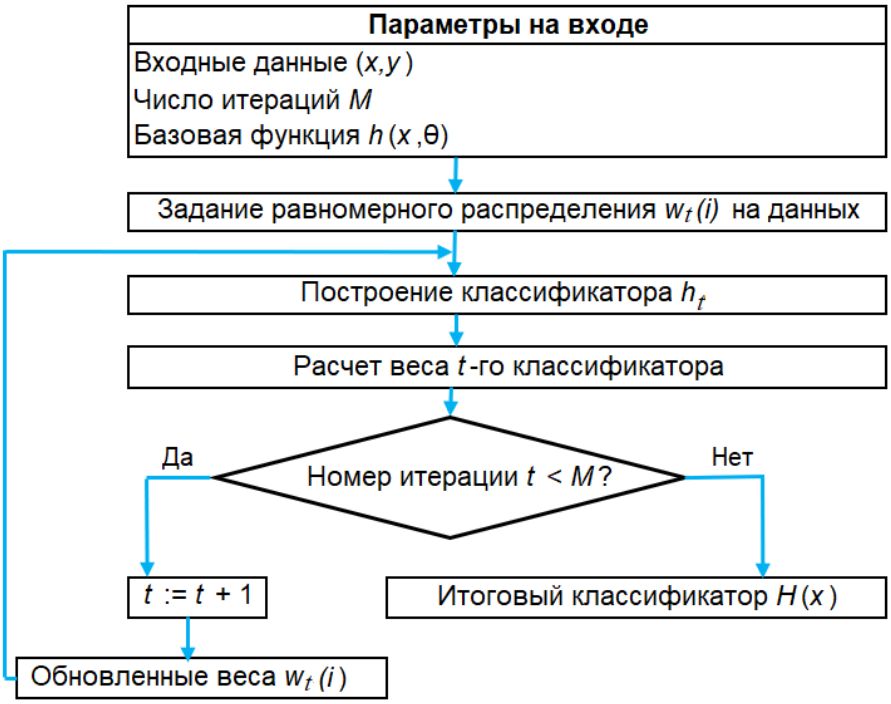

Рис. 2. Псевдокод алгоритма метода *AdaBoost*

Основным преимуществом данного алгоритма можно считать свойство сведения ошибки обучения к нулю за несколько начальных итераций, что позволяет избежать в дальнейшем серьезных ошибок. Изначально метод адаптивного бустинга был предложен в качестве бинарного классификатора, но последующие модификации позволили добиться мультиклассового расширения и возможности анализировать не только табличные данные, но и изображения.

Gradient Boosting

Градиентный бустинг (*Gradient Boosting*) – это усовершенствованная версия алгоритма бустинга, предложенная разработчиками *AdaBoost* в 1999 г. Алгоритм основывается на последовательном объединении простых моделей, функциональные зависимости которых от входных данных *x* и выходных значений *y* максимизируют снижение отрицательного градиента функции потерь. В отличие от *AdaBoost*, функция потерь может иметь не только экспоненциальный вид. Среди таких функций для задач машинного обучения в большинстве случаев используется среднеквадратичная ошибка *MSE* (9). Также можно встретить специальные непрерывные функции потерь Гаусса, Лапласа, Хубера и т. д. [9]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad (9)$$

где *n* – полное число точек;

y_i – истинное значение параметра в *i*-й точке;

$\hat{y}_i$ – предсказанное моделью значение параметра в *i*-й точке.

Итак, пусть на вход подается набор данных, состоящий из *n* точек, где предикторы $x_i \in R$, а y_i – значение зависимой переменной (в нашем случае значение индикаторной переменной БСК). Цель построения прогностической модели заключается в нахождении функциональной зависимости между *x* и *y* в виде функции $\hat{f}(x) = \hat{y}$ (10), которой соответствовало бы минимальное значение потерь, записываемая в общем виде как $\psi\left[y, \hat{f}(x)\right]$.

$$\hat{f}(x) = \arg\min_{f(x)} E_x\left[E_y\left(\psi\left[y, f(x)\right]\right) \mid x\right], \quad (10)$$

где $E_y\left(\psi\left[y, f(x)\right]\right)$ – математическое ожидание функции потерь по переменной отклика;

$E_x\left[E_y\left(\psi\left[y, f(x)\right]\right) \mid x\right]$ – математическое ожидание предыдущей функции по отношению к переменной *x*.

Для упрощения задачи можно ограничить пространство поиска функций параметрическим семейством функций *f*(*x*,θ), тогда формула (10) перепишется через формулы (11), (12):

$$\hat{f}(x) = f(x, \hat{\theta}), \quad (11)$$

$$\hat{\theta} = \arg\min_{\theta} E_x\left[E_y\left(\psi\left[y, f(x,\theta)\right]\right) \mid x\right] \quad (12)$$

Обычно при поиске оценок параметров число итераций ограничивают, например, некоторым числом *M*, и, соответственно, оценку $\hat{\theta}$ параметра θ представляют в виде (13):

$$\hat{\theta} = \sum_{i=1}^{M} \hat{\theta}_i, \quad (13)$$

где $\hat{\theta}_i$ – оценка параметра *i*-й итерации.

Наиболее распространенный способ получения оценок неизвестных параметров – метод градиентного спуска. Для него можно записать эмпирическую функцию потерь $J(\theta)$ (14), учитывающую функции потерь для всех *n* точек из набора данных.

$$J(\theta) = \sum_{i=1}^{n} \psi\left(y_i, f\left(x_i, \hat{\theta}\right)\right) \quad (14)$$

Также можно оценить градиент (15) этой функции:

$$\nabla J(\theta) = \left\{\nabla J\left(\theta_i\right)\right\} = \left[\frac{\partial J(\theta)}{\partial J\left(\theta_i\right)}\right]_{\theta=\hat{\theta}^t=\sum_{i=0}^{t-1}\hat{\theta}_i}, \quad (15)$$

где *t* – порядковый номер проводимой итерации.

На основе градиента функции потерь рассчитывается новая оценка параметра, который используется для построения элемента ансамбля в качестве так называемого поправочного коэффициента: $\hat{\theta}_t \leftarrow -\nabla J(\theta)$ (16):

$$\hat{\theta}_t = -\lambda \nabla J\left(\hat{\theta}^{t-1}\right), \quad (16)$$

где λ – параметр, задающий скорость градиентного спуска (17):

$$\lambda = \arg\min_{\lambda} \sum_{i=1}^{n} \Psi\left(y_i, f\left(x_i, \hat{\theta}^{t-1}\right)\right) \quad (17)$$

Тогда оценка функции на следующей итерации примет следующий вид:

$$\hat{f}_t \leftarrow \hat{f}_{t-1} + \rho_t h\left(x, \theta_t\right),$$

где ρ – шаг градиентного спуска;

$h\left(x, \theta_t\right)$ – параметризованная базовая функция.

Иными словами, алгоритм построения моделей методом градиентного бустинга выглядит следующим образом: на первом шаге создается дерево решений с ограниченным количеством узлов – это начальная функция f_0, где параметр представляет собой некоторую константу. Затем вычисляется разница между истинным значением и прогнозом модели, чтобы определить поправочный

коэффициент или ошибку модели. На основе этой разницы строится следующее дерево. Чтобы избежать переобучения, рекомендуется использовать контрольный набор данных, который не участвовал в обучении модели. Наглядно приведенный выше алгоритм представлен на схеме, изображенной на рис. 3.

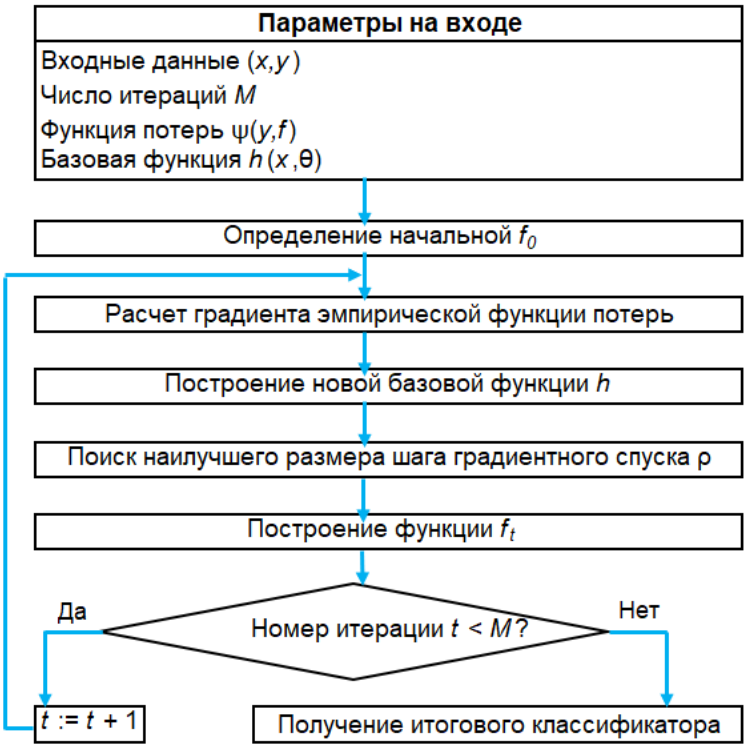

Рис. 3. Алгоритм метода *Gradient Boosting*

Метод демонстрирует высокую точность итоговых моделей, а также легко адаптируется и настраивается для различных задач. Тем не менее его практическая реализация требует значительных ресурсов памяти, что приводит к снижению скорости выполнения операций. Это может быть критично при работе с большими объемами данных. Несмотря на то, что некоторые расчеты могут выполняться параллельно, основное обучение происходит последовательно, что замедляет процесс обработки информации.

Random Forest

Метод случайного леса (*Random forest*) был предложен Лео Брейманом в 2001 г. [10]. Это алгоритм машинного обучения, в основе которого заложен ансамбль деревьев решений, задействованных в качестве базового классификатора. По аналогии с предыдущими пунктами это определение можно представить в виде $\{h(x,\theta_k), k=1,...\}$, где $\{\theta_k\}$ – независимые случайные векторы. Каждое дерево в ансамбле строится на основе случайной подвыборки обучающего множества и признаков, а также случайной параметрической величины θ_k, индивидуальной для каждого дерева. После *k* итераций на выходе окажется система $\{h_1(x), h_2(x), ..., h_k(x)\}$, на основе которой строится модель с функцией принятия решений (18), аналогичной рассмотренным в предыдущих пунктах. Для получения итоговой оценки модели результаты классификации всех деревьев усредняются. Этот подход позволяет снизить дисперсию ошибки по сравнению с одиночными деревьями решений и повысить общую точность предсказания.

$$H(x) = \arg\max_{Y} \sum_{i=1}^{k} [h_i(x) = y], \quad (18)$$

где $H(x)$ – набор моделей классификации;

h_i – единичное дерево классификации;

y – истинное значение класса;

$[h_i(x) = y]$ – скобка Айверсона.

Единичное дерево решений строится согласно принципу, описанному в пункте про CART. Основное отличие случайного леса заключается в том, что он включает в себя множество таких деревьев, каждое из которых создается на основе разных подвыборок исходного множества данных (метод бутстрепа).

В общих чертах алгоритм заключается в следующем: из исходного набора данных, признаков или функций собирается некоторая выборка, на основе которой создается дерево (рис. 4). Подобные шаги повторяются сотни или тысячи раз. Итоговая модель представляет собой объединение всех выращенных деревьев.

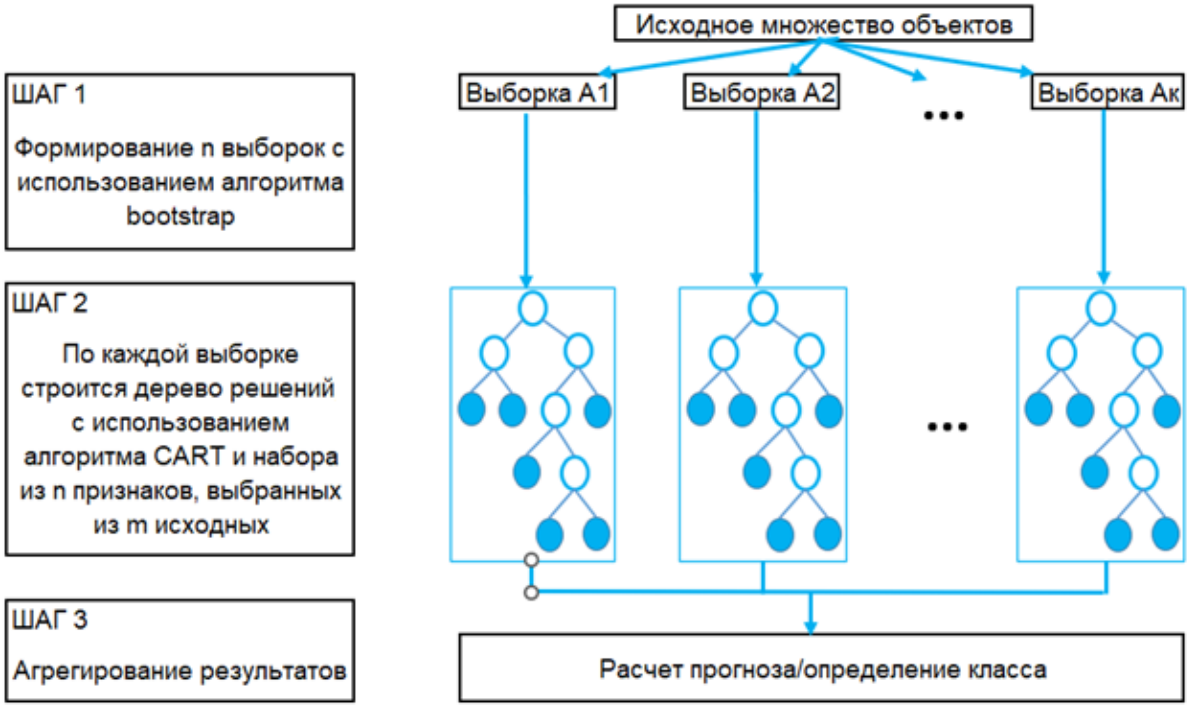

Рис. 4. Блок-схема алгоритма *Random Forest*

Среди преимуществ можно отметить отсутствие необходимости в тестовой выборке и высокое качество модели; классический случайный лес также может использоваться для ранжирования переменных. Однако этот метод характеризуется огромными деревьями и сложной интерпретацией, требуя при этом значительных временных затрат. В случае работы с табличным представлением данных он часто уступает градиентному бустингу, однако в области обработки изображений демонстрирует превосходные результаты. Важно отметить, что с увеличением числа признаков, используемых при обучении одного дерева, возрастает корреляция между деревьями, что снижает эффект ансамблирования. С другой стороны, сокращение числа признаков негативно сказывается на качестве самих деревьев, приводя к снижению их классификационной и прогностической способности.

Extra Trees

Метод дополнительных деревьев (*Extra Trees*) был предложен П. Геуртом, Д. Эрном и Л. Вехенкелем в 2006 г. Он имеет сходство со случайным лесом: он также

используется для классификации и регрессии, но отличается построением отдельных деревьев в группе и считается более устойчивым к зашумленности данных благодаря строгой рандомизации предикторов при создании обучающей выборки и выбора точки отсечения при разделении узла дерева [11, 12]. Основная идея заключается в создании ансамбля из нескольких статистически независимых деревьев решений, структура которых не зависит от выходных значений обучающей выборки. Для каждого дерева используется подвыборка из k элементов исходного массива данных для выделения признака, наиболее подходящего для разделения исследуемых данных на две части. Для определения качества модели, как и в предыдущих случаях, используется мера Джини. Полученная случайная выборка объектов приводит к созданию множества некоррелированных деревьев решений, каждое из которых «обрезается» – остаются лишь наиболее важные предикторы.

Математическое описание алгоритма схоже с построением случайного леса: модели строятся на основе деревьев решений, итоговый классификатор задается формулой (18). Отличие заключается в упомянутом выше разделении узла, псевдокод которого для *Extra Trees* приведен на рис. 5. В алгоритме *Random Forest* рассматривается случайное подмножество признаков и выбирается наилучшее для максимального разделения классов и минимизации неопределенности Джини, тогда как в алгоритме *Extra Trees* вместо оптимального разбиения подбирается набор случайных, из которых уже выбирается лучшее, благодаря чему уменьшается влияние зашумленности данных на итоговую модель. Данный подход повышает вычислительную эффективность, но может снизить точность модели.

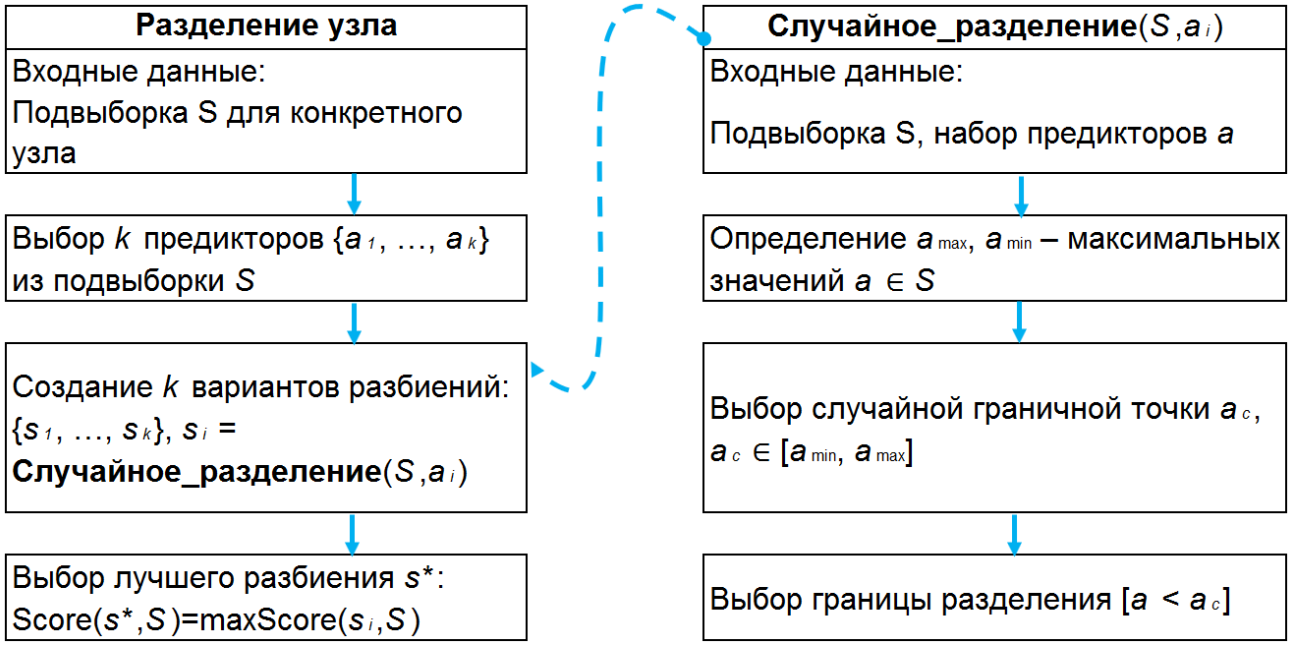

Рис. 5. Псевдокод для *Extra Trees*

Rotation Forest

Интересен также алгоритм леса вращений (*Rotation Forest*), предложенный в 2006 г. коллективом ученых, в составе J. Rodriguez, L. Kuncheva, C. Alonso. Он основан на чувствительности деревьев решений к вращению пространства признаков с использованием метода главных компонент (*Principal component analysis,* или *PCA*) [13]. Несмотря на эмпирический подход (т. е. отсутствие полной математической формализации), модели леса вращений зарекомендовали себя в качестве высокоточного алгоритма для анализа медико-биологических данных: они был успешно применены для диагностики ишемической болезни сердца, прогнозирования тяжести протекания болезни Паркинсона, заболеваний почек и т. д. [14–16]. Высокий

уровень точности достигается за счет сохранения всех основных предикторов и обучения каждого базового классификатора с использованием всего набора данных.

На вход подается набор данных, состоящий из *n* точек, где предикторы $x_i \in R$, а y_i – значение зависимой переменной. Для построения модели ротационного леса первым делом определяется размер ансамбля *L* (количество деревьев), количество подмножеств признаков *S* и размер выборки *P*. Для всех деревьев *l* = 1, …, *L* исходное множество признаков *F* разбивается на F_1, …, F_S – непересекающиеся подмножества признаков, к которым будет применен метод главных компонент. Выбор части объектов увеличивает разнообразие итоговых поворотов, что, в свою очередь, приводит к снижению статистической зависимости между базовыми классификаторами. Для каждого такого подмножества F_s (где *s* = 1, 2, …, *S*) выбираются столбцы с объектами $X_{l,s}$ из *X*, принадлежащими F_s. На следующем шаге случайным образом составляется в виде матрицы объекта-признака $X'_{l,s}$ подмножество классов из $X_{l,s}$ со строками, соответствующим объектам подмножества. Далее возможны два варианта действий: можно непосредственно к этому подмножеству $X'_{l,s}$ применить метод главных компонент либо сформировать на основе 75 % отобранных данных подмножество X''_{ij} и применить *PCA* уже к нему для нахождения *M* главных компонент. Применение метода *PCA* позволяет получить новый ортонормированный базис $a_{l,s}^1,...,a_{l,s}^M$, соответствующий направлениям с наибольшей дисперсией.

Из найденных компонент составляется матрица $A_{l,s} = \left[a_{l,s}^1,...,a_{l,s}^M \right] \in \mathbb{R}^{M \times M}$. Описанная итерация повторяется *L* раз (для всех *l* = 1, …, *L*). На выходе из матриц $A_{l,1}$, …, $A_{l,S}$ собирается блочно-диагональная матрица A_l (19):

$$A_l = \begin{bmatrix} A_{l,1} & 0 & \cdots & 0 \\ 0 & A_{l,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{l,S} \end{bmatrix}. \quad (19)$$

Реализация описанных шагов и сохранение коэффициентов главных компонент с последующим упорядочением строк данных в массиве A_l (для соответствия порядку в исходном массиве) приводит к получению матрицы вращения R_l, которая применяется к основному набору данных *X*: $X_l = XR_l$. На основе этих данных выращивается дерево алгоритма $\hat{y}_l = f\left(X_l, w_n^l, y\right)$ с учетом веса *n*-й компоненты *l*-дерева w_n^l.

Итоговый прогноз классификатора описывается формулой (20):

$$\hat{y}(x) = \arg\max_y \sum_{l=1}^{L} \left[\hat{y}_l\left(x^T R_l\right) = y \right], \quad (20)$$

где $\left[\hat{y}_l\left(x^T R_l\right) = y \right]$ – скобка Айверсона.

Схематично процесс построения леса вращений представлен на рис. 6, 7. В результате работы данного алгоритма области классификации могут быть аппроксимированы более точно и с меньшим количеством элементов ансамбля. Помимо этого, вращение помогает улучшить разнообразие деревьев и уменьшает переобучение. Данный алгоритм – наиболее молодой из рассмотренных в работе, а потому представляет для нас особый интерес, сопряженный с существенными сложностями его применения ввиду отсутствия готовых пакетов с его программной реализацией.

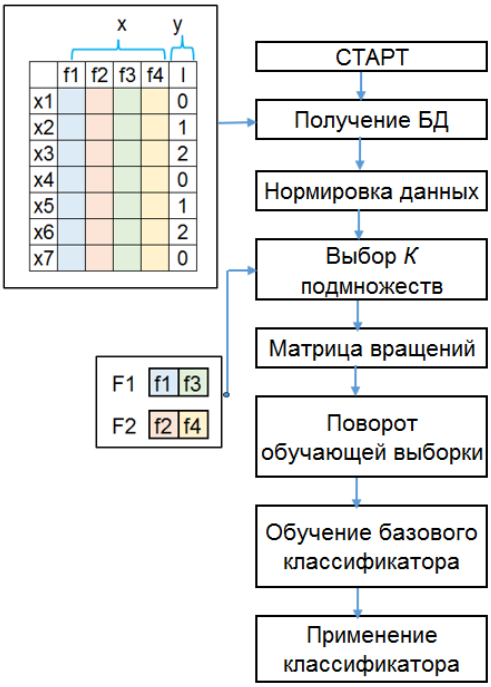

Рис. 6. Блок-схема алгоритма Rotation Forest

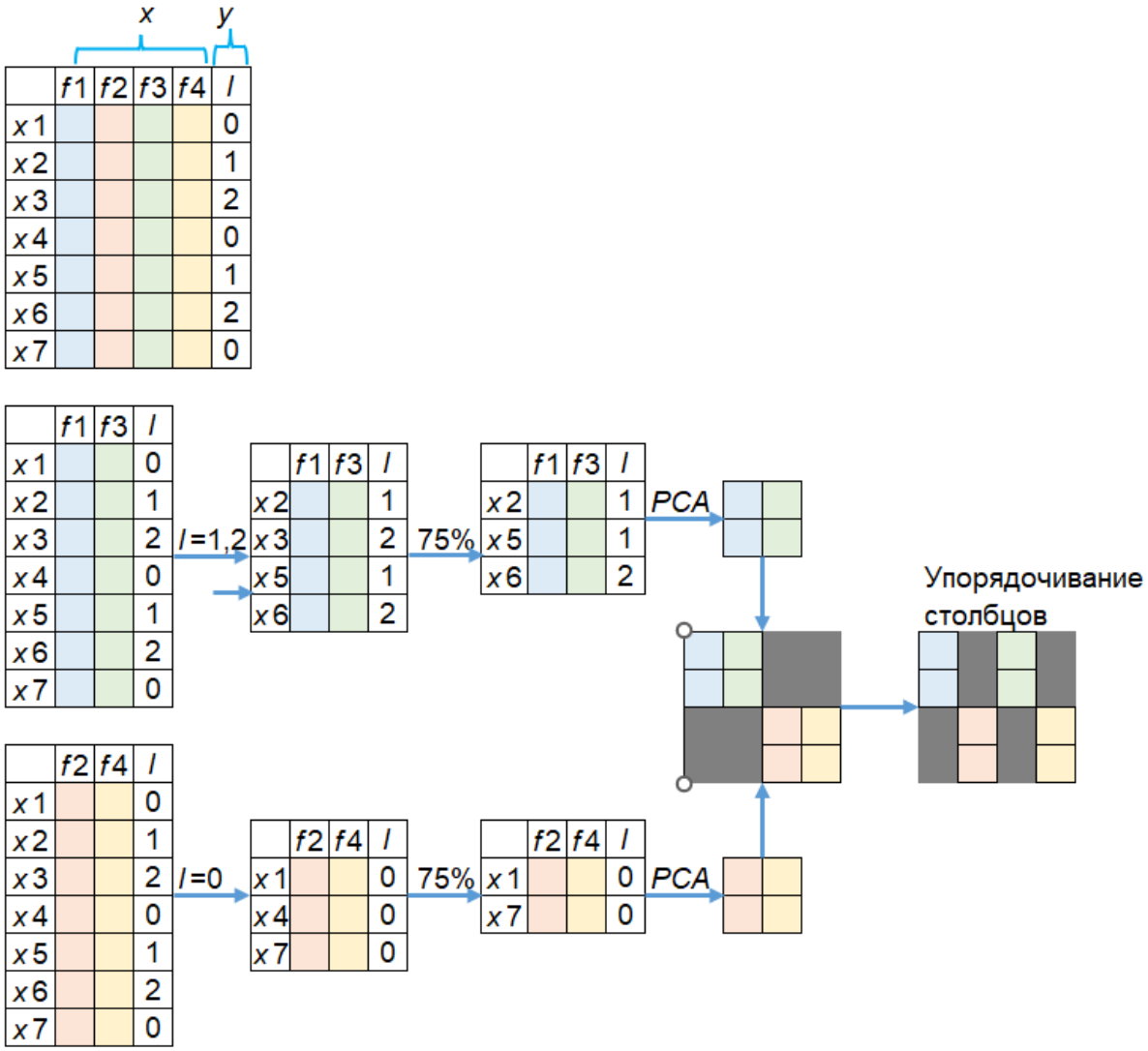

Рис. 7. Схематичное описание построения матрицы вращений

Наглядно сравнение всех рассмотренных методов по основным признакам приведено в табл. 1.

Табл. 1. Сравнение рассмотренных методов

Признак	Метод					
	CART	Boosted Trees	Random forest	Ada Boost	Extra Trees	Rotation Forest
Точность модели	±	+	+	+	±	++
Относительно малый размер	+	–	–	–	–	±
Удобная интерпретация результатов	+	–	–	–	–	–
Малое время расчета	+	–	–	+	–	–
Устойчивость к ошибкам	–	+	+	–	+	+
Возможность распараллеливания вычислений	–	+	+	–	+	+
Отсутствие кросс-валидации	–	–	+	+	–	–

Примечания:

(1) «–» – плохое значение показателей.

(2) «±» – среднее значение показателей.

(3) «+» – хорошее значение показателей.

(4) «++» – отличное значение показателей.

В целом у всех рассмотренных методов добычи данных и машинного обучения есть одно важное преимущество относительно классических методов статистического анализа – они допускают различные модификации и обобщения, а также в любом варианте достаточно устойчивы к наличию корреляций между переменными.

3. Метрики качества

Описанные в данной работе задачи машинного обучения состоят преимущественно в восстановлении зависимости между имеющимися в исходной базе предикторами и откликом бинарной переменной. Однако добиться идеального результата прогнозирования практически невозможно, поскольку реальные данные обладают рядом особенностей, среди которых зашумленность, высокий уровень коррелированности предикторов, несбалансированность классов и др. Поскольку алгоритмы моделей различаются, их результаты могут не совпадать. В связи с этим возникает необходимость использования критериев, которые позволят сопоставить итоговые результаты и установить степень их соответствия реальным (истинным) значениям. Для этого можно использовать несколько стандартных показателей, отличающихся информативностью и простотой расчета. Основной – точность, или *Accuracy*, равный доле верно классифицированных наблюдений (21):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (21)$$

Однако при анализе медико-биологических данных мы зачастую имеем дело с несбалансированными по численности пациентов выборками (к примеру, будут преобладать люди без выявленной интересующей патологии). В результате показатель точности *Accuracy* может оказаться крайне высоким, даже если все сотрудники с подтвержденными диагнозами окажутся среди ложноотрицательных результатов, что в медицинской практике чревато серьезными негативными последствиями.

Другой показатель *Recall* (22) характеризует «полноту» модели, т. е. способность выделять истинноположительные образцы; он аналогичен *чувствительности*, используемой для оценки регрессионных моделей.

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

Precision представляет собой аналог *специфичности* (*Spe*) и демонстрирует способность классификатора избегать ложноположительных результатов; рассчитывается по формуле (23).

$$precision\_score = \frac{TP}{TP + FP} \tag{23}$$

Весьма интересен показатель *F*1 или сбалансированная *F*-мера, рассчитывающаяся как гармоническое среднее между точностью и полнотой модели (24). При трактовке результатов, полученных путем расчета всех описанных показателей, следует пользоваться единым правилом: чем ближе значение показателя к 1, тем лучше.

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{24}$$

Также не лишним будет упомянуть индекс Юдена *J* – разницу между долей истинноположительных результатов и долей ложноположительных результатов. Чем ближе данный коэффициент к единице – тем лучше модель; значения около 0 говорят о случайном угадывании.

Немаловажен при трактовке результатов статистического анализа медико-биологических данных критерий *PSI* (25) – это сводный прогнозный индекс, демонстрирующий общий выигрыш в уверенности в положительных результатах прогнозирования при выполнении диагностического теста. Обратная к данному показателю величина определяет число людей, для которых необходимо провести дополнительные медицинские обследования с целью уточнения диагноза.

$$PSI = \frac{TP}{TP + FP} + \frac{TN}{TN + FN} - 1 \tag{25}$$

И последний коэффициент, который в последние годы набирает особую популярность – коэффициент корреляции Мэтьюса (26) между прогнозируемыми и истинными классами, учитывающий как истинные, так и ложноклассифицированные объекты, что делает его перспективным для задач с несбалансированными классами. Также он считается лучшим показателем для задач с необходимостью учета ошибок классификации [17], т. к. он единственный из всех метрик в случае бинарной классификации принимает высокое значение только в том случае, когда верно определено большинство как положительных элементов, так и отрицательных [18]. Значения коэффициента находятся в интервале [–1; 1]: значения выше 0,3 считаются неплохими, а выше 0,5 свойственны сильным прогностическим моделям; при стремлении результата к 0 можно говорить о случайных предсказаниях; значение –1 соответствует полностью ложной классификации. При возникновении сомнений в корректности результатов допустимо сравнение полученных значений с критерием χ^2 Пирсона, рассмотренным в первой части исследования.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{26}$$

В целом *F*1 и *MCC* предлагают надежные оценки для прогнозов, которые верно классифицируют как положительные, так и отрицательные элементы. Однако эти показатели выдают противоречивые значения, когда прогностическая модель справляется с определением только одного из двух классов. Когда прогноз выдает много истинноположительных результатов, но мало истинноотрицательных (или наоборот), величина *F*1-меры может привести к появлению вводящей в заблуждение информации, в то время как *MCC* всегда выдает результаты, отражающие общие проблемы с прогнозированием.

4. Результаты практического применения методов машинного обучения

Рассмотрим применение методов машинного обучения и добычи данных для анализа важности и ранжирования переменных на основе набора данных из первой части работы (табл. 2).

Таблица 2. Список переменных для анализа

№	Название переменной	Тип переменной
1	Возраст	непрерывная
2	Общий стаж на вредной работе	непрерывная
3	КУТ общий	дискретная
4	КУТ по АПФД	дискретная
5	КУТ по шуму	дискретная
6	КУТ по микроклимату	дискретная
7	КУТ по тяжести трудового процесса	дискретная
8	Отягощенность по ССС в семейном анамнезе	бинарная
9	Онкология в семейном анамнезе	бинарная
10	Инсульт в семейном анамнезе	бинарная
11	ИБС в семейном анамнезе	бинарная
12	Отягощенность по общей наследственности	Бинарная
13	Низкая физическая активность	бинарная
14	Злоупотребление алкоголем	бинарная
15	Рост (см)	непрерывная
16	Вес (кг)	непрерывная
17	ИМТ	непрерывная
18	Окружность талии (см)	непрерывная
19	Окружность талии / Рост	непрерывная
20	Уровень глюкозы в крови	непрерывная
21	Холестерин в крови	непрерывная
22	САД	непрерывная
23	Норма/патология по флюорографии	бинарная
24	Отсутствие/наличие эндокринных заболеваний	бинарная
25	Отсутствие/наличие заболеваний нервной системы	бинарная
26	Отсутствие/наличие БКМС	бинарная
27	Отсутствие/наличие заболеваний класса III (D50-D89)	бинарная
28	Отсутствие/наличие заболеваний класса XI (K00-K93)	бинарная
29	Отсутствие/наличие заболеваний класса XVIII (R00-R99)	бинарная

Примечания:

- КУТ – класс условий труда по федеральному закону № 426-ФЗ;
- КУТ по АФПД – КУТ при воздействии аэрозолей преимущественно фиброгенного действия;
- ССС – сердечно-сосудистая система;
- ИБС – ишемическая болезнь сердца;
- ИМТ – индекс массы тела: ИМТ = масса тела (кг)/(рост (м))^2;
- САД – систолическое артериальное давление.

Прежде чем приступить к работе, подберем оптимальную глубину деревьев для каждого алгоритма классификации, поскольку это напрямую влияет на качество итоговых моделей. Для выполнения подзадачи можно задействовать обучающую кривую [19], в качестве которой можно взять график зависимости *F*1-меры, учитывающей точность и полноту моделей, от максимальной глубины решающего дерева. Иными словами, можно наглядно увидеть влияние изменения глубины

деревьев на процесс обучения и выявить возможное переобучение моделей, проявляющееся в ухудшении качества модели при проверке на тестовой выборке при высоких показателях на обучающей выборке. Для получения статистически значимых результатов при варьировании глубины деревьев необходимо проводить *n*-кратную перекрестную проверку, где $n \geq 5$. Для того чтобы избежать присутствия повторяющихся значений метрики качества модели в разных обучающих выборках, была введена рандомизация индексов объектов в обучающих выборках для каждого разделения. В результате можно увидеть, что, например, в случае алгоритма *CART* оптимальная глубина равна 3, т. к. с последующим увеличением данного значения ухудшается качество модели (рис. 8). Такое ограничение глубины не позволяет моделям улавливать «шум» и нерелевантные правила. Тем не менее для леса вращений такая настройка глубины не применяется: в данном случае эта процедура требует значительных временных затрат, поэтому было решено от нее отказаться. Этот шаг не оказал существенного влияния на конечные результаты, поскольку, как будет показано далее, лес вращений позволяет строить модели с наилучшими показателями качества среди шести задействованных на данном этапе алгоритмов.

Дополнительно получено наглядное подтверждение преимущества алгоритма *Random Forest*, который не склонен к переобучению (рис. 9) за счет большого количества деревьев, каждое из которых строится на случайно отобранном наборе признаков и объектов.

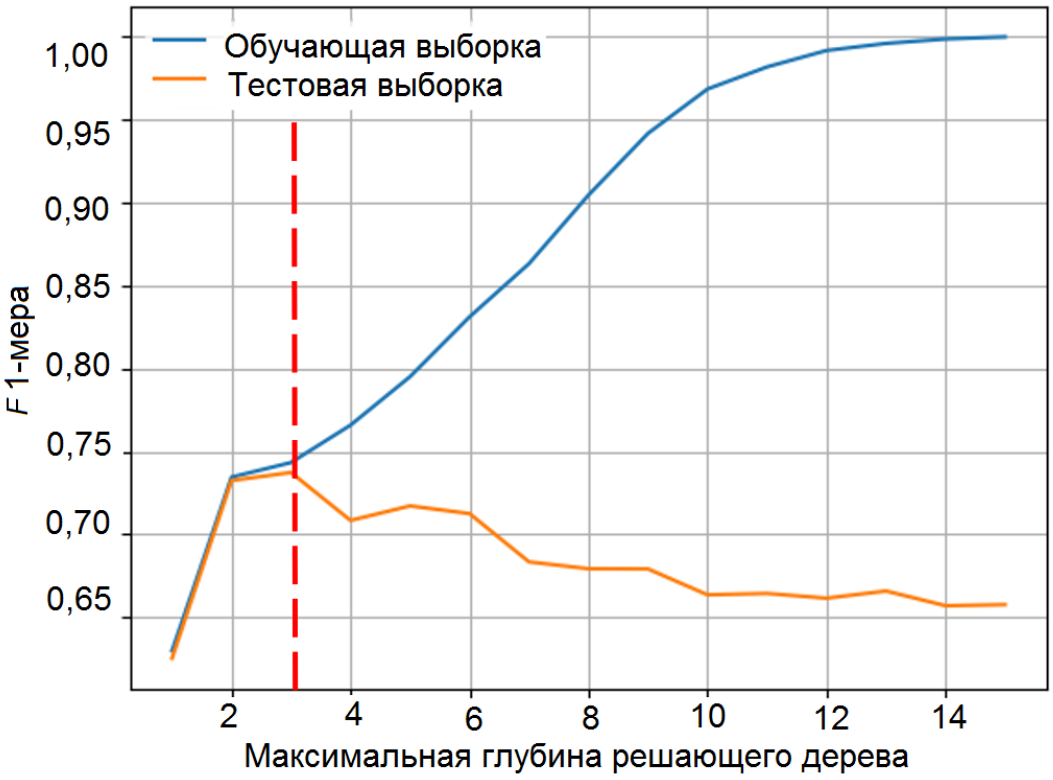

Рис. 8. Обучающие кривые от максимальной глубины дерева для *CART*

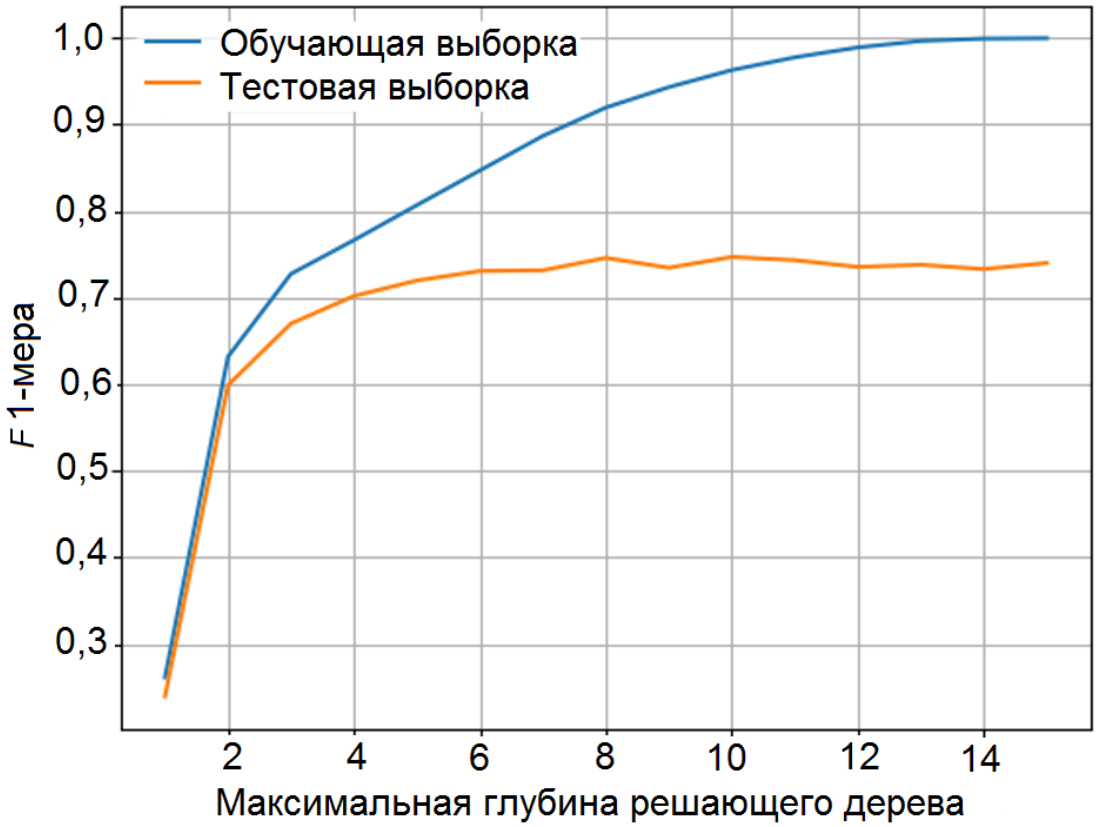

Рис. 9. Обучающие кривые от максимальной глубины для *Random Forest*

В качестве исходных данных использовались обезличенные сведения о состоянии здоровья, наследственности, условиях труда и поведенческих привычках работников металлургической промышленности. Требовалось выявить предикторы, оказывающие влияние на состояние сердечно-сосудистой системы и риск развития кардиологических патологий. Каждый метод ранжировал исследуемый список переменных по-своему, опираясь на информационную значимость признаков, оценку их вклада в прогностическую модель и анализ статистических взаимосвязей. Мы же отобрали только 10 наиболее значимых переменных.

Результаты применения методов добычи данных для каждого метода приведены на рис. 10–15. Перечень 10 наиболее значимых переменных приведен в табл. 3, где итоговый ранг важности находился как среднее результатов всех методов. Ввиду того, что самый первый метод (*CART*) включил в итоговые результаты только 6 переменных из общего списка, в таблицу для нахождения среднего ранга он не включен. Также в данном случае не стоит сравнивать напрямую значения рассчитанной для каждого предиктора неопределенности Джини, полученной в результате применения различных методов, т. к. полученные числа имеют условный характер и нужны только для ранжирования переменных. Если есть необходимость такого сопоставления, то можно перейти к безразмерной относительной важности переменных, изменяющейся в диапазоне от 0 до 1: каждой переменной сопоставляется отношение величины ее важности к максимальной важности переменных из этого набора. Стоит добавить, что само понятие важности переменных не формализовано и результаты такого анализа зависят от используемого метода [20]. Помимо этого, положение может осложниться наличием некоторого числа корреляций независимых переменных.

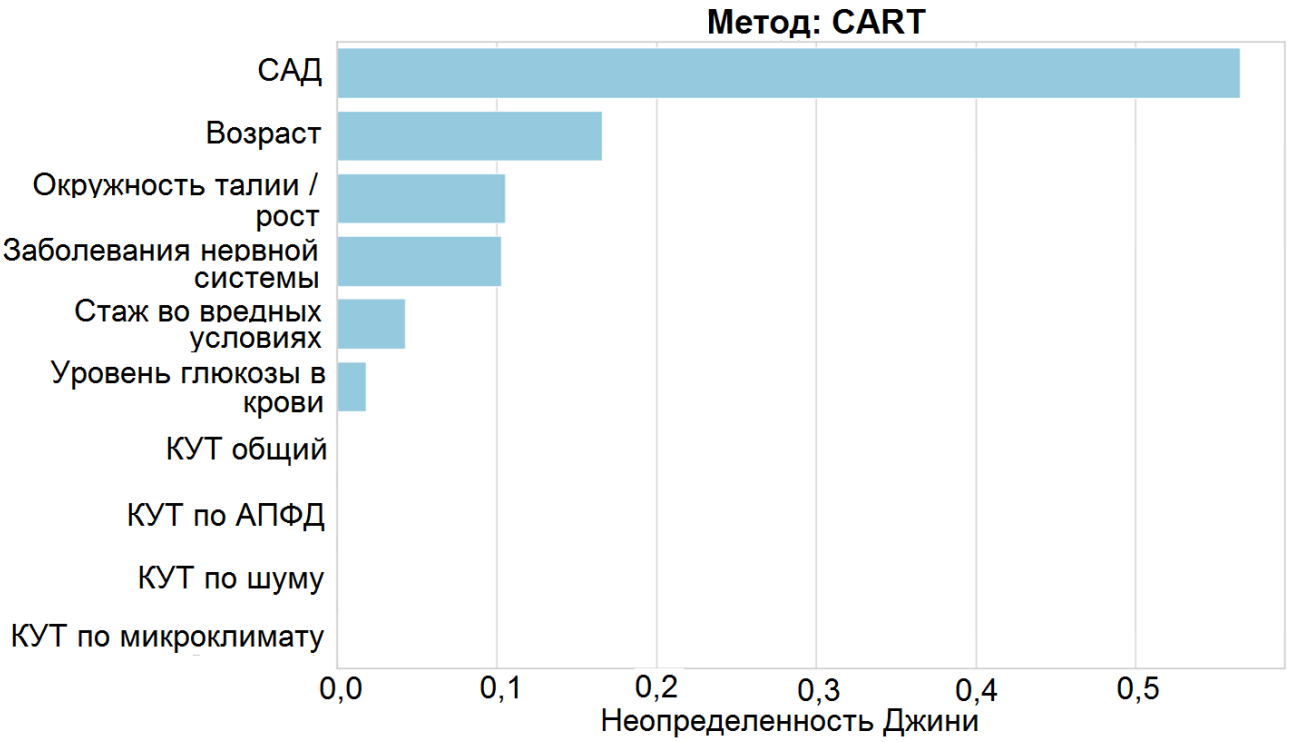

Рис. 10. Перечень наиболее важных переменных методом CART

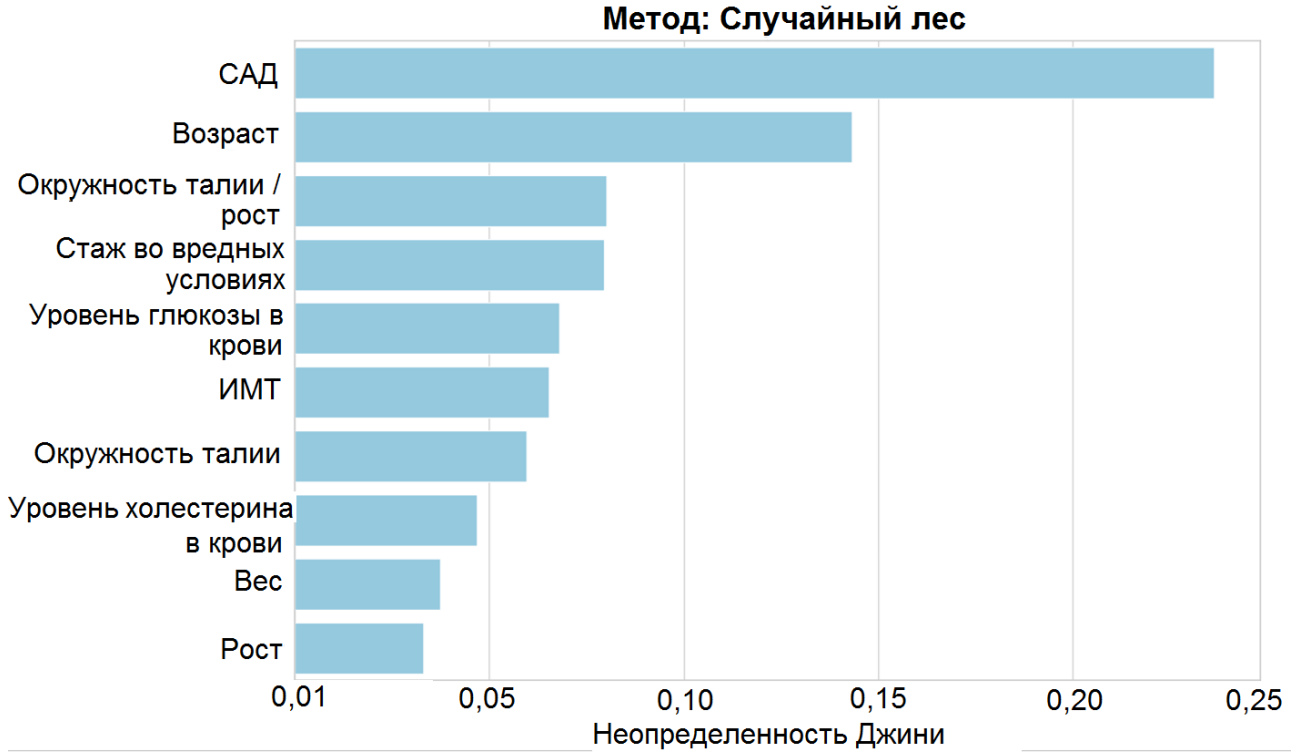

Рис. 11. Перечень наиболее важных переменных методом Случайный лес (Random Forest)

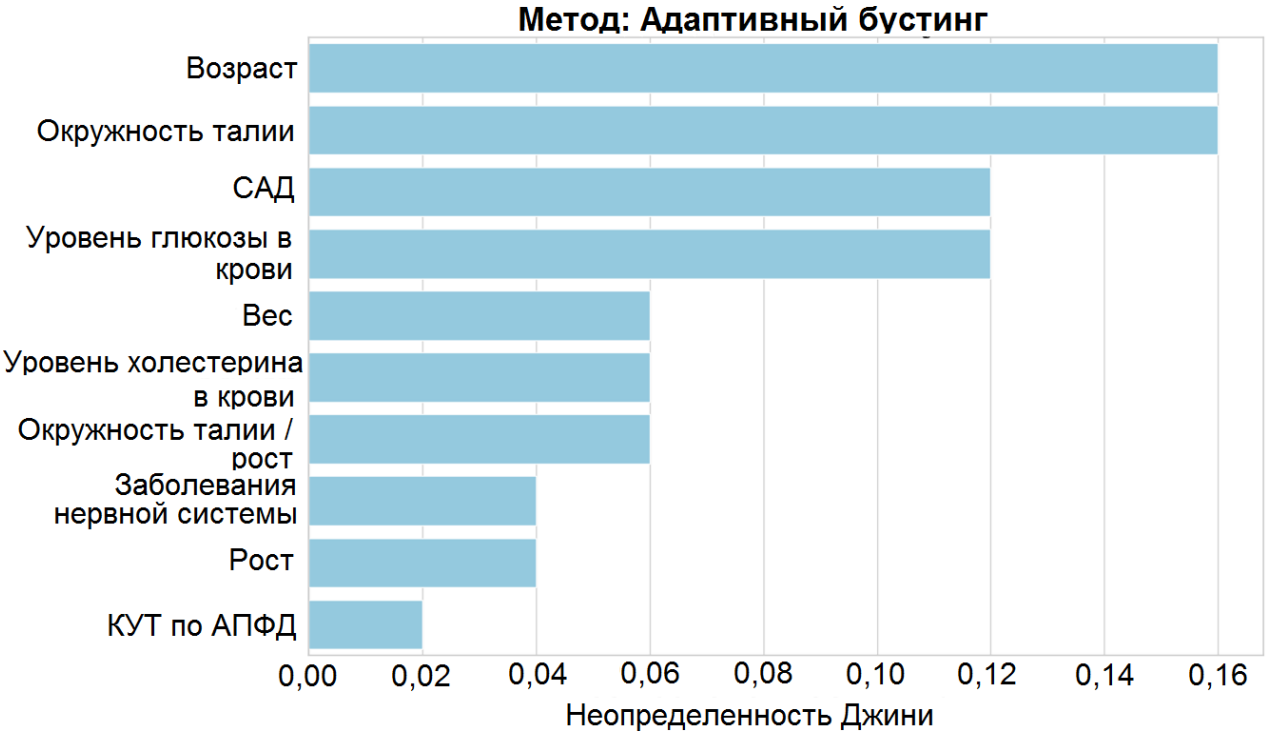

Рис. 12. Перечень наиболее важных переменных методом Адаптивный бустинг (AdaBoost)

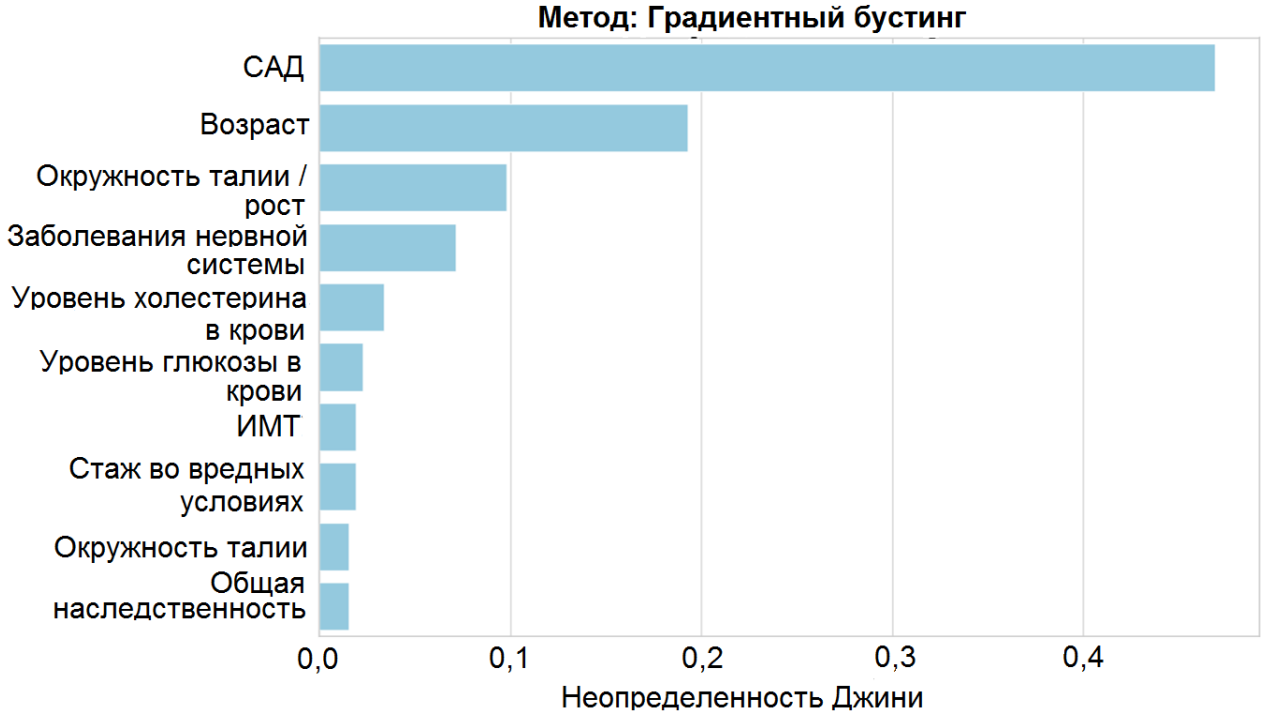

Рис. 13. Перечень наиболее важных переменных методом Градиентный бустинг (Gradient Boosting)

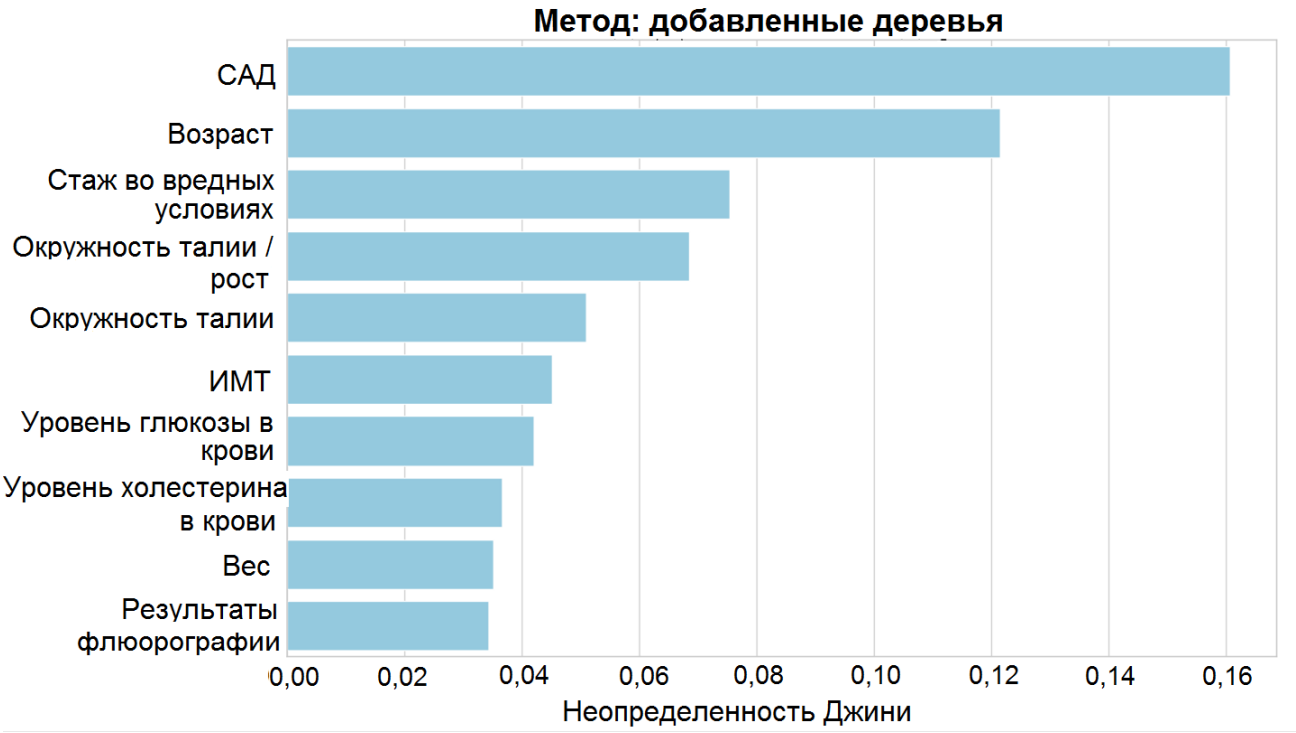

Рис. 14. Перечень наиболее важных переменных методом Добавленные деревья (Extra Trees)

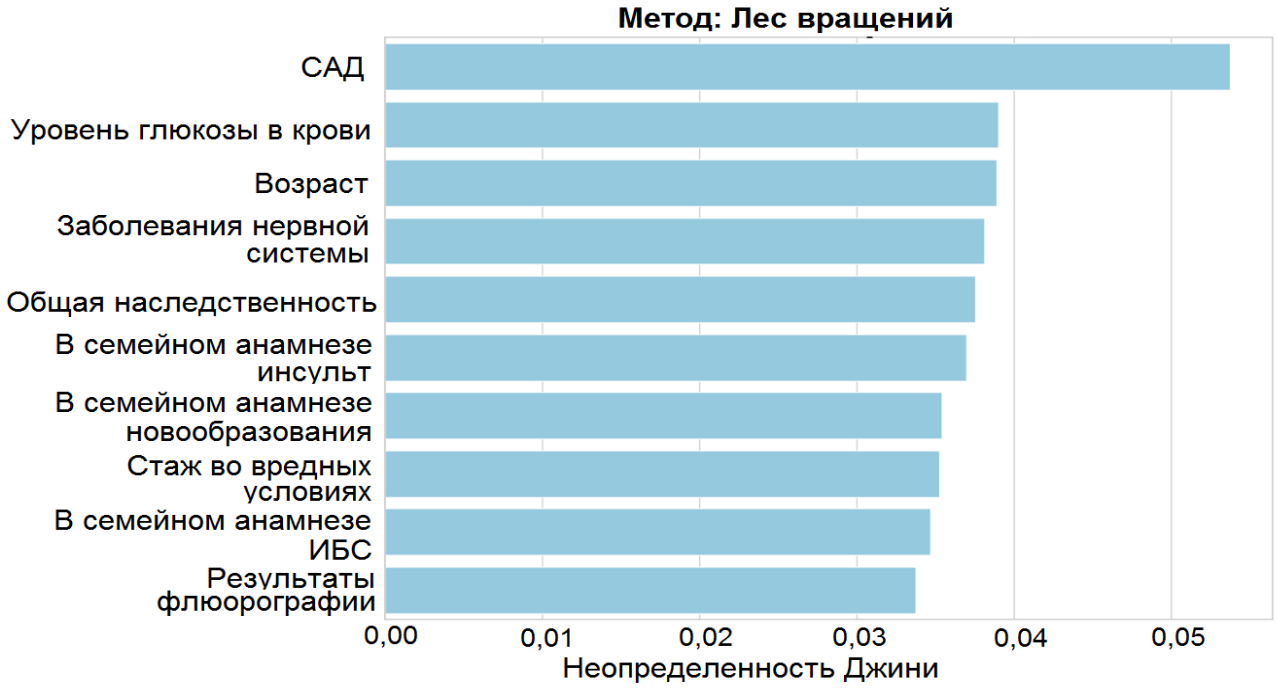

Рис. 15. Перечень наиболее важных переменных методом Лес вращений (Rotation Forest)

Таблица 3. Результаты применения методов добычи данных

Переменная	Random Forest	Ada Boost	Gradient Boosting	Extra Tree	Rotation Forest	Средний ранг
САД	1	3	1	1	1	1,40
Возраст	2	1	2	2	3	2,00
Уровень глюкозы в крови	5	4	6	7	2	4,80
Окружность талии / Рост	3	7	3	4	19	7,20
Стаж во вредных условиях	4	16	8	3	8	7,80

Окончание табл. 3

Переменная	Random Forest	Ada Boost	Gradient Boosting	Extra Tree	Rotation Forest	Средний ранг
Уровень холестерина в крови	8	6	5	8	15	8,40
ИМТ	6	12	7	6	13	8,80
Окружность талии	7	2	9	5	25	9,60
Заболевания нервной системы	12	8	4	22	4	10,00
Рост	10	9	13	11	16	11,80

Анализируя диаграмму на рис. 15, можно заметить, что алгоритм леса вращений демонстрирует низкие показатели неопределенности Джини. Этот результат обусловлен особенностями работы самого алгоритма. В частности, объекты разбиваются на случайные подмножества, для каждого из которых применяется метод главных компонент. Такой подход уменьшает вероятность доминирования одних и тех же объектов в различных деревьях. Также при большом количестве объектов значимость каждого отдельного элемента снижается, поскольку деревья могут иметь множество возможных вариантов разбиения.

Интересно отметить, что в табл. 3 в плане значимости составные переменные, такие как ИМТ, учитывающий рост и вес человека, и отношение окружности талии к росту, преобладают над переменными, входящими в их состав. Это можно объяснить тем, что составные переменные содержат больше информации и лучше отображают зависимости факторов, что способствует лучшему разбиению узлов деревьев решений [8, 21].

Стоит обратить внимание на переменные, имеющие значительный разброс ранговых значений среди различных методов. Это нельзя считать ошибкой, поскольку может служить отражением особенностей работы каждого конкретного алгоритма.

Для сравнения качества моделей были использованы показатели, описанные выше. Результаты проведенных расчетов приведены в табл. 4. Стоит отметить, что мы имеем дело с алгоритмами машинного обучения, поэтому при повторном обучении одних и тех же моделей с одними и теми же входными данными получаются различные результаты, хоть и близкие, а потому в таблице приведены усредненные значения метрик после проведения пяти испытаний программы.

Таблица 4. Оценка качества моделей

Параметр	CART	Random Forest	Ada Boost	Gradient Boosting	Extra Trees	Rotation Forest
Accuracy Score	0,819	0,813	0,798	0,819	0,804	0,839
Recall Score	0,671	0,601	0,594	0,587	0,585	0,664
Precision Score	0,774	0,806	0,766	0,840	0,792	0,833
*F*1 Score	0,719	0,689	0,669	0,691	0,673	0,739
Average Precision Score	0,633	0,622	0,595	0,636	0,607	0,669
Индекс Юдена	0,568	0,525	0,499	0,529	0,504	0,595
PSI	0,613	0,622	0,575	0,653	0,600	0,674
MCC	0,590	0,572	0,536	0,587	0,550	0,633

При анализе табл. 4 видно, что для нашей базы данных наиболее эффективным оказался новый классификатор на основе леса вращений. Он показал лучшие результаты по таким метрикам, как полнота модели, доля правильно классифицированных объектов и коэффициент корреляции Мэтьюса, благодаря тому, что на одном из этапов работы алгоритма применяется метод главных компонент, который уменьшает корреляцию между объектами и фильтрует шумы в данных. Это помогает снизить дисперсию модели и повысить ее стабильность. В результате при многократном обучении на одних и тех же данных алгоритм создает модели с устойчиво высокой производительностью, что положительно сказывается на

качестве метрик. Кроме того, «вращение» пространства увеличивает разнообразие деревьев в ансамбле и снижает риск переобучения, что также обеспечивает более стабильную работу.

Однако в методе леса вращений при повторных обучениях моделей на одних и тех же данных наблюдается существенное изменения перечня ранжированных по важности переменных. Это обусловлено малыми значениями неопределенности Джини, причина которых рассмотрена выше. Таким образом, данный метод представляется нам перспективным для построения классификационных моделей, в том числе и для медико-биологических задач. При выборе метода для выявления значимых переменных стоит отдавать предпочтение бустинговым алгоритмам, также показавшим хорошие результаты по метрикам точности. Эти алгоритмы при многократном обучении на одних и тех же данных (при наличии условия по ограничению глубины деревьев) стабильно выделяют одинаковые наборы значимых переменных, изредка отличающиеся порядком следования двух соседних переменных. Если требуется демонстрационный схематичный вариант прогностической модели, следует задействовать метод *CART*: его результаты отсутствуют в табл. 4, поскольку при ограничении глубины одиночное дерево выдает список переменных, соответствующий данному ограничению, что делает нецелесообразным его включение для определения среднего ранга всех переменных. Однако по точности классификации (см. табл. 3) данный алгоритм уступает только алгоритму леса вращений. Это объясняется тем, что, в отличие от ансамблей, тут исключаются все менее точные модели – остается одна модель с наилучшим разделением переменных.

Дополнительно мы оценили особенности построения моделей при различных количествах предикторов. Результаты приведены в табл. 5–7. Оказалось, что излишнее количество предикторов перегружает данные и приводит к ухудшению качества моделей, а потому для получения достоверных результатов рекомендуется проводить предварительную обработку данных классическими статистическими методами. Однако даже без очистки данных новый алгоритм Rotation Forest позволяет получить достаточно сильные классификационные модели.

Таблица 5. Оценка качества моделей с 3 предикторами

Параметр	CART	Random Forest	Ada Boost	Gradient Boosting	Extra Trees	Rotation Forest
Accuracy Score	0,800	0,810	0,788	0,802	0,764	0,737
Recall Score	0,629	0,594	0,587	0,587	0,336	0,587
Precision Score	0,750	0,802	0,743	0,785	0,941	0,627
*F*1 Score	0,684	0,683	0,656	0,672	0,495	0,607
Average Precision Score	0,560	0,616	0,579	0,603	0,545	0,510
Индекс Юдена	0,519	0,517	0,481	0,503	0,325	0,404
PSI	0,570	0,614	0,548	0,594	0,680	0,417
MCC	0,544	0,564	0,513	0,546	0,470	0,410

Таблица 6. Оценка качества моделей с 10 предикторами

Параметр	CART	Random Forest	Ada Boost	Gradient Boosting	Extra Trees	Rotation Forest
Accuracy Score	0,819	0,827	0,805	0,805	0,802	0,735
Recall Score	0,671	0,615	0,622	0,601	0,580	0,539
Precision Score	0,774	0,838	0,767	0,782	0,791	0,636
*F*1 Score	0,719	0,709	0,687	0,680	0,669	0,583

Окончание табл. 6

Параметр	CART	Random Forest	Ada Boost	Gradient Boosting	Extra Trees	Rotation Fores
Average Precision Score	0,633	0,648	0,608	0,608	0,603	0,602
Индекс Юдена	0,568	0,553	0,523	0,513	0,500	0,376
PSI	0,613	0,661	0,587	0,595	0,597	0,412
MCC	0,590	0,604	0,554	0,553	0,546	0,394

Таблица 7. Оценка качества моделей с 41 предиктором

Параметр	CART	Random Forest	Ada Boost	Gradient Boosting	Extra Trees	Rotation Forest
Accuracy Score	0,778	0,766	0,73	0,766	0,747	0,783
Recall Score	0,636	0,552	0,511	0,538	0,476	0,594
Precision Score	0,695	0,705	0,635	0,713	0,694	0,727
*F*1 Score	0,664	0,620	0,567	0,614	0,564	0,653
Average Precision Score	0,567	0,544	0,493	0,543	0,511	0,572
Индекс Юдена	0,489	0,431	0,356	0,424	0,365	0,477
PSI	0,512	0,494	0,401	0,498	0,457	0,532
MCC	0,500	0,462	0,378	0,460	0,409	0,504

Работу рассмотренных алгоритмов мы дополнительно проверили на других базах данных, созданных специально для отладки алгоритмов машинного обучения. Результаты оказались очень хорошими: в большинстве случаев значения точности прогностических моделей (*Accuracy Score*), полноты (*Recall Score*), коэффициента корреляции Мэтьюса (*MCC*), *F*1-меры превышают отметку в 0,90. Поскольку за основу текущей работы была взята реальная база данных со своими особенностями, среди которых несбалансированность классов, достичь отметки 0,9 практически невозможно, но и имеющиеся результаты можно оценить как очень хорошие.

Для подтверждения значимости полученных результатов оценили также размер эффекта классическими статистическими методами, описанными в первой части данного исследования. Сходство результатов, приведенных в табл. 3, 8, подтверждает возможность применения методов машинного обучения для выявления значимых переменных и их ранжирования по степени влияния на исследуемую целевую переменную.

Таблицы 8. Оценка размера эффекта классическими статистическими методами

Предиктор	*d* Коэна		*dCor*		$\varphi_к$		Средний ранг
	Значение	Ранг	Значение	Ранг	Значение	Ранг	
Возраст	1,007	2	0,426	2	0,361	1	1,67
САД	1,208	1	0,496	1	0,229	4	2,00
Стаж во вредных условиях	0,772	3	0,333	3	0,243	2	2,67
Окружность талии/рост	0,726	4	0,330	4	0,221	5	4,33
Окружность талии	0,680	5	0,311	5	0,216	6	5,33
Патологии на флюорографии	0,488	8	0,226	8	0,230	3	6,33

Окончание табл. 8

Предиктор	*d* Коэна		*dCor*		$\varphi_к$		Средний ранг
	Значение	Ранг	Значение	Ранг	Значение	Ранг	
ИМТ	0,571	6	0,258	7	0,170	7	6,67
Уровень глюкозы в крови	0,559	7	0,262	6	0,088	10	7,67
Вес	0,434	9	0,193	9	0,154	8	8,67
Уровень холестерина в крови	0,216	10	0,104	10	0,115	9	9,67

5. Адаптация алгоритмов для баз данных с несбалансированными классами

Изначально деревья решений, на которых основаны рассмотренные методы машинного обучения, были разработаны для решения проблем сбалансированной классификации, а потому они чувствительны к неравномерному распределению и не учитывают особенности проблем мультиклассового дисбаланса, поскольку нацелены на общую точность и могут игнорировать малочисленные классы. Однако в случае реальных баз данных такого добиться сложно. Так, в случае задействованной в данной работе медико-биологической базы данных число здоровых (т. е. без выявленных патологий БСК) существенно превышает количество людей с подтвержденными диагнозами. Это негативно сказывается на качестве получаемых классификационных моделей, т. к. они начинают лучше предсказывать доминирующий класс (здоровых) и хуже справляются с выявлением больных, а ведь именно они интересуют нас в первую очередь.

Используемый по умолчанию критерий разделения узлов деревьев – неопределенность Джини – весьма чувствителен к такому перекосу. При обучении на основе сильно несбалансированных наборов данных классовый дисбаланс препятствует их способности усвоить концепцию класса меньшинства. Решению данной проблемы посвящен ряд современных исследований. Один из вариантов – применение расстояния Хеллингера в качестве критерия разделения [22, 23]. Он был предложен Чеслаком в 2008 г. [24] и в общем случае определяет разницу между двумя распределениями вероятностей (27), (28). В случае деревьев решений он применяется для оценки различий в предсказательных распределениях двух моделей и для построения деревьев вместо неопределенности Джини.

$$D_H(P,Q) = \sqrt{\int_\Omega \left(\sqrt{p} - \sqrt{q}\right)^2 d\lambda} = \sqrt{2\left(1 - \int_\Omega \sqrt{pq}\, d\lambda\right)}, \quad (27)$$

где p, q – вероятности для некого класса в распределениях P, Q;

Ω – измеримое пространство;

λ – параметр пространства Ω;

$\int_\Omega \sqrt{pq}\, d\lambda$ – интеграл Хеллингера.

Более простой вариант записи:

$$D_H(P,Q) = \sqrt{\frac{1}{2}\sum_{i=1}^{k}\left(\sqrt{p_i} - \sqrt{q_i}\right)^2}, \quad (28)$$

где i – индекс класса в распределениях P, Q;

k – количество классов.

При разбиении дочернего узла дерева t_p на дочерние узлы t_l, t_r выбирается распределение, обеспечивающее максимизацию «выигрыша» (*Gain*) расстояния Хеллингера.

$$Gain = D_{H\_p} - \left(\frac{|S_l|}{|S|} D_{H\_l} + \frac{|S_r|}{|S|} D_{H\_r} \right), \quad (29)$$

где $D_{H\_p}, D_{H\_l}, D_{H\_r}$ – расстояния Хеллингера в родительском и левом, правом дочерних узлах;

S, S_l, S_r – множество данных в родительском и левом, правом дочерних узлах.

У описанного подхода есть несколько неоспоримых преимуществ: он более устойчив к выбросам и лучше работает при классовых дисбалансах, поскольку не склонен завышать вес преобладающего класса. Однако, как и для леса вращений, для него отсутствует готовый официальный пакет для языка программирования Python.

На практике мы использовали приближение к расстоянию Хеллингера для трех алгоритмов. Как можно видеть из табл. 9, данный способ оказывается рабочим и приводит к заметному улучшению качества получаемых классификационных моделей.

Таблица 9. Сравнение результатов применения неопределенности Джини и расстояния Хеллингера

Параметр	Неопределенность Джини			Расстояние Хеллингера		
	CART	Random Forest	Extra Tree	CART	Random Forest	Extra Tree
Accuracy Score	0,819	0,817	0,810	0,839	0,823	0,827
AUROC	0,817	0,890	0,897	0,831	0,890	0,895
Recall Score	0,671	0,622	0,594	0,643	0,622	0,629
Precision Score	0,774	0,802	0,820	0,852	0,840	0,826
*F*1 Score	0,719	0,701	0,698	0,733	0,715	0,714
Average Precision Score	0,718	0,830	0,824	0,751	0,830	0,828
Индекс Юдена	0,568	0,542	0,517	0,585	0,560	0,560
NND	1,759	1,847	1,934	1,711	1,786	1,787
PSI	0,613	0,624	0,614	0,686	0,665	0,653
MCC	0,590	0,581	0,564	0,633	0,610	0,604

Полученные результаты проще будет объяснить, если записать рассмотренные критерии через априорные («базовые») вероятности π_i, т. е. с учетом исходного распределения классов в данных до обучения модели (см. табл. 10).

Таблица 10. Сравнение критериев с учетом априорных вероятностей.

Критерий	Формула	Устойчивость к дисбалансу
Неопределенность Джини	$\sum_{i=1}^{k} p_i (1 - p_i) \pi_i$	Низкая
Энтропия	$-\sum_{i=1}^{k} p_i \log p_i \cdot \pi_i$	Средняя
Расстояние Хеллингера	$\left(\sqrt{p_1 \pi_1} - \sqrt{p_2 \pi_2} \right)^2$	Высокая

Что касается леса вращений, то для него существует еще один способ повышения качества итоговых моделей в случае классовой несбалансированности:

метод главных компонент, используемый при построении матрицы вращений, можно заменить на метод независимых компонент (*Independent Component Analysis,* или *ICA*) [22]. Применение этой модификации метода леса вращений мы планируем рассмотреть в последующих публикациях.

6. Заключение

В представленном исследовании проведен комплексный анализ методов машинного обучения для ранжирования переменных по их влиянию на целевую характеристику. Экспериментальные результаты приведены на основе анализа медико-биологических данных. Продемонстрирована применимость методов машинного обучения для построения прогностических моделей и выявления наиболее значимых признаков. Сопоставление результатов, полученных с использованием классических статистических методов и методов машинного обучения на одной базе данных, демонстрирует значительное сходство между двумя перечнями наиболее значимых переменных. Это подтверждает допустимость применения моделей типа «черный ящик» для анализа табличных данных и перспективность дальнейших исследований в данном направлении.

Дополнительно представлен сравнительный анализ задействованных в работе методов машинного обучения, который позволил уточнить некоторые особенности их применения. Так, например, если цель заключается в построении качественной и точной прогностической модели, предпочтение стоит отдать лесу вращений. Градиентный бустинг показал наибольшую эффективность в ранжировании значимости переменных. Тем не менее, учитывая активное развитие всех этих подходов, невозможно утверждать, что один из них значительно превосходит другой, поскольку итоговый результат зависит от специфики решаемой задачи.

Также упомянуты варианты адаптации алгоритмов для анализа баз данных с несбалансированными классами. Исследования в данном направлении представляют собой перспективные работы с практической значимостью, поскольку в большинстве реальных баз данных присутствует перекос по численности наблюдений в каком-либо направлении.

Исследование выполнено за счет субсидий Минобрнауки РФ на выполнение научных тем FUMN-2024-0002, FUMN-2024-0003.

7. Список источников

1. Применение методов машинного обучения для анализа изображений хронических ран / А. Г. Назаренко, Е. Б. Клейменова, А. И. Молодченков [и др.] // Искусственный интеллект и принятие решений. – 2025. – № 1. – С. 103–114. – DOI 10.14357/20718594250109. – EDN NDTFCY.
2. Detection and analysis of COVID-19 in medical images using deep learning techniques / D. Yang, C. Martinez, L. Visuña [et al.] // Scientific Reports. – 2021. – Vol. 11, No. 1. – DOI 10.1038/s41598-021-99015-3. – EDN NZZOFL.
3. *Varoquaux, G*. Machine learning for medical imaging: methodological failures and recommendations for the future / G. Varoquaux, V. Cheplygina // npj Digital Medicine. – 2022. – Vol. 5, No. 1. – P. 1–8. – DOI 10.1038/s41746-022-00592-y. – EDN SIKHQX.
4. *Мифтахова, А. А.* Применение метода дерева решений для решения задач классификации и прогнозирования / А. А. Мифтахова // Инфокоммуникационные технологии. – 2016. – Т. 14, № 1. – С. 64–70. – DOI 10.18469/ikt.2016.14.1.10. – EDN WDCYVR.
5. *Breiman, L.* Classification and regression trees / L. Breiman. – New York: Chapman and Hall/CRC, 2017. – 368 p. – ISBN 9781315139470.
6. Using AdaBoost to improve the performance of simple classifiers / A. A. Mahkamov, T. S. Jumayev, D. S. Tuhtanazarov, A. I. Dadamuxamedov // Artificial Intelligence, Blockchain, Computing and Security. – 2024. – Vol. 2. – P. 755–760. – DOI 10.1201/9781032684994-122.

7. High-Dimensional Data Bootstrap / V. Chernozhukov, D. Chetverikov, K. Kato, Yu. Koike // Annual Review of Statistics and Its Application. – 2023. – Vol. 10, No. 1. – P. 427–449. – DOI 10.1146/annurev-statistics-040120-022239. – EDN IGPGTC.
8. *Schapire, R. E*. The boosting approach to machine learning: An overview / R. E. Schapire // Nonlinear estimation and classification. – 2003. – P. 149–171. – DOI 10.1007/978-0-387-21579-2_9.
9. *Natekin, A*. Gradient boosting machines, a tutorial / A. Natekin, A. Knoll // Frontiers in neurorobotics. – 2013. – Vol. 7. – P. 1–21. – DOI 10.3389/fnbot.2013.00021.
10. *Breiman, L*. Random forests / L. Breiman // Machine learning. – 2001. – Vol. 45. – P. 5–32. – DOI 10.1023/A:1010933404324.
11. *Hussein, N.* Performance Evaluation of Extra Trees Classifier by using CPU Parallel and Non-Parallel Processing / N. Hussein, S. R. M. Zeebaree // Indonesian Journal of Computer Science. – 2024. – Vol. 13, No. 2. – P. 1859–1872. – DOI 10.33022/ijcs.v13i2.3802. – EDN XQXVTX.
12. *Geurts, P*. Extremely randomized trees / P. Geurts, D. Ernst, L. Wehenkel // Machine learning. – 2006. – Vol. 63. – P. 3–42. – DOI 10.1007/s10994-006-6226-1.
13. *Rodriguez, J. J.* Rotation Forest: A new classifier ensemble method / J. J. Rodriguez, L. I. Kuncheva, C. J. Alonso // IEEE transactions on pattern analysis and machine intelligence. – 2006. – Vol. 28, No. 10. – P. 1619–1630. – DOI 10.1109/TPAMI.2006.211.
14. *Dritsas, E.* Efficient Data-Driven Machine Learning Models for Cardiovascular Diseases Risk Prediction / E. Dritsas, M. Trigka // Sensors. – 2023. – Vol. 23, No. 3. – P. 1161. – DOI 10.3390/s23031161. – EDN NLAEDR.
15. *Sheikhi, S.* An Efficient Rotation Forest-Based Ensemble Approach for Predicting Severity of Parkinson's Disease / S. Sheikhi, M. T. Kheirabadi //Journal of healthcare engineering. – 2022. – No. 1. – P. 1–9. – DOI 10.1155/2022/5524852.
16. *Dritsas, E.* Machine Learning Techniques for Chronic Kidney Disease Risk Prediction / E. Dritsas, M. Trigka // Big Data and Cognitive Computing. – 2022. – Vol. 6, No. 3. – P. 98. – DOI 10.3390/bdcc6030098. – EDN AWJIWY.
17. The impact of class imbalance in classification performance metrics based on the binary confusion matrix / A. Lique, A, Carrasco, A. Martin, A. I. Heras // Pattern Recognition. – 2019. – Vol. 91. – P. 216–231. – DOI 10.1016/j.patcog.2019.02.023.
18. *Chicco, D*. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation / D. Chicco, G. Jurman // BMC Genomics. – 2020. – Vol. 21, No. 1. – P. 6. – DOI 10.1186/s12864-019-6413-7. – EDN OVTHSE.
19. *Viering, T*. The Shape of Learning Curves: A Review / T. Viering, M. Loog // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2023. – Vol. 45, No. 6. – P. 7799-7819. – DOI 10.1109/tpami.2022.3220744. – EDN XOOXUU.
20. *Dong, J*. Exploring the cloud of variable importance for the set of all good models / J. Dong, C. Rudin // Nature Machine Intelligence. – 2020. – Vol. 2, No. 12. – P. 810–824. – DOI 10.1038/s42256-020-00264-0. – EDN NEBRGG.
21. *Wei, P*. Variable importance analysis: A comprehensive review / P. Wei, Z. Lu, J. Song // Reliability Engineering & System Safety. – 2015. – Vol. 142. – P. 399–432. – DOI 10.1016/j.ress.2015.05.018. – EDN UQHSLR.
22. Improving random forest and rotation forest for highly imbalanced datasets / S. Chong, J. Shenggen, L. Yiguang, Y. Zhonghua // Intelligent Data Analysis. – 2015. – Vol. 19, No. 6. – P. 1409–1432. – DOI 10.3233/IDA-150789.
23. *Ortega Vázquez, C*. Hellinger distance decision trees for PU learning in imbalanced data sets / C. Ortega Vázquez, S. Vanden Broucke, J. De Weerdt // Machine Learning. – 2023. – DOI 10.1007/s10994-023-06323-y. – EDN ELADXZ.
24. *Cieslak, D. A.* Learning decision trees for unbalanced data / D. A. Cieslak, N. V. Chawla // Machine Learning and Knowledge Discovery in Databases: European Conference. – 2008. – P. 241–256.

Сведения об авторах:

Кабакова Елизавета Александровна, инженер Института промышленной экологии УрО РАН, г. Екатеринбург, Россия. Эл. почта: Liza.Kabakowa@yandex.ru

Панов Владимир Григорьевич, к. ф.-м. н., ведущий научный сотрудник Института промышленной экологии УрО РАН, г. Екатеринбург, Россия. Эл. почта: vpanov@ecko.uran.ru

SOME METHODS FOR ESTIMATING THE IMPORTANCE OF VARIABLES IN STATISTICS. II. MACHINE LEARNING METHODS

E. A. Kabakova, V. G. Panov

Institute of Industrial Ecology, Ural Branch, Russian Academy of Sciences, Ekaterinburg, Russia

Some machine learning (ML) methods used for variable importance estimation are considered. The paper provides an overview of several tree-based classification methods (CART, AdaBoost, Gradient Boosted Trees, Random Forest, Extra Trees, Rotation Forest) and compares peculiarities of their application to biomedical data analysis. Additionally, possible adaptations of the algorithms for processing datasets with imbalanced classes are proposed.

Key words: risk factors; mathematical statistics methods; machine learning; data mining methods; classification trees; importance of variables.

References

1. Application of machine learning methods for image analysis of chronic wounds / A. G. Nazarenko, E. B. Kleimenova, A. I. Molodchenkov [et al.] // Artificial intelligence and decision-making. – 2025. – No. 1. – P. 103–114. – DOI 10.14357/20718594250109. – EDN NDTFCY. (In Russian).
2. Detection and analysis of COVID-19 in medical images using deep learning techniques / D. Yang, C. Martinez, L. Visuña [et al.] // Scientific Reports. – 2021. – Vol. 11, No. 1. – DOI 10.1038/s41598-021-99015-3. – EDN NZZOFL.
3. *Varoquaux, G*. Machine learning for medical imaging: methodological failures and recommendations for the future / G. Varoquaux, V. Cheplygina // npj Digital Medicine. – 2022. – Vol. 5, No. 1. – P. 1–8. – DOI 10.1038/s41746-022-00592-y. – EDN SIKHQX.
4. *Miftahova, A. A.* Application of the decision tree method for solving classification and forecasting problems / A. A. Miftahova // Infocommunication technologies. – 2016. – Vol. 14, No. 1. – P. 64–70. – DOI 10.18469/ikt.2016.14.1.10. – EDN WDCYVR. (In Russian).
5. *Breiman, L.* Classification and regression trees / L. Breiman. – New York: Chapman and Hall/CRC, 2017. – 368 p. – ISBN 9781315139470.
6. Using AdaBoost to improve the performance of simple classifiers / A. A. Mahkamov, T. S. Jumayev, D. S. Tuhtanazarov, A. I. Dadamuxamedov // Artificial Intelligence, Blockchain, Computing and Security. – 2024. – Vol. 2. – P. 755–760. – DOI 10.1201/9781032684994-122.
7. High-Dimensional Data Bootstrap / V. Chernozhukov, D. Chetverikov, K. Kato, Yu. Koike // Annual Review of Statistics and Its Application. – 2023. – Vol. 10, No. 1. – P. 427–449. – DOI 10.1146/annurev-statistics-040120-022239. – EDN IGPGTC.
8. *Schapire, R. E*. The boosting approach to machine learning: An overview / R. E. Schapire // Nonlinear estimation and classification. – 2003. – P. 149–171. – DOI 10.1007/978-0-387-21579-2_9.
9. *Natekin, A*. Gradient boosting machines, a tutorial / A. Natekin, A. Knoll // Frontiers in neurorobotics. – 2013. – Vol. 7. – P. 1–21. – DOI 10.3389/fnbot.2013.00021.
10. *Breiman, L*. Random forests / L. Breiman // Machine learning. – 2001. – Vol. 45. – P. 5–32. – DOI 10.1023/A:1010933404324.
11. *Hussein, N.* Performance Evaluation of Extra Trees Classifier by using CPU Parallel and Non-Parallel Processing / N. Hussein, S. R. M. Zeebaree // Indonesian Journal of Computer Science. – 2024. – Vol. 13, No. 2. – P. 1859–1872. – DOI 10.33022/ijcs.v13i2.3802. – EDN XQXVTX.
12. *Geurts, P*. Extremely randomized trees / P. Geurts, D. Ernst, L. Wehenkel // Machine learning. – 2006. – Vol. 63. – P. 3–42. – DOI 10.1007/s10994-006-6226-1.
13. *Rodriguez, J. J.* Rotation Forest: A new classifier ensemble method / J. J. Rodriguez, L. I. Kuncheva, C. J. Alonso // IEEE transactions on pattern analysis and machine intelligence. – 2006. – Vol. 28, No. 10. – P. 1619–1630. – DOI 10.1109/TPAMI.2006.211.
14. *Dritsas, E.* Efficient Data-Driven Machine Learning Models for Cardiovascular Diseases Risk Prediction / E. Dritsas, M. Trigka // Sensors. – 2023. – Vol. 23, No. 3. – P. 1161. – DOI 10.3390/s23031161. – EDN NLAEDR.

15. *Sheikhi, S.* An Efficient Rotation Forest-Based Ensemble Approach for Predicting Severity of Parkinson's Disease / S. Sheikhi, M. T. Kheirabadi //Journal of healthcare engineering. – 2022. – No. 1. – P. 1–9. – DOI 10.1155/2022/5524852.
16. *Dritsas, E.* Machine Learning Techniques for Chronic Kidney Disease Risk Prediction / E. Dritsas, M. Trigka // Big Data and Cognitive Computing. – 2022. – Vol. 6, No. 3. – P. 98. – DOI 10.3390/bdcc6030098. – EDN AWJIWY.
17. The impact of class imbalance in classification performance metrics based on the binary confusion matrix / A. Lique, A, Carrasco, A. Martin, A. I. Heras // Pattern Recognition. – 2019. – Vol. 91. – P. 216–231. – DOI 10.1016/j.patcog.2019.02.023.
18. *Chicco, D*. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation / D. Chicco, G. Jurman // BMC Genomics. – 2020. – Vol. 21, No. 1. – P. 6. – DOI 10.1186/s12864-019-6413-7. – EDN OVTHSE.
19. *Viering, T*. The Shape of Learning Curves: A Review / T. Viering, M. Loog // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2023. – Vol. 45, No. 6. – P. 7799-7819. – DOI 10.1109/tpami.2022.3220744. – EDN XOOXUU.
20. *Dong, J*. Exploring the cloud of variable importance for the set of all good models / J. Dong, C. Rudin // Nature Machine Intelligence. – 2020. – Vol. 2, No. 12. – P. 810–824. – DOI 10.1038/s42256-020-00264-0. – EDN NEBRGG.
21. *Wei, P*. Variable importance analysis: A comprehensive review / P. Wei, Z. Lu, J. Song // Reliability Engineering & System Safety. – 2015. – Vol. 142. – P. 399–432. – DOI 10.1016/j.ress.2015.05.018. – EDN UQHSLR.
22. Improving random forest and rotation forest for highly imbalanced datasets / S. Chong, J. Shenggen, L. Yiguang, Y. Zhonghua // Intelligent Data Analysis. – 2015. – Vol. 19, No. 6. – P. 1409–1432. – DOI 10.3233/IDA-150789.
23. *Ortega Vázquez, C*. Hellinger distance decision trees for PU learning in imbalanced data sets / C. Ortega Vázquez, S. Vanden Broucke, J. De Weerdt // Machine Learning. – 2023. – DOI 10.1007/s10994-023-06323-y. – EDN ELADXZ.
24. *Cieslak, D. A.* Learning decision trees for unbalanced data / D. A. Cieslak, N. V. Chawla // Machine Learning and Knowledge Discovery in Databases: European Conference. – 2008. – P. 241–256.