DOI: 10.18720/SPBPU/2/id20-63

3. ENSEMBLE LEARNING METHOD BASED ON ORACLE DATA MINING FOR CARDIOVASCULAR DISEASES PREDICTION

***Aleksandra Dmitrievna Soboleva**, student, Peter the Great St. Petersburg Polytechnic University, Russia, St.Petersburg, Polytechnicheskaya str., house 29, BOX 195251, soboleva2.ad@edu.spbstu.ru.*

***Oleg Yurievich Sabinin**, cand. sci. (engineering), associate professor, Peter the Great St. Petersburg Polytechnic University, Russia, St.Petersburg, Polytechnicheskaya str., house 29, BOX 195251, olegsabinin@mail.ru.*

Annotation. *The prediction problem, the existing machine learning methods and related issues are considered in the chapter. The modification of the bagging method, which aggregates two fundamentally different machine learning algorithms (Generalized Liner Model and Support Vector Machine), is proposed and justified. The research of the method based on three different heart disease datasets. It confirms the effectiveness of the method for solving problem of risk estimation of the cardiovascular disease.*

Keywords. *Machine learning, prediction problem, classification, regression, Oracle, data mining, cardiovascular diseases.*

МЕТОД КОМПОЗИЦИИ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ НА ОСНОВЕ ORACLE DATA MINING ДЛЯ ПРОГНОЗИРОВАНИЯ СЕРДЕЧНО-СОСУДИСТЫХ ЗАБОЛЕВАНИЙ

***Александра Дмитриевна Соболева**, студент, Санкт-Петербургский политехнический университет Петра Великого, Россия, Санкт-Петербург, ул. Политехническая, дом 29, индекс 195251, soboleva2.ad@edu.spbstu.ru.*

***Олег Юрьевич Сабинин**, кандидат технических наук, доцент, Санкт-Петербургский политехнический университет Петра Великого, Россия, Санкт-Петербург, ул. Политехническая, дом 29, индекс 195251, olegsabinin@mail.ru.*

Аннотация. *В главе рассмотрены существующие методы машинного обучения, решающие задачу прогнозирования, и обозначены их недостатки. Предложен и обоснован метод, решающий задачу прогнозирования посредством агрегирования результатов двух алгоритмов машинного обучения (Обобщенная Линейная Модель и Машина опорных векторов), противоположных по природе. Рассмотрена проблема сердечно-сосудистых заболеваний и выдвинута гипотеза о применимости данного метода для ранней диагностики этих заболеваний. Проведено исследование метода на примере трех различных наборов данных анализов пациентов и их анамнеза для оценки рисков кардиологических заболеваний, подтвердившее эффективность разработанного подхода для решения задачи ранней диагностики сердечно-сосудистых заболеваний.*

Ключевые слова. *Машинное обучение, задача прогнозирования, классификация, регрессия, Oracle, интеллектуальный анализ данных, сердечно-сосудистые заболевания.*

Введение

В настоящее время компьютерные технологии заняли одно из главных мест, как в повседневной жизни человека, так и в бизнесе. Благодаря быстрому развитию и совершенствованию аппаратной части и программного обеспечения цифровых устройств за пару десятков лет заметно снизилась стоимость вычислительных ресурсов, в том числе и параллельных, оперативной и постоянной памяти. Все это привело к накоплению больших объемов разнородных данных, исчисляющихся терабайтами, которые обозначаются термином большие данные. Такое количество информации человек не способен обработать вручную, кроме того, и традиционные статистические и аналитические методы плохо справляются с задачами анализа и обработки больших данных [3.20]. В связи с этим появилась потребность разрабатывать новые методы и алгоритмы обработки данных, которые способны эффективно анализировать большие данные при применении машинного обучения.

Существует множество различных алгоритмов машинного обучения, среди которых выделяются несколько главных групп по свойствам решаемых ими задач. При применении алгоритма машинного обучения для решения конкретной задачи в большинстве случаев требуются дополнительные эври-

стики, эксперименты и различные модели для выявления зависимостей и закономерностей, свойственных имеющимся данным и предположениям.

Одной из важнейших задач машинного обучения является задача прогнозирования, которая заключается в предсказании значений некоторых параметров или свойств исследуемой системы на основе наблюдений и анализа известных параметров и поведения системы в прошлом и настоящем. Под системой в данном случае подразумевается комплекс природных законов или бизнес-правил, объектов, на которые эти правила оказывают влияние, взаимосвязей между ними, который направлен на достижение определенных целей. Например, в медицине необходимо предсказывать риски заболеваний по анализам пациента, в экономике – поведение рынка как ценных бумаг, так и потребительского. Появляется возможность, например, создавать персонализированную рекламу, в банковской сфере – оценивать кредитные риски, в сфере безопасности – выявлять мошенников и преступников, в повседневной жизни – экономить время и беречь здоровье, например, за счет предсказания времени прибытия общественного транспорта.

3.1. Цель работы

Для прогнозирования каких-либо значений, будь то курс криптовалюты в следующем месяце, вероятность развития сердечного заболевания через несколько лет или же состав потребительской корзины, требуется выявить закономерность по существующим данным, которые были собраны в прошлом и настоящем.

Существует несколько алгоритмов машинного обучения, решающих данную задачу, каждый из которых ищет закономерность, основываясь на теоремах математической статистики, теории вероятностей, дискретной математики или теории графов [3.16]. Кроме того, существующие алгоритмы объединяются в композиции для получения более точной модели предсказания [3.8; 3.19]. Но все же остается проблема выбора алгоритма для решения задачи прогнозирования.

Целью данной работы является разработка метода композиции алгоритмов машинного обучения для решения задачи прогнозирования сердечно-

сосудистых заболеваний на основе технологии Oracle Data Mining. Метод композиции позволяет взаимно компенсировать ошибки отдельных алгоритмов.

3.2. Алгоритмы машинного обучения для задачи прогнозирования

Рассмотрим существующие подходы и методы решения задачи прогнозирования.

К прогностическому обучению, которое также называют обучением с учителем, относят классические задачи интеллектуального анализа данных, такие как классификация и регрессия [3.40]. При обучении прогнозирующей модели на вход подается тренировочный набор данных с указанием для каждой записи в данных значения прогнозируемого параметра. По тренировочному набору алгоритмы обучения аппроксимируют функцию прогнозирования в том или ином заданном классе функций (регрессия, логическое правило, разделяющая гиперповерхность в пространстве признаков).

3.2.1. Регрессия и классификация

Регрессия одновременно очень схожа и отличается от классификации. Главное отличие регрессии в том, что она позволяет обрабатывать как дискретные, так и непрерывные величины, что является значительным достоинством, поскольку большая часть измерений в реальном мире описывается законами или функциями [3.31]. Второе принципиальное отличие регрессии от классификации состоит в том, что результатом классификации является идентификатор класса или вероятность принадлежности элемента к некоторому классу, а регрессии – определенное значение, выбранного для прогнозирования параметра.

Сформулируем математическую постановку задачи классификации в терминах теории вероятностей. Предполагаем, что множество пар «объект, класс» $X \times Y$ является вероятностным пространством с неизвестной вероятностной мерой P. Имеется конечная обучающая выборка наблюдений $X^m = (x_1, y_1), ..., (x_m, y_m)$, сгенерированная согласно вероятностной мере P.

Требуется построить алгоритм

$$a : X \to Y, \tag{3.1}$$

способный классифицировать произвольный объект $x \in X$ [3.6].

Задачу классификации решает большой круг алгоритмов машинного обучения. К нему относят:

- байесовский классификатор [3.36];
- линейный классификатор [3.25];
- решающие деревья [3.38];
- решающие списки [3.39];
- логистическая регрессия [3.37];
- метод опорных векторов [3.7];
- модификации вышеперечисленных алгоритмов.

Сформулируем задачу регрессии. Пусть имеется множество $\{x_1, \ldots, x_n \mid x \in \mathbb{R}^m\}$ n объектов с m признаками, такие объекты называются предикторами, и множество $\{y_1, \ldots, y_n \mid y \in \mathbb{R}\}$ соответствующих им значений функции, которые называются откликами. Требуется найти вектор констант w такой, что бы удовлетворялась формула (3.2) [3.43].

$$y \approx wx. \tag{3.2}$$

Поскольку задача регрессии очень схожа с задачей классификации, алгоритмы, решающие эти две задачи, также схожи. Задачу регрессии решают:

- линейная регрессия;
- нелинейная регрессия [3.23];
- метод опорных векторов.

Таким образом, с помощью решения задачи классификации мы сможем спрогнозировать динамику некоторого параметра, то есть предсказывать уменьшился он или увеличился с помощью введения дополнительного бинарного атрибута. Например, таким образом возможно спрогнозировать динамику роста или спада цены на фондовом рынке [3.36]. Кроме того, с помощью решения задачи классификации можно оценить риск появления заболевания у пациента через месяц, а с помощью решения задачи регрессии можно предсказать непосредственно значение параметра, например, цену на фондовом рынке или уровень сахара в крови у пациента.

3.2.2. Проблемы существующих методов

Выбор алгоритма машинного обучения зависит от требований и условий поставленной задачи. Точность решения задачи классификации и регрессии очень чувствительна к данным. Таким образом, при сильно зашумленных данных, или при малой и не представительной обучающей выборке, которая содержит свойства, не распространяющиеся на все множество, для которого производится прогноз, невозможно получить качественный результат [3.26].

Существует также проблема эффекта переобучения, которая заключается в том, что на тренировочном наборе данных получается модель высокой точности, а на реальных данных параметры модели не удовлетворяют поставленной задаче [3.36]. В связи с этим требуется очень тщательно подбирать данные для обучения и тестирования, чтобы экземпляры тренировочного набора отвечали представительности. Представительной считается такая обучающая выборка, которая в заданном пространстве признаков и заданном классе решающих функций позволяет построить правило прогнозирования новых объектов (тестовой выборки) с ошибкой, не превышающей заданной величины [3.47]. При решении практической задачи на этапе обучения, как правило, нет точной предварительной информации об объектах, которые будут предсказаны, таким образом обучающая выборка должна быть достаточной, то есть количество объектов в выборке должно соответствовать требованиям алгоритма, в выборку должно попадать большое число различных комбинаций входных атрибутов и выходного прогноза, а также она должна быть сбалансированной, то есть объекты различных классов должны быть равномерно распределены в выборке во избежание доминирования одного класса над остальными [3.14].

Поскольку не всегда есть возможность проводить трудоемкий анализ данных для выявления их свойств и класса зависимостей, в них имеющихся, то следует выбирать несколько алгоритмов машинного обучения, реализующих разные принципы обработки данных.

3.3. Метод композиции алгоритмов машинного обучения

Метод композиции алгоритмов является мощным средством увеличения точности прогнозирования по сравнению с использованием только одного алгоритма [3.32]. Возьмем несколько алгоритмов, которые решают как задачу классификации, так и регрессии, и имеют различную природу. Построим над этими алгоритмами композицию. Такие алгоритмы называются базовыми алгоритмами композиции. Результатом композиции будут агрегированные по некоторому правилу, например, взвешенному голосованию, результаты нескольких базовых алгоритмов, построенных на основе различных подмножеств тренировочной выборки.

Заметим, что, чем меньше алгоритмов участвуют в композиции, тем меньше используется вычислительных ресурсов и памяти. Кроме того, при использовании такого подхода обучение каждого алгоритма становится независимым от обучения остальных, что приводит к минимизации накладных расходов при распараллеливании и возможности распределения большого количества данных между разными вычислительными устройствами.

Метод агрегирования результатов нескольких алгоритмов рекомендуется использовать с нечетным количеством базовых алгоритмов, что связано с проблемой состояния неопределенности, в случае, когда результаты базовых алгоритмов отличаются друг от друга. Однако, если произвести некоторую модификацию правила выбора результирующего прогноза, то можно построить композицию над двумя алгоритмами. Модификация заключается в использовании вероятности отнесения данного объекта к выбранному классу. То есть результирующим является тот ответ, у которого вероятность больше. В случае, если вероятности равны, выберем результат первой базовой модели. Из-за того, что, значения вероятностей принадлежат бесконечному множеству действительных чисел в интервале от 0 до 1, в отличие от конечного множества результатов классификации, то вероятность получения ситуации неопределенности стремится к 0. При решении задачи регрессии будем выбирать среднее арифметическое значение из двух результатов.

3.4. Базовые алгоритмы композиции

Предположим, что алгоритмы, которые ищут различные формы зависимости между классовым атрибутом объекта и остальными его свойствами, способны компенсировать друг друга. Возьмем два алгоритма, которые решают обе задачи, имеют различную природу, и построим над ними композицию с помощью агрегации результатов базовых алгоритмов. Такими алгоритмами являются обобщенная линейная модель, которая обнаруживает линейные зависимости в данных, и метод опорных векторов с нелинейным гауссовским ядром, который обнаруживает нелинейные зависимости из семейства гауссовских функций в данных.

3.4.1. Обобщенная линейная модель

Обобщенная линейная модель представляет собой два алгоритма: линейную регрессию и логистическую регрессию [3.35].

3.4.1.1. Линейная регрессия

Алгоритм линейной регрессии предполагает, что зависимость между входными объектами и прогнозом линейна. Это можно выразить формулой (3.3), в которой x – вектор прогнозируемых объектов из множества всех объектов X, w – вектор констант, а ε – аддитивная случайная величина, являющаяся ошибкой между линейными прогнозами и истинными значениями [3.36].

$$y(x) = w^T x + \varepsilon \tag{3.3}$$

Кроме предположения о линейности регрессионной зависимости, существует еще 4 предположения Гаусса-Маркова, на которых основана линейная регрессия:

1. наблюдения, по которым оценивается модель, случайны;
2. ни один признак не является линейной комбинацией других признаков;
3. ошибка случайна, то есть ее математическое ожидание равно 0;
4. дисперсия ошибки не зависит от значений признака, то есть она константна; данное свойство называется гомоскедастичностью [3.17].

Третье предположение о случайности ошибки подразумевает, что ее распределение является нормальным (Гауссовским). Это можно записать формулой (3.4), где μ – среднее значение, а σ^2 – дисперсия.

$$\varepsilon \sim \mathcal{N}(\mu(x), \sigma^2(x)) \tag{3.4}$$

Тогда формулу (3.3), можно переписать в виде (3.5).

$$\rho(y \mid x, \theta) = \mathcal{N}(y \mid \mu(x), \sigma^2(x)), \tag{3.5}$$

где ρ - это регресионная модель. В таком случае μ – это линейная зависимость, а σ^2 – фиксированная ошибка. Тогда параметрами модели становятся $\theta = (w, \sigma^2)$. Такая запись показывает явную связь между регрессионной моделью и нормальным распределением.

Обучение алгоритма линейной регрессии заключается в поиске вектора констант w, при условии, что фиксированная ошибка σ^2 должна быть минимальной. Ошибку линейной регрессии можно выражать с помощью разных метрик, чаще всего используют среднеквадратичную ошибку, представленную в формуле (3.6), где Z - множество объектов обучающей выборки, k - количество объектов обучающей выборки [3.42].

$$Q(a,Z) = \frac{1}{k}\sum_{i=1}^{k}(a(x_i) - y_i)^2 \tag{3.6}$$

Таким образом, обучение алгоритма линейной регрессии сводится к минимизации функционала ошибки. На практике данная задача в большинстве случаев решается градиентными методами, например, с помощью градиентного спуска, но в некоторых случаях применимо и аналитическое решение – метод наименьших квадратов [3.5; 3.41].

Следует отметить также, что к достоинствам линейной регрессии можно отнести быстроту и простоту создания модели. Также данная модель позволяет сделать дополнительные выводы о характере зависимости предикторов и отклика по коэффициентам регрессии. К тому же, данный алгоритм хорошо изучен: известны его проблемы и методы их решения, например, проблема мультиколлениарности, то есть проблема высокой корреляции между входными переменными множественной регрессии, что вызывает неустойчивость работы модели к малым изменениям исходных данных [3.22].

Также к недостаткам линейной регрессии можно отнести, ее неприменимость в случаях, когда не выполняется первое предположение Гаусса-Маркова, и не оптимальность оценок метода наименьших квадратов при невыполнении остальных предположений.

3.4.1.2. Логистическая регрессия

Алгоритм линейной регрессии может быть расширен для решения задачи бинарной классификации. Для этого требуется два изменения:

1. заменим нормальное распределение ошибки, на распределение Бернулли, поскольку оно больше подходит для поиска решения задачи бинарной классификации, $y \in \{0,1\}$ [3.12];
2. в качестве прогноза будем брать знак предсказанного прогноза, а не его значение.

Таким образом получим алгоритм логистической регрессии, представленный формулой (3.7) [3.36].

$$\rho(y \mid x, w) = B(y \mid sigm(w^T x)), \tag{3.7}$$

где B - распределение Бернулли, $sigm$ - логистическая функция, представленная формулой (3.8).

$$sigm(\eta) = \frac{1}{1 + exp(-\eta)} \tag{3.8}$$

Данный алгоритм, также, как и алгоритм линейной регрессии ищет вектор констант w , минимизируя логистическую ошибку. Для этого чаще всего используют стохастический градиентный метод для минимизации ошибки [3.21].

В качестве прогноза в данном случае мы получим вероятность ρ отнесения объекта к классу 1. Таким образом, объект получит идентификатор класса 0, если вероятность $\rho < 0{,}5$, и идентификатор класса 1, если вероятность $\rho > 0{,}5$. В случае, когда $\rho = 0{,}5$ классификатор не может определить класс объекта, поскольку он с равной вероятностью относит объект к обоим классам.

Преимущества логистической регрессии заключаются в возможности не только классифицировать объект, но также определить вероятность отношения объекта к классу. Кроме того, логистическая регрессия показывает хоро-

шие результаты за счет модификации весов в случае, когда объект находиться близко к границе классов.

К недостаткам логистической регрессии стоит отнести недостатки, наследуемые от метода стохастического градиента, который требует стандартизации данных, отсева выбросов, применения регуляризации весов и отбора признаков. Также в случаях, когда предположения логистической регрессии, наследуемые от предположений линейной регрессии, не выполняются, оценки вероятностей могут быть далеки от реальных данных.

3.4.2. Метод опорных векторов

Метод опорных векторов основан на построении оптимальной разделяющей гиперплоскости, для которой требуется, чтобы объекты обучающей выборки находились от нее максимально далеко. Таким образом, оптимальная разделяющая гиперплоскость должна быть максимально отдалена от ближайших к ней объектов всех классов, которые называют опорными векторами, то есть требуется максимизация зазора между опорными векторами разных классов для повышения надежности классификации [3.24].

Классификатор, строящий разделяющую поверхность классов, может быть выражен формулой (3.9).

$$a(x, w) = arg \max_{y \in Y} \sum_{j=1}^{n} w_{y_j} f_j(x) = arg \max_{y \in Y} \langle x, w_y \rangle, \tag{3.9}$$

где x - объект, y - идентификатор класса объекта, f_j - j-ый признак объекта, w_{y_j} – вес j-го признака, w_0 – порог принятия решения, $w = (w_0, w_1, , w_n)$ – вектор весов, $\langle x, w \rangle$ – скалярное произведение признакового описания объекта на вектор весов. Кроме того, предполагается, что введен нулевой признак $f_0(x) = -1$.

Отметим, что алгоритм $a(x, w)$ не изменится, если умножить веса w и w_0 на одну и ту же положительную константу, причем так, чтобы выполнялось условие, представленное в формуле (3.10) [3.45].

$$\min_{i=1...m} y_i(\langle w, x_i \rangle - w_0) = 1 \tag{3.10}$$

Таким образом, получим полосу, состоящую из множества точек $\{x : -1 \leqslant (\langle w, x_i \rangle - w_0) \leqslant 1\}$ и разделяющую классы. Границами данной полосы будут

две гиперплоскости с вектором весов в качестве вектора нормали, а оптимальная разделяющая гиперплоскость будет равноудалена от них. Именно на этих гиперплоскостях выполняется условие, записанное формулой (3.10). Максимизация данной полосы приводит к максимизации зазора между классами. Поскольку ширина полосы является обратной величиной нормы вектора весов, то ее минимизация приводит к максимизации зазора. Таким образом, получаем задачу квадратичного программирования: требуется найти значения параметров весов w и w_0, при которых выполняются m ограничений неравенств и норма вектора w минимальна [3.45]. Математическое определение задачи представлено формулой (3.11).

$$\begin{cases} \|w\| \to min, \\ y_i(\langle w, x_i \rangle - w_0) \geqslant 1, i = 1 \ldots m. \end{cases} \tag{3.11}$$

Метод опорных векторов также применяется и к решению задач регрессии. Он сводится к задаче классификации с двумя классами. В данном случае регрессионная зависимость делит плоскость на два класса: ниже линии регрессионной зависимости и выше нее. Разделяем данную плоскость так же, как и в задаче классификации полосой ширины h и строим линию, разделяющую классы, равноудаленно от границ полосы. Эта линия и будет являться регрессионной зависимостью и ответом задачи регрессии.

Главным преимуществом метода опорных векторов является сведение алгоритма обучения к задаче квадратического программирования, которая имеет единственное решение с эффективным вычислением, в том числе в случаях большого объема обучающей выборки. Кроме того, оптимальное положение разделяющей гиперплоскости зависит только от опорных векторов обучающей выборки, которые составляют малую долю всех объектов. Максимизация зазора приводит к более устойчивому алгоритму классификации.

К недостаткам данного алгоритма относят неустойчивость к шуму данных обучающей выборки, поскольку если выбросы попали в число опорных векторов они будут влиять на построение оптимальной гиперплоскости. Кроме того, для метода опорных векторов существует проблема линейной неразделимости, которая связана с невозможностью разделения классов гиперплоскостью в том пространстве, в котором заданы исходные объекты [3.10]. Один

из способов решения данной проблемы – увеличение размерности пространства за счет некоторого преобразования, которое приведет к линейной разделимости классов. Функция такого преобразования называется ядром, а пространство спрямляющим. Данный способ еще не изучен и не существует единого способа построения ядра и спрямляющего пространства, что также накладывает неудобство использования метода опорных векторов. В данной работе для метода прогнозирования предлагается использовать Гауссовское ядро.

Таким образом обобщенная линейная модель и метод опорных векторов с нелинейным гауссовским ядром решают обе задачи и строят разделяющую поверхность классов разными методами: классическими градиентными методами, минимизируя ошибку, и средствами квадратичного программирования, максимизируя зазор между классами, соответственно. Таким образом, алгоритмы способны дополнять друг друга. Кроме того, улучшение результата прогнозирования происходит за счет построения индивидуальной обучающей выборки для каждого базового алгоритма путем случайного выбора элементов из тренировочного набора, что снижает зашумленность данных.

В результате составления композиции на основе данных алгоритмов получим метод, который позволит делать прогнозы как номинальных, так и непрерывных значений, на основе данных, описываемых линейными или не линейными функциями из гауссовского семейства, не требуя предварительного аналитического и статистического анализа данных для настройки базовых алгоритмов композиции.

3.5. Организация данных и выбор задачи прогнозирования

В наши дни, как уже было сказано ранее, накоплены экзабайты данных ретроспективного характера. Только половина этих данных структурирована и может быть подвержена интеллектуальному анализу. Самый распространенный способ структуризации и хранения данных, это организация баз данных. В связи с этим, для обеспечения большей производительности и защиты, а также снижения накладных расходов на передачу данных, проведение

интеллектуального анализа средствами базы данных является более предпочтительным.

В данной работе используется технология системы управления реляционными базами данных Oracle Enterprise Edition версии 12c Oracle Data Mining. При построении моделей был использован PL/SQL API, который реализован пакетом DBMS_DATA_MINING [3.28].

За последние несколько лет сердечно-сосудистые заболевания стали наиболее распространенной причиной преждевременной смерти во всем мире по данным Глобального хранилища данных об охране окружающей среды Всемирной организации здравоохранения [3.46]. Ежегодно умирает около 17 миллионов человек от заболеваний сердечно-сосудистой системы, что составляет 31% от всех смертей в мире. Однако около 80% из них можно было бы предотвратить при ранней диагностике заболевания.

Сердечно-сосудистые заболевания включают в себя такие болезни как:

– инфаркт миокарда;
– инсульт;
– ишемическая болезнь сердца;
– стенокардия;
– аритмия;
– гипертония;
– нарушения свертываемости крови;
– и другие.

Причинами сердечно-сосудистых заболеваний являются атеросклероз, повреждение системы кровообращения при диабете, инфекции или вирусе. Данные болезни хорошо изучены и известны факторы, которые негативно влияют на сердечно-сосудистую систему человека, увеличивая риск появления сердечно-сосудистых заболеваний. К таким факторам относят:

– возраст;
– артериальное давление;
– показатели общего холестерина;
– индекс массы тела;
– наличие сахарного диабета;
– курение;

– и прочие.

Оценка рисков кардиологических заболеваний на основе анализов пациента является одной из задач медицины, которая хорошо поддается математическому и интеллектуальному анализу [3.1]. Кроме того, существует аналог ее решения, который называется формулой Фременгхэма или системой оценки риска SCORE [3.15]. Данная формула используется на практике врачами и основана на подсчете суммы баллов, которые присваиваются или изымаются в зависимости от показаний пациента. Стоит отметить, что данная формула гарантирует результат лишь на 30%.

В ходе данной работы были построены 3 модели, использующие предложенный метод композиции алгоритмов машинного обучения, на примере технологии Oracle Data Mining, прогнозирующие присутствие или отсутствие кардиологических заболеваний у пациента по трем различным наборам данных:

– набор «Framingham Heart Study» от Национального Университета Сердца, Легких и Крови [3.2];
– набор «Heart Disease» из UCI Machine Learning Repository [3.3];
– набор данных, предоставляемый участникам хакатона AgeHack для задачи «Предсказание ССЗ» на платформе ML Boot Camp [3.11].

Под моделью здесь и далее подразумевается совокупность конкретной задачи, алгоритма и набора данных. В качестве прогнозирования в данном случае понимается классификация пациента на два класса: 0 - здоров, 1 - болен, на основе измененных по тенденции развития показателей анализов и анамнеза врачем. Например, приходит молодой пациент, ведущий малоподвижный образ жизни и обладающий вредными привычками, к кардиологу. Врач на основе проведенных анализов предполагает, что уровень сахара и холестирина в крови поднимутся за 5 лет на несколько единиц, и настоящие значения заменяет предполагаемыми и запускает программу. Таким образом, врач сможет спрогнозировать сердечно-сосудистое заболевание у пациента.

Кроме этих трех моделей была построена еще одна модель, демонстрирующая применимость метода для решения задачи регрессии, которая прогнозирует время, через которое у пациента появится кардиологическое заболевание с некоторой вероятностью на поднаборе данных «Framingham Heart

Study» от Национального Университета Сердца, Легких и Крови, состоящего из пациентов с наличием сосудисто-сердечных заболеваний.

3.6. Описание исходных данных

Все наборы данных, использованные в этой работе, являются анонимизированными данными клинических исследований и находятся в открытом доступе в сети Интернет.

3.6.1. Набор данных «Framingham Heart Study»

Набор «Framingham Heart Study», который был запрошен в Национальном Университете Сердца, Легких и Крови, представляет собой данные о пациенте, такие как возраст, пол, индекс массы тела, показатели давления, наличие или отсутствие диабета и так далее [3.18]. Показатели для каждого пациента представлены 3 записями, поскольку были сняты 3 раза на протяжении 12 лет через равные промежутки времени. Из данного набора было выбрано случайным образом 10000 строк, которые позднее были разбиты на две части: обучающую и тестовую, которые составляют 7000 и 3000 записей соответственно. За один объект набора «Framingham Heart Study» был взят пациент в определенный период сдачи анализов и показаний, для этого переменные идентификатора пациента и периода были объединены в одну с помощью операции конкатенации.

В этом наборе данных содержится информация о 7 разных типах сердечно-сосудистых заболеваний:

- инфаркт миокарда;
- инсульт;
- ишемическая болезнь сердца;
- стенокардия;
- коронарная болезнь сердца;
- гипертония;
- иные сердечно-сосудистые заболевания (CVD).

Для проведения сравнительного анализа с остальными наборами данных, в которых всего один атрибут является индикатором присутствия или отсутствия у пациента сердечно-сосудистых заболеваний, было решено преобразовать 7 атрибутов в один с помощью оператора логического сложения, поскольку каждый из семи признаков бинарный: принимает значение 0 или 1. В таком случае у пациента присутствуют сердечно-сосудистые заболевания, если он болен хотя бы одним видом. Показания о холестерине присутствовали только у одной шестой части пациентов, потому как этот показатель измерялся только в последнем периоде из трех, что не позволяет восстановить неопределенные значения. В связи с этим, показания о холестерине не попали в выборки для обучения и тестирования моделей.

3.6.2. Набор данных «Heart Disease»

Набор данных «Heart Disease» из репозитория данных для машинного обучения Калифорнийского Университета в Ирвине представляет собой 303 записи о пациентах. Каждая запись состоит из 14 атрибутов, в которые входят возраст, пол, тип болей в груди (один из наиболее частых симптомов сердечно-сосудистых заболеваний), показатели кровяного давления, показатели холестерина, уровень сахара в крови, результаты электрокардиографии, сердечный ритм и прочие [3.4].

Прогнозируемый атрибут принимает значения от 0 до 4, где 0 обозначает отсутствие у пациента сердечно-сосудистых заболеваний, а остальные числа обозначают результаты ангиографии. Ангиография – это метод получения трехмерных реконструкций артериальной системы пациента для того, чтобы качественно и количественно оценить поражение артерий [3.33]. Приведем задачу мулитиклассовой классификации к бинарной, для этого заменим значения 1, 2, 3 и 4 таким образом, что 0 будет обозначать отсутствие у пациента заболевания, а 1 – присутствие.

Поскольку рассматриваемый набор данных содержит мало объектов, не будем разбивать его на обучающую и тестовую выборки, а применим метод кросс-валидации [3.13]. Данный подход заключается в разбиении исходной выборки на блоки и обучения столько раз, на сколько блоков была разбита исходная выборка. Причем при каждом обучении один блок выступает в ро-

ли тестовой выборки, а остальные – в качестве обучающей. Таким образом, мы обучим алгоритм на всех объектах исходной выборки, а для оценки качества работы алгоритма усредним полученные метрики на каждом тестовом блоке. В данной работе набор «Heart Disease» был разбит на 3 равных блока (в каждом по 101 объекту).

3.6.3. Набор данных хакатона AgeHack

Третий набор данных, позволяющий предсказывать наличие или отсутствие сердечно-сосудистых заболеваний, для данный работы был взят с платформы для проведения чемпионатов и хакатонов по машинному обучению от компании Mail.Ri Group Machine Learning Boot Camp. Одной из задач хакатона AgeHack было предсказание наличия сердечно-сосудистых заболеваний по результатам классического врачебного осмотра, причем участникам хакатона, а после соревнований и всем заинтересованным пользователям платформы, предоставлялись настоящие клинические данные, собранные в медицинских учреждениях.

Данный набор состоит из 100000 записей, в которых собрана информация о поле, возрасте, росте, весе, показателей давления, холестерина, глюкозы, о вредных привычках и образе жизни пациентов [3.44]. Поскольку данные получены из медицинских учреждений в них находится довольно много опечаток и неточностей. В связи с этим из набора были исключены записи с явными ошибками, например, рост пациента составляет 1 метр, а вес 200 килограмм, и большими отклонениями от нормы: систалическое артериальное давление составляет 10 мм рт. ст., а диастолическое 80 мм рт. ст. Из оставшихся записей были случайным образом выбраны 10000, которые далее были поделены на обучающую и тестовую выборки, составляющих 7000 и 3000 объектов соответственно.

Далее в соответствии с рекомендациями Oracle были преобразованы типы данных в обоих наборах данных следующим образом:

- категориальные переменные, например, идентификатор пациента, преобразовываются к строковому типу данных;
- даты преобразовываются к численному типу данных посредством вычитания из сегодняшней даты – даты, указанной в наборе.

Таким образом, были сформированы обучающие и тестовые выборки, в той форме, в которой модель способна правильно их трактовать.

3.7. Реализация метода композиции базовых алгоритмов

Для реализации метода композиции базовых алгоритмов машинного обучения потребовалось написать на языке PL/SQL код-надстройку для вызова процедуры создания модели пакета DBMS_DATA_MINING. Это связно с тем, что помимо непосредственного вызова процедуры создания модели требуется разделение обучающей выборки на тренировочную и контрольную, подготовка индивидуальных, т.е. для каждого базового алгоритма композиции, выборок, настройка конфигурации каждого базового алгоритма на основе контрольной выборки и получение прогнозируемого значения на основе результатов базовых моделей. Под базовой моделью в данной работе понимается совокупность базового алгоритма и его индивидуальной выборки.

3.7.1. Выделение контрольной и тренировочной выборок

Для настройки конфигурации базовых алгоритмов требуется выделить подмножество объектов, которые не будут входить в подмножество данных, подающихся на вход базовому алгоритму для обучения, и подмножество данных, на котором не будет производиться оценка качества предложенного метода. Для этого выделяют еще один тип выборки: контрольный [3.9]. Именно на контрольной выборке будут оценены алгоритмы, с разными значениями параметров конфигурации, и сформирована конфигурация, содержащая наилучшее сочетание параметров по требуемым метрикам. Параметры конфигурации - значения параметров алгоритма, например, пороговое значение для логистической регрессии. Эти параметры позволяют производить более тонкую настройку модели, тем самым увеличивая качество прогноза, или вводить ограничения для увеличения производительности, например размер буфера ядра для метода опорных векторов.

Ранее в данной работе термины обучающая и тренировочная выборка употреблялись в качестве синонимов. Поскольку теперь и далее у нас есть обу-

чающая выборка объектов, которая подается на вход предлагаемому методу, и есть обучающая выборка, которая подается на вход каждому базовому алгоритму предложенного метода, будем обозначать первую – обучающей, а вторую – тренировочной. Схема разделения исходных данных для обучения и оценки качества предлагаемого метода представлена на рисунке 3.1.

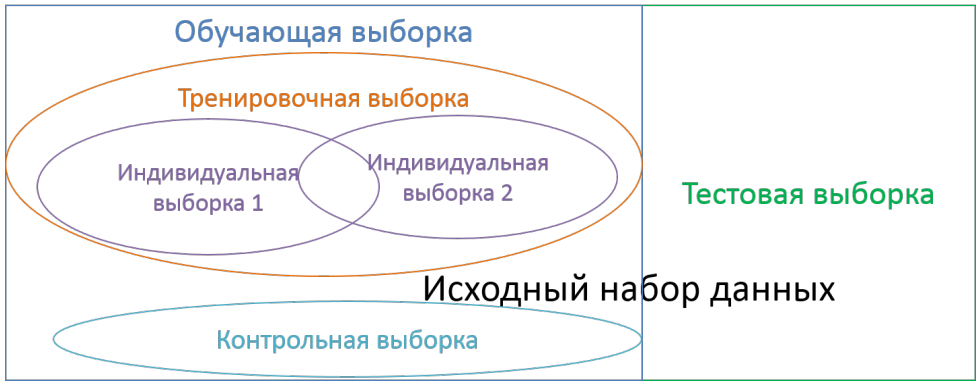

Рис. 3.1. Схема разделения исходных данных

В данной работе в качестве контрольной выборки было решено использовать 20% объектов обучающей выборки, выбранных случайно. Остальные объекты составляют тренировочную выборку.

3.7.2. Создание индивидуальной выборки

Поскольку метод основывается на композиции, агрегирующей результаты нескольких базовых моделей, построенных на основе различных подмножеств элементов тренировочной выборки, первым этапом построения модели является создание этих выборок. В нашем случае их две, поскольку по описанным ранее причинам в методе используется два базовых алгоритма.

Отметим, что каждая такая выборка должна содержать подмножество элементов тренировочной выборки, которые могут повторяться как в одной и той же индивидуальной выборке, так и в разных. В следствии чего, строки тренировочного набора были пронумерованы, каждый раз генерировалось псевдослучайное число и в текущую индивидуальную подвыборку была добавлена та строка из тренировочной выборки, номер которой совпадал с полученным числом. Данная операция была повторена столько раз, сколько строк требовалось в подвыборке. В данной работе размер индивидуальной

выборки составлял 70% от тренировочной выборки, поскольку он должен быть меньше, чем размер выборки, выделенный на обучение базовых алгоритмов, но все же строк должно быть достаточно для устойчивости к переобучению.

Для повышения производительности было принято решение каждый раз добавлять не по одной строке, а по 1000, для наборов данных, содержащих более 1000 объектов в тренировочной выборке. Для этого сначала было сгенерировано 1000 псевдослучайных чисел, а затем в таблицу каждой индивидуальной выборки были добавлены те строки из тренировочной выборки, номера которых совпадали с полученными числами. Поскольку псевдослучайные числа повторялись, то каждый раз в выборку добавлялось меньше 1000 строк. Данное свойство было учтено, вследствие чего алгоритм генерации псевдослучайных чисел и их добавления в подвыборку был запущен на 1 раз больше. Псевдокод алгоритма создания индивидуальной выборки представлен на рисунке 3.2.

3.7.3. Настройка конфигурации базовых моделей

Технология Oracle Data Mining позволяет производить более тонкую настройку модели с помощью указания значений параметров алгоритма. У каждого алгоритма свой набор параметров. В таблице 3.1 представлены параметры для алгоритма обобщенной линейной модели, а в таблице 3.2 для метода опорных векторов. В этих таблицах указано название параметра, на что он влияет и его значение. Эти параметры не изменялись в ходе работы предложенного метода.

В таблицах 3.3 и 3.4 представлены параметры и их значения, которые они принимали в ходе настройки конфигурации базовых моделей, для обобщенной линейной модели и метода опорных векторов соответственно.

Настройка конфигурации каждой базовой модели происходила последовательно: на каждом шаге изменялось значение одного параметра, строилась модель на основе новой конфигурации, оценивалось качество полученной модели на контрольной выборке, в том случае если качество увеличивалось, то параметр оставался с новым значением в конфигурации, иначе ему возвращалось предыдущее значение.

Algorithm

Input: Traint set - set, Individual sets count - n

Output: n individual sets - $Isets$

1. $st = trainsetsize$;
2. $s = st * 0.7$;
3. **for** $i = 1 \rightarrow n$ **do**
4. **if** $s > 1000$ **then**
5. **for** $j = 1 \rightarrow s/1000$ **do**
6. $numbers \leftarrow \varnothing$;
7. **for** $k = 1 \rightarrow 1000$ **do**
8. $numbers \leftarrow \{x \in \mathbb{Z} \mid 1 \leqslant x \leqslant st\}$;
9. $Isets_i \leftarrow \{\forall obj \in set, obj.Index \in numbers\}$;
10. **else**
11. **for** $j = 1 \rightarrow 3$ **do**
12. $numbers \leftarrow \varnothing$;
13. **for** $k = 1 \rightarrow st$ **do**
14. $numbers \leftarrow \{x \in \mathbb{Z} \mid 1 \leqslant x \leqslant st\}$;
15. $Isets_i \leftarrow \{\forall obj \in set, obj.Index \in numbers\}$;
16. **for** $obj \in Isets_i$ **do**
17. $objId = concat(objId, obj.IsetIndex)$;

Рис. 3.2. Псевдокод алгоритма `SplitTrainToIndividual`

Таблица 3.1

Параметры конфигурации для обобщенной линейной модели

Имя параметра	Значение параметра	Описание	Комментарий
ODMS_MISSING_VALUE_TREATMENT	ODMS_MISSING_VALUE_MEAN_MODE	Определяет стратегию замены пропущенных в обучающей выборке значений: удалить строку и поставить среднее значение аттрибута.	В данной работе не будем удалять строки, поскольку их может быть мало, а будем заменять пропущенное значение на среднее.
ODMS_ROW_WEIGHT_COLUMN_NAME	null	Имя атрибута обучающей выборки, для которого есть дополнительный вес для каждой строки.	Метод не знает о бизнес правилах и не может выделить более важный атрибут.
PREP_AUTO	PREP_AUTO_ON	Разрешение или запрет автоматической подготовки данных.	Включим автоматическую подготовку данных для нормировки атрибутов и обработки пропущенных в обучающей выборке значений.
CLASS_WEIGHTS_TABLE_NAME	null	Имя таблицы, которая содержит веса для каждого значения прогнозируемого класса для логистической регрессии.	Данный параметр нужен, если классы не сбалансированы. В методе выборки строятся сбалансированно и данный параметр нам настраивать не требуется.
GLMS_REFERENCE_CLASS_NAME	наиболее часто встречающееся	Целевое значение, которое будет использоваться в качестве эталонного значения в модели логистической регрессии.	

Таблица 3.2

Параметры конфигурации для метода опорных векторов

Имя параметра (Значение параметра)	Описание	Комментарий
SVMS_ ACTIVE_LEARNING (SVMS_AL_ DISABLE)	Когда активное обучение включено, метод опорных векторов использует активное обучение для снижения размера модели. Иначе строит стандартную модель.	Используем малые наборы данных (размер<100000 объектов), поэтому не будем применять активное обучение.
SVMS_ COMPLEXITY_ FACTOR (null)	Значение коэффициента сложности.	Значение по умолчанию вычисляется на основе данных алгоритмом.
SVMS_EPSILON (null)	Значение фактора остановки алгоритма.	Значение по умолчанию вычисляется на основе данных алгоритмом.
SVMS_ KERNEL_CACHE_ SIZE (50000000)	Размер кеша для ядра.	По умолчанию 50000000 Б.
SVMS_ KERNEL_FUNCTION (svms_gaussian)	Тип ядра алгоритма.	Используем нелинейное гауссовское.
SVMS_ STD_DEV (null)	Значение стандартного отклонения для метода опорных векторов.	Значение по умолчанию вычисляется на основе данных алгоритмом.
PREP_AUTO (PREP_ AUTO_ON)	Разрешение или запрет автоматической подготовки данных.	Включим автоматическую подготовку данных для нормировки атрибутов и обработки пропущенных в обучающей выборке значений.

Таблица 3.3

Изменяемые параметры конфигурации для обобщенной линейной модели

Имя параметра	Применяемые значения параметра	Описание	Комментарий
GLMS_-CONF_-LEVEL	0.9; 0.95; 0.99.	Уровень достоверности для доверительных интервалов.	
GLMS_-RIDGE_-REGRESSION	GLMS_-RIDGE_-REG_-ENABLE. GLMS_-RIDGE_-REG_-DISABLE	Разрешение или запрет использования гребневой регрессии при построении модели [.]	Решает проблему мультиколлинеарности данных.
GLMS_-RIDGE_-VALUE	1; 10; 100.	Коэффициент регуляризации.	Его можно указывать только в том случае, когда разрешена гребневая регрессия.

Таблица 3.4

Изменяемые параметры конфигурации для обобщенной линейной модели

Имя параметра	Применяемые значения параметра	Описание	Комментарий
SVMS_-CONV_-TOLERANCE	0.001; 0.005; 0.01.	Предел сходимости для метода опорных векторов.	По умолчанию равен 0.001.

Качество модели для задачи классификации оценивалось по двум метрикам: доля правильных ответов (accuracy) и точность (precision) или полнота (recall) [3.34]. Формулы данных метрик представлены в таблице 3.5. Какая метрика важнее - точность или полнота - определяет пользователь, указывая соответствующее значение входного параметра метода и идентификатор класса, для которого будем максимизировать данную метрику, исходя из особенностей предметной области. Точность и полнота определяют какая из ошибок классификатора несет больше затрат: ошибка ложного срабатывания или ложный пропуск, соответственно. Для оценки риска сердечно-сосудистых заболеваний более критичной и дорогостоящей ошибкой (цена - человеческая жизнь) является отсутствие диагноза у больного пациента, чем диагностирование сердечно-сосудистого заболевания у здорового, следовательно, метрика точности для класса 1 (болен) будет более значимой.

Качество модели для задачи регрессии оценивалось по среднеквадратичной ошибке, формула которой также представлена в таблице 3.5. Поскольку эта метрика более чувствительна по сравнению с долей неправильных ответов для задачи классификации, нам не требуется дополнительных метрик для оценки качества модели.

Отметим, что в случае задачи классификации доля правильных ответов и точность или полнота максимизировались в интервале [0,1], а в случае задачи регрессии среднеквадратичная ошибка, наоборот, минимизировалась на всем множестве действительных чисел.

Для удобства перебора различных параметров и их значений была задана начальная конфигурация и был составлен массив, содержащий названия параметров со значениями, по которому алгоритм итерировался и последовательно изменял значение параметров.

Процедуры настройки конфигурации базовых моделей для обобщенной линейной модели и метода опорных векторов представлены на рисунках 3.3 и 3.4 соответственно. Отметим, что данные процедуры могут выполняться параллельно, поскольку не зависят друг от друга.

Таблица 3.5

Формулы используемых метрик

Название	Формула	Обозначения
Доля правильных ответов (accuracy)	$Q(a,Z) = \frac{1}{m}\sum_{i=1}^{m}[a(x_i) = y_i]$	a – прогнозирующая модель, Z – тестовая выборка, x_i – i-ый объект тестовой выборки, y_i – исходный идентификатор класса объекта x_i, $a(x_i)$ – спрогнозированный идентификатор класса объекта x_i моделью, m – количество объектов тестовой выборки X.
Точность (presicion)	$p(a,Z) = \frac{TP}{TP+FP}$	a – прогнозирующая модель, Z – тестовая выборка, TP – количество прогнозов, совпадающих с выбранным индикатором класса (считаем метрику для 1 класса) и с актуальным значением (True positive), FP (False positive) – количество прогнозов, совпадающих с выбранным индикатором класса (считаем метрику для 1 класса), но не совпадающих с актуальным значением.
Полнота (recall)	$r(a,Z) = \frac{TP}{TP+FN}$	a – прогнозирующая модель, Z – тестовая выборка, TP – количество прогнозов, совпадающих с выбранным индикатором класса (считаем метрику для 1 класса) и с актуальным значением (True positive), FN (False negative) – количество прогнозов, не совпадающих с выбранным индикатором класса (считаем метрику для 1 класса) и не совпадающих с актуальным значением.
Среднеквадратичная ошибка	$RMSE = \left(\frac{1}{m}\sum_{i=1}^{m}(y_i - \bar{y}_i)^2\right)^{1/2}$	y_i – исходный идентификатор класса объекта, $\bar{y}_i$ – спрогнозированный идентификатор класса объекта, m – количество объектов тестовой выборки.

Procedure

Input: Train set - T, Control set - C, Problem type - $type$, Important metric - m

Output: Best config settings for GLM - SG

1. $Initializechangingparametersarray forGLM(\bar{S})$;
2. $Initializedefaultconfiguration forGLM(SG)$;
3. $model = Buildmodel(T, SG, type)$;
4. $bMetr = GetMetrics(model, C, type)$;
5. **for** $s \in \bar{S}$ **do**
6. **if** $s.SetName = GLMS\_RIDGE\_REGRESSION$ **then**
7. $ridge = \{x \in SG, x.SetName = GLMS\_RIDGE\_VALUE\}$;
8. $SG = SG \setminus \{x \in SG, x.SetName = GLMS\_RIDGE\_VALUE\}$;
9. $old = \{x \in SG, x.SetName = s.SetName\}$;
10. $SG = SG \setminus \{x \in SG, x.SetName = s.SetName\} \cup s$;
11. $model = Buildmodel(T, SG, type)$;
12. $metr = GetMetrics(model, C, type)$;
13. **if** $(type = C) \wedge (((m = P) \wedge ((metr.acc + metr.prec > bMetr.acc + bMetr.prec) \vee (metr.acc + metr.prec = bMetr.acc + bMetr.prec \wedge metr.rec > bMetr.rec))) \vee ((m = R) \wedge ((metr.acc + metr.rec > bMetr.acc + bMetr.rec) \vee (metr.acc + metr.rec = bMetr.acc + bMetr.rec \wedge metr.prec > bMetr.prec)))) \wedge (type = R \wedge metr.rmse > bMetr.rmse)$ **then**
14. $bMetr = metr$;
15. **else**
16. $SG = SG \setminus s \cup old$;
17. **if** $s.SetName = GLMS\_RIDGE\_REGRESSION$ **then**
18. $SG = SG \cup ridge$;

Рис. 3.3. Псевдокод алгоритма `ConfigGLMAlgSettings`

Procedure

Input: Train set - T, Control set - C, Problem type - $type$, Important metric - m

Output: Best config settings for SVM - SS

1. $Initializechangingparametersarray for SVM(\bar{S})$;
2. $Initializedefaultconfiguration for SVM(SS)$;
3. $model = Buildmodel(T, SS, type)$;
4. $bMetr = GetMetrics(model, C, type)$;
5. **for** $s \in \bar{S}$ **do**
6. $\quad old = \{x \in SS, x.SetName = s.SetName\}$;
7. $\quad SG = SG \setminus \{x \in SS, x.SetName = s.SetName\} \cup s$;
8. $\quad model = Buildmodel(T, SS, type)$;
9. $\quad metr = GetMetrics(model, C, type)$;
10. $\quad$ **if** $(type = C) \wedge (((m = P) \wedge ((metr.acc + metr.prec > bMetr.acc + bMetr.prec) \vee (metr.acc + metr.prec = bMetr.acc + bMetr.prec \wedge metr.rec > bMetr.rec))) \vee ((m = R) \wedge ((metr.acc + metr.rec > bMetr.acc + bMetr.rec) \vee (metr.acc + metr.rec = bMetr.acc + bMetr.rec \wedge metr.prec > bMetr.prec)))) \wedge (type = R \wedge metr.rmse > bMetr.rmse)$ **then**
11. $\quad\quad bMetr = metr$;
12. $\quad$ **else**
13. $\quad\quad SS = SS \setminus s \cup old$;

Рис. 3.4. Псевдокод алгоритма `ConfigSVMAlgSettings`

3.7.4. Описание процедур, реализующих метод композиции базовых алгоритмов машинного обучения

Как уже было сказано ранее, реализация предложенного в данной работе метода требует надстройки над PL/SQL API, поставляемый технологией Oracle Data Mining. Для этого был создан PL/SQL пакет, который содержит:

A. процедуру обучения модели;
B. функцию получения прогноза;
C. функции получения значения метрики:
    1. доли правильных ответов (для задачи классификации);
    2. точности (для задачи классификации);
    3. полноты (для задачи классификации);
    4. среднеквадратичной ошибки (для задачи регрессии).

Процедура обучения модели принимает на вход:

– название модели;
– имя таблицы с обучающей выборкой;
– тип задачи (классификация или регрессия);
– имя атрибута, являющийся идентификатором объекта;
– имя атрибута, который требуется спрогнозировать.

Для задачи классификации также на вход принимается название метрики (полнота или точность) и идентификатор класса, для которого требуется максимизировать выбранную метрику. Далее в процедуре обучающая выборка делится на тренировочную и контрольную. Затем тренировочная выборка делится на две индивидуальные. После этого для каждого из двух базовых алгоритмов метода происходит настройка конфигурации по описанному выше алгоритму и на основе полученной конфигурации строятся 2 финальные базовые модели (последняя модель не обязательно лучшая, так как последний измененный параметр мог ухудшить качество модели). Псевдокод процедуры обучения представлен на рисунке 3.5.

Функция получения прогноза принимает на вход:

– название модели;
– имя таблицы, с записями об объектах, на которых надо оценить качество предложенного в данной работе метода;
– имя атрибута, которые требуется спрогнозировать;

Algorithm

Input: Model name -$name$, Learning set - L, Problem type - $type$, Important metric - m, Name of object identificator - id, Name of target attribute - $target$

1. $T \leftarrow \varnothing$;
2. $C \leftarrow \varnothing$;
3. $SplitLearnToTrainAndControl(L, 20\%, T, C)$;
4. $SplitTrainToIndividual(T, 2)$;
5. $ConfigGLMAlgSettings(T_1, type, m)$;
6. $ConfigSVMAlgSettings(T_2, type, m)$;
7. $DropAllSettingsModel(name)$;
8. $DropIndividualSets(name)$;
9. $SplitTrainToIndividual(T, 2)$;
10. $GLMmodel = Buildmodel(T_f 1, SG, type)$;
11. $SVMmodel = Buildmodel(T_f 2, SS, type)$;

Рис. 3.5. Псевдокод алгоритма `LearnModel`

– имя атрибута, являющийся идентификатором объекта.

Функция возвращает имя таблицы, в которой содержатся идентификаторы переданных в функцию объектов и спрогнозированные значения. Данная таблица формируется на основе запросов к базовым моделям и выбора того прогноза, у которого вероятность больше, для задачи классификации, и среднего арифметического значения – для задачи регрессии. Вероятности отнесения объекта к спрогнозированному классу возвращает PL/SQL API при решении задачи классификации, и выражают уверенность базового классификатора в отнесение объекта к данному классу [3.30].

Функции подсчета метрики принимают на вход:

– название модели;
– имя таблицы, с записями об объектах, для которых надо составить прогноз (может быть и 1 объект);
– тип задачи (классификация или регрессия);
– имя атрибута, являющийся идентификатором объекта;
– для полноты и точности также требуется указать идентификатор класса, для которого вычисляется метрика.

Возвращают эти функции действительное число, являющееся значением метрики, соответствующей вызванной процедуры. Данные функции основаны на получении прогноза для входных объектов и подсчете метрики по представленным ранее в таблице 3.5 математическим формулам.

3.8. Анализ полученных результатов

После построения моделей, прогнозирующих риски кардиологических заболеваний, было оценено качество этих моделей с помощью уже описанных ранее метрик на основе тестовых выборок. Значения метрик для метода прогнозирования представлены в строке с названием алгоритма «Предложенный метод прогнозирования» в таблицах 3.6, 3.7 и 3.8. Для проведения сравнительного анализа в ходе данной работы были также построены модели на основе тех же обучающих выборок, что и предложенный метод, с помощью реализованных алгоритмов машинного обучения в технологии Oracle Data Mining, решающих задачу классификации [3.27].Такими алгоритмами являются:

- наивный байесовский классификатор;
- дерево решений;
- тип задачи (классификация или регрессия);
- обобщенная линейная модель;
- метод опорных векторов.

Для данных алгоритмов не настраивалась конфигурация, а параметры алгоритмов принимали значения по умолчанию, то есть те значения, которые были определены разработчиками или вычисляются на основе данных во время построения и обучения модели [3.29].

Из таблицы 3.6 видно, что для набора данных «Framingham Heart Study» предлагаемый метод прогнозирования находится на первом месте по доле правильных ответов и точности. Именно эти две метрики требуется максимизировать в задаче оценки рисков сердечно-сосудистых заболеваний.

По результатам исследования для набора данных «Heart Disease», представленным в таблице 3.7 видно, что точность предложенного в данной работе метода уступает только обобщенной линейной модели. Точность ком-

Таблица 3.6

Результаты исследования решения задачи прогнозирования присутствия или отсутствия кардиологического заболевания у пациента для набора данных «Framingham Heart Study»

Алгоритм	Доля правильных ответов	Точность	Полнота
Наивный байесовский классификатор	0.847	0.522	0.305
Дерево решений	0.868	0.745	0.242
Обобщенная линейная модель	0.869	0.784	0.231
Метод опорных векторов	0.87	0.812	0.229
Предложенный метод прогнозирования	0.872	0.885	0.212

Таблица 3.7

Результаты исследования решения задачи прогнозирования присутствия или отсутствия кардиологического заболевания у пациента для набора данных «Heart Disease»

Алгоритм	Доля правильных ответов	Точность	Полнота
Дерево решений	0.739	0.739	0.735
Метод опорных векторов	0.815	0.825	0.762
Наивный байесовский классификатор	0.828	0.822	0.797
Обобщенная линейная модель	0.832	0.856	0.763
Предложенный метод прогнозирования	0.835	0.847	0.784

Таблица 3.8

Результаты исследования решения задачи прогнозирования присутствия или отсутствия кардиологического заболевания у пациента для набора данных хакатона «AgeHack»

Алгоритм	Доля правильных ответов	Точность	Полнота
Метод опорных векторов	0.644	0.616	0.719
Дерево решений	0.733	0.748	0.683
Обобщенная линейная модель	0.733	0.756	0.668
Предложенный метод прогнозирования	0.735	0.751	0.685
Наивный байесовский классификатор	0.735	0.768	0.655

позиции, для каждого базового алгоритма которой производится более тонкая настройка конфигурации при условии максимизации доли правильных ответов и точности, меньше, чем у базового алгоритма без дополнительной настройки, из-за агрегации результатов двух алгоритмов, точность одного из которых значительно ниже. Заметим, что данное правило не распространяется на долю правильных ответов, которая для предложенного метода и набора данных «Heart Disease» максимальна среди моделей исследования.

Для набора данных хакатона «AgeHack», результаты исследования которого представлены в таблице 3.8, можно сказать, что предложенный метод меньше всего дал неправильных ответов на тестовой выборке и его доля правильных ответов совпадает с той же метрикой для наивного баейсовского классификатора. Точность же предложенного метода ниже точности наивного байесовского классификатора и обобщенной линейной модели.

Таким образом, метод композиции алгоритмов для решения задачи прогнозирования, основанный на двух противоположных по природе алгоритмах машинного обучения, таких, как обобщенная линейная модель и машина опорных векторов с гауссовским ядром, увеличивает долю правильных ответов базовых алгоритмов в случае решения задачи оценки рисков сердечно-сосудистых заболеваний у пациентов на основе их анамнеза и анализов, и компенсирует точность базового алгоритма с меньшим показателем точности за счет второго базового алгоритма, точность которого выше.

Теперь рассмотрим оценки ошибок для моделей, решающих задачу регрессии, которые представлены в таблице 3.9.

Таблица 3.9

Результаты исследования решения задачи прогнозирования время, через которое у пациента появится кардиологическое заболевание с некоторой вероятностью на поднаборе данных «Framingham Heart Study»

Алгоритм	Среднеквадратичная ошибка
Метод опорных векторов	2292.841
Предложенный метод прогнозирования	1776.751
Обобщенная линейная модель	1766.848

Наилучший результат у модели, построенной на основе обобщенной линейной модели, а наихудший – у модели, построенной на основе метода

опорных векторов с гауссовским ядром. Результат предложенного в данной работе метода располагается между результатами своих базовых алгоритмов с параметрами, значение которых определяется разработчиками технологии Oracle Data Mining. Это происходит за счет того, что каждый раз значением прогноза становится среднее значение из результатов двух базовых алгоритмов, и тем самым значение с большим отклонением компенсируется значением с меньшим, и значение с меньшим отклонением отклоняется еще больше за счет значения с большим.

Заметим, что среднеквадратичная ошибка предложенного метода ближе к лучшему результату, нежели худшему. Этот эффект достигается настройкой конфигурации базовых алгоритмов метода прогнозирования на основе обучающей выборки.

Выводы

Предложенный и реализованный в данной работе метод прогнозирования, основанный на двух противоположных по природе алгоритмах машинного обучения, таких, как обобщенная линейная модель и метод опорных векторов с гауссовским ядром, обладает такими свойствами как усиление качества базовых алгоритмов, устойчивость к исходным зашумленным данным и к переобучению. Последние два свойства наследуются от метода композиционного агрегирования результатов нескольких базовых алгоритмов, построенных на основе различных подмножествах элементов обучающей выборки.

По результатам исследования предложенного в данной работе метода на трех различных наборах данных анализов и анамнезов пациентов можно выдвинуть гипотезу о применимости данного метода для решения задачи оценки рисков сердечно-сосудистых заболеваний.

Библиографический список

3.1. 2016 European Guidelines on cardiovascular disease prevention in clinical practice: The Sixth Joint Task Force of the European Society of Cardiology and Other Societies on Cardiovascular Disease Prevention in Clinical Practice (constituted by representatives of 10 societies and by invited experts)Developed with the special contribution of the European Association for Cardiovascular Prevention & Rehabilitation (EACPR) / M. F. Piepoli [et al.] // European Heart Journal. — 2016. — Vol. 37, no. 29. — P. 2315–2381. — URL: http://dx.doi.org/10.1093/eurheartj/ehw106.

3.2. About the Framingham Heart Study. — 2018. — URL: https://www.framinghamheartstudy.org/fhs-about/ (visited on 03.04.2018).

3.3. *Aha D. W.* Heart Disease Databases. — 1988. — URL: http://archive.ics.uci.edu/ml/datasets/heart+disease (visited on 03.04.2018).

3.4. *Aha D. W.* Heart Disease Databases. Data Set Description. — 1988. — URL: http://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/heart-disease.names (visited on 03.04.2018).

3.5. *Ahmerov R. R.* Metody optimizacii gladkih funkcij. — 1992. — P. 99. — (In Russian).

3.6. Applied statistics. Classification and reduction of dimensionality. Vol. 3 / S. A. Aivazyan [et al.]; ed. by S.A.Aivazyan. — Finansy i statistika, 1989. — 607 p. — (In Russian).

3.7. *Boser B. E.*, *Guyon I. M.*, *Vapnik V. N.* A Training Algorithm for Optimal Margin Classifiers // Proceedings of the Fifth Annual Workshop on Computational Learning Theory. — Pittsburgh, Pennsylvania, USA: ACM, 1992. — P. 144–152. — (Ser.: COLT '92).

3.8. *Breiman L.* Bagging predictors // Machine Learning. — 1996. — Vol. 24, no. 2. — P. 123–140.

3.9. *Brownlee J.* What is the Difference Between Test and Validation Datasets? — 2017. — URL: https://machinelearningmastery.com/difference-test-validation-datasets/.

3.10. *Burges C. J. C.* Advances in Kernel Methods // / ed. by B. Scholkopf, C. J. C. Burges, A. J. Smola. — Cambridge, MA, USA: MIT Press, 1999. — Chap. Geometry and Invariance in Kernel Based Methods — p. 89–116.

3.11. *Camp M.* AgeHack. CV Prediction. — 2017. — URL: https://mlbootcamp.ru/round/12/sandbox/ (visited on 04.04.2018); (In Russian).

3.12. *Cramer H.* Random Variables and Probability Distributions. — CAMBRIDGE UNIV PR, 2004. — 132 p.

3.13. Cross-Validation / K. A. Ross [et al.] // Encyclopedia of Database Systems. — Springer US, 2009. — P. 532–538.

3.14. Data Mining: Practical Machine Learning Tools and Techniques / I. H. Witten [et al.]. — 4th ed. — Amsterdam: Morgan Kaufmann, 2017. — P. 654.

3.15. Development of a risk score for atrial fibrillation (Framingham Heart Study): a community-based cohort study / R. B. Schnabel [et al.] // The Lancet. — 2009. — Vol. 373, no. 9665. — P. 739–745.

3.16. *Dey A.* Machine Learning Algorithms: A Review // International Journal of Computer Science and Information Technologies. — 2016. — Vol. 7, no. 3. — P. 1174–1179.

3.17. *Dougherty C.* Introducion to Panel Data Models // Introduction to Econometrics. Vol. 15. — 2007.

3.18. Framingham Heart Study Longitudinal Data Documentation. — URL: https://biolincc.nhlbi.nih.gov/static/studies/teaching/framdoc.pdf (visited on 15.12.2017).

3.19. *Freund Y.*, *Schapire R. E.* A Short Introduction to Boosting // In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence. — Morgan Kaufmann, 1999. — P. 1401–1406.

3.20. *Gaidyshev I.* Analysis and data processing. — SPb: Piter, 2001. — 750 p.

3.21. *Granichin O. N.* Vvedenie v metody stohasticheskoj optimizacii i ocenivaniya. — Publishing house of St. Petersburg University Publishing house of St. Petersburg University, 2003. — 131 p. — (In Russian).

3.22. *Gunst R. F.*, *Webster J. T.* Regression analysis and problems of multicollinearity // Communications in Statistics. — 1975. — Vol. 4, no. 3. — P. 277–292.

3.23. *Hastie T.*, *Tibshirani R.* Generalized Additive Models: Rejoinder // Statistical Science. — 1986. — Vol. 1, no. 3. — P. 314–318.

3.24. *Hastie T.*, *Tibshirani R.*, *Friedman J.* The Elements of Statistical Learning. Vol. 1. — 2001. — (Ser.: Springer Series in Statistics; 10).

3.25. *Herbrich R.* Learning Kernel Classifiers: Theory and Algorithms. — Cambridge, MA, USA: MIT Press, 2001. — P. 384.

3.26. *Kaftannikov I. L.*, *Parasich A. V.* Problems of Training Set's Formation in Machine Learning Tasks // Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control & Radioelectronics. — 2016. — Vol. 16. — P. 15–24. — (In Russian).

3.27. *Kannan P.* Algorithm Names // Oracle Database PL/SQL Packages and Types Reference, 12c Release 1 (12.1). — 2016. — URL: https://docs.oracle.com/database/121/ARPLS/d_datmin.htm#ARPLS608.

3.28. *Kannan P.* DBMS_DATA_MINING // Oracle Database PL/SQL Packages and Types Reference, 12c Release 1 (12.1). — 2016. — URL: https://docs.oracle.com/database/121/ARPLS/d_datmin.htm#ARPLS192.

3.29. *Kannan P.* Mining Function Settings // Oracle Database PL/SQL Packages and Types Reference, 12c Release 1 (12.1). — 2016. — URL: https://docs.oracle.com/database/121/ARPLS/d_datmin.htm#CACFFEEF.

3.30. *Kannan P.* PREDICTION_PROBABILITY // Oracle Database PL/SQL Packages and Types Reference, 12c Release 1 (12.1). — 2016. — URL: https://docs.oracle.com/database/121/SQLRF/functions150.htm#SQLRF06212.

3.31. *Kashnitsky Y.* Classification and regression liner model. — 2017. — URL: https://habrahabr.ru/company/ods/blog/323890/ (visited on 10.02.2018); (In Russian).

3.32. *Kashnitsky Y. S.*, *Ignatov D. I.* Recommender-based Classifier Ensemble // Intellectual systems. Theory and applications. — 2015. — Vol. 19, no. 4. — P. 37–55.

3.33. *Kondratyev E. V.* Optimization of Radiation Exposure During CT Angiography of the Aorta and Peripheral Arteries // Meditsinskaya Vizualizatsiya. — 2012. — No. 3. — P. 41–50. — (In Russian).

3.34. *Labintcev E.* Metrics in the problems of machine learning. — 2017. — URL: https://habr.com/company/ods/blog/328372/ (visited on 10.04.2018); (In Russian).

3.35. *McCullagh P.*, *Nelder J. A.* Generalized Linear Models, Second Edition. — Taylor & Francis, 1989. — (Ser.: Chapman & Hall/CRC Monographs on Statistics & Applied Probability).

3.36. *Murphy K. P.* Machine Learning. — MIT Press Ltd, 2012. — 1104 p.

3.37. *Pampel F. C.* Logistic Regression: A Primer. — SAGE Publications, 2000. — P. 96. — (Ser.: Quantitative Applications in the Social Sciences).

3.38. *Quinlan J. R.* Induction of decision trees // Machine Learning. — 1986. — Vol. 1, no. 1. — P. 81–106.

3.39. *Rivest R. L.* Learning Decision Lists // Machine Learning. — 1987. — Vol. 2, issue 3. — P. 229–246.

3.40. *Singh A.*, *Thakur N.*, *Sharma A.* A review of supervised machine learning algorithms // 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). — 2016. — P. 1310–1315.

3.41. *Strang G.* Linear algebra and its applications. — Belmont, CA: Thomson, Brooks/Cole, 2006. — P. 487.

3.42. *Strijov V. V.* Error function in regression analysis // Factory Laboratory. — 2013. — Vol. 79(5). — P. 65–73. — (In Russian).

3.43. *Strijov V. V.* The methods for the inductive generation of regression models. — Moscow, Computing Center RAS, 2008. — URL: http://strijov.com/papers/strijov08ln.pdf; (In Russian).

3.44. *Stycenko I.* AgeHack - the first online hackaton to extend life on the platform MLBootCamp. — 2017. — URL: https://habr.com/company/mailru/blog/330960/ (visited on 05.04.2018); (In Russian).

3.45. *Vorontsov K. V.* Support vectors machine lectures. — 2007. — URL: http://www.ccas.ru/voron/download/SVM.pdf; (In Russian).

3.46. World Health Organization. Cardiovascular diseases (CVDs). — 2015. — URL: http://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds) (visited on 01.04.2018).

3.47. *Zagorujko N. G.* Prikladnye metody analiza dannyh i znanij. — Novosibirsk: IM SO RAN, 1999. — 270 p. — (In Russian).