Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической
кибернетики и компьютерных наук

АВТОМАТИЗИРОВАННАЯ МЕДИЦИНСКАЯ ДИАГНОСТИКА МАСКИРОВАННОЙ ФОРМЫ АРТЕРИАЛЬНОЙ ГИПЕРТЕНЗИИ У ПАЦИЕНТОВ МОЛОДОГО ВОЗРАСТА

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Неверовой Елены Андреевны

Научный руководитель
доцент кафедры ТП, к.ф.-м.н. _____ А. А. Кузнецов

Заведующий кафедрой
к.ф.-м.н. _____ С. В. Миронов

Саратов 2018

ВВЕДЕНИЕ

Информатизация российского здравоохранения — положительная тенденция последних лет, основанная на использовании преимуществ технического прогресса. Компьютерные технологии способны не только ускорить и упростить процесс взаимодействия пациентов с медицинским персоналом за счёт введения электронных очередей, регистратур и историй болезни, но и помочь врачам в диагностике заболеваний. Последнее заключается в появлении систем, помогающих врачу оперативно определить верное направление исследования состояния пациента и его последующего лечения. Такие системы разрабатываются на основе интеллектуального анализа статистических данных. Подобные программы применяются за рубежом, и развитие отечественной медицины невозможно без появления аналогов.

Автоматизированные диагностические системы требуются во всех областях медицины, в том числе в кардиологии. Одной из важных задач врачей-кардиологов является диагностика артериальной гипертензии на ранней стадии. Эта стадия носит название маскированной формы. Её выявление — чрезвычайно сложная задача, так как явных проявлений болезни, как правило, нет, и человек может не обращать внимания на её косвенные признаки.

Разделение пациентов на больных и здоровых — пример задачи классификации, одного из наиболее часто встречающихся типов задач интеллектуального анализа данных. Для её решения используются методы машинного обучения.

Целью настоящей бакалаврской работы является разработка алгоритма классификации, позволяющего выявлять больных с маскированной формой артериальной гипертензии.

Для достижения поставленной цели были определены следующие **задачи**:

- изучение предметной области;
- формулирование задачи классификации и изучение теоретических методов её решения;
- изучение способов анализа качества классификаторов;
- ознакомление с библиотеками языка программирования `Python`, реализующими алгоритмы машинного обучения и обеспечивающими работу с данными;

— разработка и реализация алгоритма классификации на языке программирования `Python`.

Структура и объём работы. Бакалаврская работа состоит из введения, трёх глав, заключения, списка использованных источников и приложения. Общий объём работы — 77 страниц, из них 58 страниц — основное содержание (включая 2 таблицы и 14 рисунков), 19 страниц приложения с программным кодом. Список использованных источников информации содержит 29 наименований.

В первой главе **«Исследование предметной области»** приводятся сведения, касающиеся медицинской диагностики маскированной артериальной гипертензии и скрининговых исследований.

Вторая глава **«Формальная постановка задачи классификации и подходы к её решению»** содержит важные сведения, касающиеся задачи классификации, решаемой в бакалаврской работе: приводится формальная постановка задачи и описание процесса её решения, выделяются его основные этапы, рассматриваемые в данной работе, подробно описываются теоретические основы использованных на этих этапах методов, а также инструментарий, использованный при разработке программ.

Третья глава **«Разработка алгоритма для автоматизированной диагностики маскированной артериальной гипертензии»** посвящена процессу разработки и написанным программам: описывается формат и источник исходных данных, конкретизируется задача классификации, положенная в основу работы, приводится описание всех разработанных программ, описываются суть и результаты каждого этапа разработки в отдельности, после чего кратко характеризуется весь процесс разработки и полученные результаты обобщаются.

В заключении подводятся итоги проделанной работы.

1 Исследование предметной области

Артериальная гипертензия (АГ) — синдром повышения систолического артериального давления (САД) ≥ 140 мм рт. ст. и/или диастолического артериального давления (ДАД) ≥ 90 мм рт. ст. Это хроническое заболевание является причиной сердечно-сосудистых и цереброваскулярных заболеваний, часто приводящих в преждевременной смерти [1].

1.1 Классификация артериальной гипертензии

Используя отечественную классификацию и опираясь на опыт зарубежных исследований, можно рассматривать упрощённую классификацию пациентов по уровням артериального давления: нормотоники, пациенты с маскированной или с манифестной (явной и ярко выраженной) формой артериальной гипертензии.

1.2 Маскированная артериальная гипертензия

В данной работе акцентируется внимание на маскированной артериальной гипертензии, так как это достаточно редкая и трудно диагностируемая форма АГ. Как правило, при данной форме артериальной гипертензии значения офисного АД (на приёме у врача) составляют $120 - 139$ мм рт. ст (САД) и $80 - 89$ мм рт. ст. (ДАД), но в других условиях показатели давления бывают выше данных значений.

Риск перерастания маскированной АГ в манифестную крайне высок [1].

1.3 Скрининг в медицине

Скрининг — комплекс мероприятий в сфере здравоохранения, направленный на активное выявление болезней или предболезненных состояний у лиц, считающих себя здоровыми.

Можно выделить два важных критерия оценки скрининговых тестов: чувствительность и специфичность [2].

Чувствительный диагностический тест проявляется в гипердиагностике, то есть в максимальном предотвращении пропуска больных. Специфичный диагностический тест проводится для выявления только доподлинно больных. Как правило, применяется, когда лечение больного связано с серьёзными побочными эффектами и гипердиагностика пациентов нежелательна [3].

2 Формальная постановка задачи классификации и подходы к её решению

2.1 Описание задачи классификации

Data Science — совокупность дисциплин, описывающих методы и процессы работы с данными разного вида (структурированными или неструктурированными) и происхождения для извлечения из них знаний. Как часть Data Science рассматривается работа с большими данными, методами Data Mining и машинного обучения [4].

Data Mining — интеллектуальный анализ данных любого объёма с целью нахождения практически полезной нетривиальной информации, позволяющей принимать решения в различных сферах человеческой деятельности; совокупность методов и инструментов для извлечения таких знаний.

Задача классификации — наиболее часто решаемая задача Data Mining, состоящая в следующем: дано множество объектов, разделённых на некоторые классы, причём только для части объектов известно, к какому классу они относятся. Используя данные из этой выборки, необходимо разработать алгоритм, который был бы способен определить классовую принадлежность произвольного объекта множества.

Множество объектов, для которых известны соответствующие классы, называется *обучающей выборкой*. Множество остальных объектов — это *тестовая выборка*.

Также определено множество признаков, по которым будет определяться классовая принадлежность. Признаковое описание — наиболее распространённый вид входных данных. Признаки могут быть как числовыми, так и нечисловыми [4, 5].

2.2 Процесс решения задачи классификации

Процесс решения задачи классификации в общем случае состоит из шести шагов:

1. постановка цели исследования;
2. сбор данных;
3. подготовка данных;
4. исследование данных;
5. построение модели (моделирование данных);

6. демонстрация результатов и автоматизация анализа (если требуется).

На практике линейное выполнение всех шагов встречается редко. Какие-то этапы зачастую приходится выполнять циклично [6].

Для решения задачи классификации необходимо отобрать значимые признаки (параметры), описывающие объекты входных данных, поэтому шаг исследования данных заключается в отборе нужных параметров, после чего осуществляется выбор модели, её обучение и анализ. Далее подробнее описаны подходы к реализации этих двух этапов, использованные в данной работе.

2.3 Методы отбора параметров

2.3.1 Алгоритм Apriori

Алгоритм `Apriori` — алгоритм определения ассоциативных правил. Ассоциативные правила позволяют находить закономерности между связанными событиями.

Алгоритмы поиска ассоциативных правил предназначены для нахождения всех правил $X \to Y$, поддержка и достоверность которых удовлетворяют заданным минимальным значениям поддержки (`minsupport`) и достоверности (`minconfidence`).

`Apriori` работает в два этапа: сначала необходимо найти часто встречающиеся наборы элементов, а затем извлечь из них правила, то есть сначала подсчитываются значения поддержки для одноэлементных наборов, а на следующем шаге генерируются потенциально часто встречающиеся наборы из большего числа элементов (кандидаты) и подсчитываются значения поддержки уже для них. Наборы переводятся в разряд часто встречающихся при условии удовлетворения `minsupport`. После того как найдены все часто встречающиеся наборы элементов, начинается генерация правил с достоверностью `minconfidence` [7].

2.3.2 Критерий хи-квадрат (критерий согласия Пирсона)

Критерий согласия — статистический критерий, в котором проверяется, согласуются ли данные выборки с выдвинутой гипотезой или опровергают её.

Статистика для критерия согласия Пирсона выглядит следующим образом:

$$\phi = \sum_{i=1}^{k} \frac{(n_i - np_i)^2}{np_i} = n \sum_{i=1}^{k} \frac{(n_i/n - p_i)^2}{p_i} \sim \chi^2(k - l - 1),$$

где p_i — вероятность попадания в i-й интервал.

Также необходимо определить уровень значимости α — вероятность допустимой ошибки при определении верной гипотезы.

Метод позволяет определить статистически значимые для постановки диагноза параметры.

По описанной выше формуле вычисляется $\chi^2_{\text{наблюдаемое}}$, а по математическим таблицам по заданному уровню значимости α и числу степеней свободы $k - l - 1$ находится $\chi^2_{\text{теоретическое}}$. Если $\chi^2_{\text{наблюдаемое}} \geq \chi^2_{\text{теоретическое}}$, то зависимость конкретного диагноза от рассматриваемого параметра будет статистически значимой.

2.4 Построение модели на основе логистической регрессии

2.4.1 Понятие регрессии

Регрессия применяется в случаях, когда необходимо определить, как наблюдаемая величина зависит от одной или нескольких других величин, если зависимость не функциональная. В этом случае можно считать среднее значение этой величины функцией от указанных величин.

Величины, от которых зависит наблюдаемая величина, называются *независимыми, объясняющими*, а она сама — *зависимой, или объясняемой*.

Пусть наблюдаемая величина Y зависит от случайной величины (случайного вектора) X. Зависимость среднего значения Y от значений X обозначается через $f(t)$ (t — значение случайного вектора X) [8].

2.4.2 Линейная регрессия

Линейная регрессия — регрессия, при которой оценка функции $f(t)$ является линейной.

Основная идея линейного классификатора — разбиение гиперплоскостью пространства признаков на две полуплоскости, каждая из которых соответствует прогнозу одного из двух значений целевого класса.

2.4.3 Логистическая регрессия

Логистическая регрессия — частный случай бинарной линейной регрессии.

Ключевой особенностью данной статистической модели является возможность прогнозирования не только ответа на поставленную задачу клас-

сификации, но и вероятности, с которой объект попадает в положительный класс.

В этом случае производится анализ связи между независимыми переменными (признаками задачи классификации) и зависимой переменной, принимающей два значения: 1 — событие произошло или 0 — событие не произошло.

Логистическая регрессия — модель бинарной классификации, поэтому все признаки также должны быть бинаризованы и количество классов, прогнозируемых в модели, должно быть сведено до двух, если имеет место многоклассовая классификация.

2.4.4 Мультиклассовая классификация

Существует два основных способа сведения многоклассовой задачи к бинарной:

— метод «каждый против остальных/всех» (`one-vs-rest` или `one-vs-all`): для каждого из n классов обучается классификатор, считающий данный класс «положительным», а все остальные — «отрицательными», затем для всех примеров вычисляются значения $f_1(x), ..., f_n(x)$, из которых выбирается максимальное, и ответом является соответствующая выбранному значению метка класса;

— метод «каждый против каждого» (`one-vs-one`).

Первый метод — наиболее часто используемая стратегия. В случае логистической регрессии вычисляется вероятность попадания в каждый из классов, и в качестве ответа выдаётся метка класса, для которого получена наибольшая вероятность [9].

2.4.5 Обобщающая способность модели и переобучение

Основное требование, предъявляемое к каждой обучаемой модели, — способность обобщаться, то есть хорошо работать на любых новых данных. С обобщающей способностью связана проблема *переобучения, или чрезмерной подгонки (*`overfitting`*)*, заключающаяся в учёте специфических особенностей обучаемой выборки (шумов). В результате переобучения модель начинает выявлять несуществующие закономерности.

Существует два способа обучения и проверки качества модели:

— использование отложенной выборки (`hold-out set`): все данные разбиваются на тестовую ($20 - 40\%$) и обучающую (оставшиеся $60 - 80\%$)

выборки;

— кросс-валидация (`cross-validation`), заключающаяся в проведении нескольких циклов разбиения данных на тестовую и обучающую выборки с последующим обучением модели.

Значимый недостаток первого способа — ограничение обучающей выборки, в которую могут не попасть важные для эффективной работы модели примеры. Для использования всех данных при обучении применяется кросс-валидация.

2.5 Анализ качества обученного классификатора

В данном разделе рассматриваются метрики и способы анализа качества работы бинарных классификаторов, которые в процессе «настройки» классификатора позволяют определить, стал ли он работать «лучше» или «хуже».

2.5.1 Матрица несоответствий

Предсказанный обученным классификатором класс может совпасть с истинным или нет. Ошибки классификации бывают двух видов: `False Negative` (ошибка первого рода — объект «положительного» класса неверно классифицирован как «негативный») и `False Positive` (ошибка второго рода — объект «отрицательного класса» неверно отнесён к «положительному» классу) [4].

Далее будут использованы следующие сокращения:

— `TP` — количество `True Positive` результатов;

— `TN` — количество `True Negative` результатов;

— `FP` — количество `False Positive` результатов;

— `FN` — количество `False Negative` результатов.

Используемые в работе метрики (характеристики) бинарного классификатора:

— **Полнота, или чувствительность (**`True Positive Rate` **— истинно положительные результаты)**, показывает, какую часть от реального числа «положительных» объектов составляют все предсказания «положительного» класса:

$$TPR = \frac{TP}{TP + FN}.$$

— **Ложно положительные результаты (**`False Positive Rate`**,** `FPR`**)** показывают, какая часть реальных «отрицательных» объектов были неверно

классифицированы как «положительные»:

$$FPR = \frac{FP}{FP + TN}.$$

— **Специфичность (**`specificity`**)** показывает, какая часть предсказаний об «отрицательных» объектах была сделана верно:

$$Specificity = \frac{TN}{TN + FP}.$$

Специфичность можно вычислять как $1 - FPR$ [3, 6].

Метрики чувствительности и специфичности полностью соответствуют показателям, используемым в медицине для диагностики заболеваний.

2.5.2 ROC-кривая и AUC

`ROC`-кривая — график, показывающий зависимость полноты, или истинно положительных результатов (`TPR`), от ложно положительных результатов (`FPR`) при варьировании порогового значения, определяющего границу двух классов.

Порог (`threshold`) в модели логистической регрессии принимает значения в диапазоне от 0 до 1. Пример получает метку положительного класса, если вероятность его принадлежности к этому классу больше или равна пороговому значению, и метку отрицательного класса в противном случае.

`AUC` (`area under curve`) — площадь под графиком `ROC`-кривой. Чем больше это значение, тем качественнее модель. `AUC` — критерий сравнения нескольких моделей.

2.6 Проблема несбалансированности данных

При решении задачи классификации может возникнуть проблема несбалансированности данных: объектов одного или нескольких классов, которые наиболее важно выделять, значительно меньше, чем объектов других классов.

Подходы к решению описанной проблемы:

— `undersampling` — сокращение количества объектов больших классов (самый простой вариант — случайное удаление заданного количества примеров преобладающего класса);

— `oversampling` — дублирование объектов маленьких классов для выравнивания размеров классов [10, 11].

2.7 Применение машинного обучения при решении задачи классификации

В типовом процессе Data Science машинное обучение, как правило, применяется на этапе построения модели. Но следует заметить, что его методы могут быть полезны также при подготовке и исследовании данных [6].

В настоящее время наиболее популярный инструмент решения задач Data Science — язык программирования Python и его библиотеки со встроенными алгоритмами машинного обучения и удобными структурами данных для представления выборок.

Для работы с данными были использованы следующие библиотеки:

— pandas: предоставляет большие возможности для представления данных (в данной работе использовалась структура данных DataFrame);
— matplotlib: пакет двухмерной графики для визуализации (был использован модуль pyplot для отображения ROC-кривых);
— NumPy: используется для научных расчётов;
— SciPy: интегрирует популярные фундаментальные пакеты, используемые в научных целях (NumPy, matplotlib, pandas);
— csv: формат файлов CSV удобен для представления исходных данных и сохранения полученной в процессе исследования информации;
— json: используется для сериализации структур данных.

Алгоритмы машинного обучения реализованы в библиотеке scikit-learn. Были использованы следующие возможности scikit-learn:

— модель логистической регрессии (класс LogisticRegression модуля linear_model);
— функция бинаризации меток классов (label_binarize модуля preprocessing);
— класс StratifiedKFold модуля model_selection для проведения кросс-валидации модели с помощью k блоков и стратификации;
— функции roc_curve и auc модуля metrics для построения ROC-кривых и получения значений AUC;
— библиотека joblib модуля externals для сохранения обученной модели в файл формата PKL и последующей загрузки в другие программы.

Для решения проблемы несбалансированности данных был использован класс RandomUnderSampler библиотеки imblearn, реализующий стратегию случайного удаления объектов преобладающего класса.

3 Разработка алгоритма для автоматизированной диагностики маскированной артериальной гипертензии

3.1 Проект по разработке программы расчёта риска маскированной артериальной гипертензии

Рассматриваемая задача диагностики маскированной формы артериальной гипертензии предложена врачами НИИ кардиологии города Саратова. Ими были предоставлены результаты обследования пациентов.

Интерес врачей НИИ кардиологии направлен на получение возможности прогнозирования маскированной формы АГ у лиц молодого возраста (в возрастной группе от 17 до 35 лет). Это связано с возможностью ранней диагностики АГ, своевременного лечения и корректировки образа жизни, что может предотвратить или по крайней мере отсрочить развитие явной артериальной гипертензии.

Программное обеспечение для диагностики маскированной артериальной гипертензии разрабатывается совместно с Череваткиным А. В.

3.2 Описание задачи классификации и её решения

Задача классификации, положенная в основу проекта, состояла в определении принадлежности конкретного пациента к одному из 2 классов:

- **ОК** — класс нормотоников;
- **МАГ** — класс пациентов, у которых выявлена маскированная артериальная гипертензия.

В решении описанной задачи можно выделить два основных этапа, каждый из которых состоит из нескольких шагов:

- отбор параметров для будущего классификатора:
  - формирование параметров из значений факторов, представленных в исходных данных;
  - формирование пар параметров;
  - непосредственно отбор пар параметров по двум критериям;
- разработка алгоритма классификации:
  - преобразование исходной базы данных к бинарному виду;
  - обучение и анализ классификатора;
  - применение классификатора для расчёта риска наличия маскированной формы болезни.

При решении задачи классификации учитывалась несбалансированность данных выборки: объекты класса МАГ составляют лишь 11% от всех объектов классов МАГ и ОК.

Для решения поставленной задачи был написан ряд программ на языке программирования `Python 3.6` с использованием описанных выше библиотек.

3.3 Отбор параметров

Перед обучением модели классификатора прежде всего необходимо определить набор статистически значимых параметров.

Обследование включало в себя 254 фактора (пункты медицинского опросника). В самом начале работы был произведён отбор наиболее значимых факторов. Было оставлено 30 факторов.

При реализации первой тестовой версии классификатора было выяснено, что большие помехи в работу классификатора вносят лишние биохимические признаки, которые были получены при рассмотрении выборки, состоящей из молодых людей младше 21 года. Среди пациентов этой группы результаты биохимии присутствовали только у больных АГ (здоровые люди не сдавали анализы). В связи с этим используется два типа выборки: все пациенты классов МАГ и ОК (`full`), для которых не рассматриваются биохимические факторы, или пациенты этих классов старше 20 лет (`bounded`), анализируемые по полному списку факторов.

3.3.1 Представление параметров

Для удобства работы для представления параметров задачи классификации был написан класс `Parameter`.

Рассматриваемые параметры могут быть нескольких видов:

- **нижняя граница (`lower`)**, то есть можно сделать вывод, что пациенту соответствует данный параметр, если его значение больше либо равно данной границе;
- **верхняя граница (`upper`)**, то есть у пациента с этим параметром значение меньше либо равно данной границе;
- **NaN** (отсутствие параметра).

3.3.2 Создание параметров для всех факторов

Для создания списка параметров, его сохранения и загрузки в программу было написано несколько функций. Для каждого вида выборки создаётся свой список параметров.

3.3.3 Формирование списков пар параметров

Идея использования пар параметров основана на первом этапе алгоритма `Apriori`. Так как классы не сбалансированы, для них устанавливаются различные значения минимальной поддержки.

Результат работы скрипта, реализующего составление пар, для обоих значений типа выборки — два набора файлов по числу рассматриваемых записей (пациентов) в исходной базе данных, содержащих списки пар параметров, соответствующих каждому пациенту.

3.3.4 Фильтрация списка пар параметров с помощью критерия хи-квадрат

Для эффективной работы будущего классификатора необходимо отобрать статистически значимые пары параметров, влияющие на постановку диагноза. Это позволяет сделать критерий хи-квадрат (критерий согласия Пирсона).

Результат выполнения скрипта, реализующего фильтрацию списка пар, для обоих значений типа выборки — два файла со списками пар параметров, которые после конвертации будут использоваться для анализа результатов этапа отбора параметров и обучения классификатора.

3.3.5 Преобразование отфильтрованного списка пар параметров

Скрипт преобразования отфильтрованных списков пар параметров позволяет получить 2 файла формата `JSON` с парами параметров в виде списка списков словарей для использования на других этапов. Кроме того, для анализа результатов отбора пары параметров сохраняются в файл формата `CSV`, где указывается вся необходимая информация.

3.4 Обучение модели логистической регрессии

В качестве модели для будущего классификатора был выбран метод логистической регрессии, так как её можно применять при небольшом объёме входных данных и она не должна сильно переобучаться. Для уменьшения

данного эффекта, а также для решения проблемы несбалансированности данных была применена кросс-валидация со стратификацией StratifiedKFold. Также для борьбы с преобладанием примеров здоровых людей был применён метод сокращения размера класса ОК, реализованный в RandomUnderSampling библиотеки imblearn.

3.4.1 Обработка данных для последующего обучения модели

Для работы модели логистической регрессии все признаки должны быть представлены в бинарном виде, поэтому перед её обучением необходимо провести предварительную обработку входного файла.

Необходимо сформировать два новых файла формата CSV, в которых столбцы — пары отобранных параметров, а строки — данные пациентов, представленные в бинарном виде: если значения соответствующих столбцов из исходного файла удовлетворяют обоим параметрам пары, в столбец этой пары записывается единица, если не удовлетворяют хотя бы одному — ноль.

Результат работы скрипта, реализующего указанное преобразование, для обоих вариантов пар параметров — два CSV-файла, содержащих два варианта бинаризованной базы данных, которые будут использованы при обучении модели логистической регрессии.

3.4.2 Обучение модели и анализ качества полученного классификатора

Результат работы скрипта, реализующего обучение и анализ качества модели, для обоих вариантов преобразованной базы данных пациентов — два файла формата PKL с обученными на разных входных файлах классификаторами, настроенными на выявление маскированной формы артериальной гипертензии, два JSON-файла со средними пороговыми значениями, два JSON-файла с соответствующими им средними значениями ложно позитивных предсказаний и два изображения с ROC-кривыми (проверка работы обоих классификаторов).

3.5 Алгоритм выявления больных маскированной формой артериальной гипертензии

В результате обсуждения результатов обучения моделей с врачами из НИИ кардиологии было решено, что скрининговый тест должен обладать вы-

сокой специфичностью, то есть выявлять только доподлинно больных маскированной формой артериальной гипертензии.

Для определения риска наличия у пациента маскированной артериальной гипертензии по вероятностям, предсказываемым обученными классификаторами, пороговым значениям и соответствующим им количествам ложно позитивных предсказаний, полученным в процессе обучения моделей, вычисляется среднее арифметическое значение специфичности.

3.6 Обобщение полученных результатов

Решение описанной задачи классификации осуществлялось в два этапа:

1. отбор параметров;
2. обучение модели классификатора и оценка его качества.

Данные этапы тесно связаны: чтобы обучить модель, необходимо отобрать значимые параметры, но чтобы получить эффективный классификатор, необходимо понять, почему предыдущие версии работали хуже, а это часто требует внесения изменений в стратегии формирования и отбора параметров. Таким образом, разработка алгоритма классификации велась итеративно.

На протяжении всей работы были неизменны следующие подходы:

— представление данных пациентов в виде пар параметров;
— применение для обучения модели логистической регрессии;
— использование двух подвыборок исходной выборки и, соответственно, двух классификаторов.

В первом прототипе были реализованы многоклассовая классификация и представление параметров как интервалов значений. После тестирования на данных, не входящих в исходную выборку, от многоклассовой классификации пришлось отказаться по причине отрицательного влияния лишнего класса манифестной артериальной гипертензии на диагностирование маскированной формы болезни. Кроме того, интервалы оказались слишком узкими.

Для второго прототипа решалась задача бинарной классификации с целью разделения классов здоровых людей и больных маскированной формой АГ, а параметры представлялись как нижние границы, верхние границы или значение `NaN`.

В результате работы был реализован алгоритм, с помощью обученных классификаторов вычисляющий риск наличия у пациента маскированной формы артериальной гипертензии в виде среднего значения специфичности.

ЗАКЛЮЧЕНИЕ

В рамках данной дипломной работы решалась задача бинарной классификации, заключающаяся в разделении пациентов на два класса: класс здоровых людей и класс людей с маскированной формой артериальной гипертензии.

Подготовка к решению этой задачи включала в себя изучение литературы по маскированной артериальной гипертензии и медицинскому скринингу, а также консультации с врачами-кардиологами НИИ кардиологии города Саратова, что позволило определить основные факторы, влияющие на развитие болезни у пациентов исследуемой выборки, и стратегию диагностики.

Решение задачи классификации осуществлялось в два этапа: отбор параметров и обучение модели классификатора, включая оценку его качества. Данные этапы взаимосвязаны, поэтому разработка алгоритма классификации велась итеративно.

Этап отбора параметров заключался в применении части алгоритма поиска ассоциативных правил `Apriori` и статистического критерия хи-квадрат. Признаки пациентов, получаемые после его завершения, представляют собой пары параметров трёх видов, среди которых могут встречаться пары из одного и того же параметра. На этапе отбора параметров формировалось два набора пар, по которым затем строились два классификатора.

На втором этапе для получения классификаторов была выбрана модель логистической регрессии. Её обучение велось с помощью `k-fold` кросс-валидации со стратификацией с целью снижения эффекта переобучения. Также был использован метод случайного удаления объектов преобладающего класса для решения проблемы несбалансированности данных.

В результате дипломной работы был разработан и реализован алгоритм, с помощью обученных классификаторов вычисляющий риск наличия у пациента маскированной формы артериальной гипертензии в виде значения специфичности. Данный алгоритм положен в основу диагностической системы, применяющейся для расчёта индивидуального риска маскированной артериальной гипертензии у лиц молодого возраста.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Чазова, И. Е.* Клинические рекомендации. Диагностика и лечение артериальной гипертонии / И. Е. Чазова, Е. В. Ощепкова, Ю. В. Жернакова // *Кардиологический вестник*. — 2015. — № 1. — С. 3–30.

2 *Холланд, В.* Основы политики. Скрининг в Европе / В. Холланд, С. Стюарт, К. Масссеря. — ВОЗ, 2008.

3 Логистическая регрессия и ROC-анализ — математический аппарат [Электронный ресурс]. — URL: `https://basegroup.ru/community/articles/logistic` (Дата обращения 11.05.2018). Загл. с экр. Яз. рус.

4 *Swamynathan, M.* Mastering Machine Learning with Python in Six Steps / M. Swamynathan. — Apress, 2017.

5 *Дюк, В.* Data Mining: учебный курс / В. Дюк, А. Самойленко. — СПб.: Питер, 2001.

6 *Силен, Д.* Основы Data Science и Big Data. Python и наука о данных / Д. Силен, А. Мейсман, Али М. — СПб.: Питер, 2017.

7 Apriori — масштабируемый алгоритм поиска ассоциативных правил [Электронный ресурс]. — URL: `https://basegroup.ru/community/articles/apriori` (Дата обращения 08.05.2018). Загл. с экр. Яз. рус.

8 *Чернова, Н. И.* Математическая статистика / Н. И. Чернова. — Новосибирск: РИЦ НГУ, 2014.

9 scikit-learn [Электронный ресурс]. — URL: `http://scikit-learn.org/` (Дата обращения 11.05.2018). Загл. с экр. Яз. англ.

10 Несбалансированные данные [Электронный ресурс]. — URL: `https://ru.coursera.org/learn/supervised-learning/lecture/M97UX/niesbalansirovannyie-dannyie` (Дата обращения 18.05.2018). Загл. с экр. Яз. рус.

11 Различные стратегии сэмплинга в условиях несбалансированности классов [Электронный ресурс]. — URL: `https://basegroup.ru/community/articles/imbalance-datasets` (Дата обращения 18.05.2018). Загл. с экр. Яз. рус.