МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра информационных систем управления

КЛИЦУНОВА Екатерина

АЛГОРИТМЫ ОБРАБОТКИ НЕСБАЛАНСИРОВАННЫХ ДАННЫХ

Дипломная работа

Научный руководитель:
кандидат технических наук,
доцент М.М. Лукашевич

Допущена к защите
«____»_____20___г.
Заведующий кафедрой информационных систем управления
доктор технических наук, доцент А. М. Недзьведь

Минск, 2025

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ

ADASYN – адаптированный алгоритм синтетического увеличения данных (Adaptive Synthetic).

CNN – алгоритм уменьшения большего класса по правилу сжатия ближайших соседей (Condensed Nearest Neighbors)

CSV – текстовый формат, предназначенный для представления табличных данных (Comma-Separated Values)

ENN – алгоритм уменьшения большего класса на основе измененного алгоритма k-ближайших соседей (Edited Nearest Neighbor)

FN – количество случаев, когда объектам положительного класса ошибочно присваиваются метки отрицательного класса (False Negative).

FP – количество случаев, когда объектам отрицательного класса ошибочно присваиваются метки положительного класса (False Positive).

NCR – алгоритм уменьшения большего класса по правилу удаления соседей (Neighborhood Cleaning Rule)

OSS – алгоритм уменьшения большего класса методом одностороннего выбора (One-Sided Selection)

Random Oversampling – алгоритм случайного увеличения данных

Random Undersampling – алгоритм случайного удаления данных

SMOTE – алгоритм синтетического увеличения данных (Synthetic Minority Oversampling Technique)

SVM – машина опорных векторов (Support Vector Machine)

TN – количество случаев, когда модель корректно идентифицировала отрицательный класс (True Negative).

TP – количество случаев, когда модель корректно идентифицировала положительный класс (True Positive).

РЕФЕРАТ

Структура и объём дипломной работы

49 страниц, 16 рисунков, 8 таблиц, 5 приложений, 26 источников

Ключевые слова: НЕСБАЛАНСИРОВАННЫЕ ДАННЫЕ, МЕТОДЫ БАЛАНСИРОВКИ ДАННЫХ, МАШИННОЕ ОБУЧЕНИЕ, КЛАССИФИКАЦИЯ, ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ, РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА, ВЕБ-ПРИЛОЖЕНИЕ

Объект исследования – алгоритмы и методы машинного обучения, предназначенные для работы с несбалансированными наборами данных.

Предмет исследования – влияние методов балансировки данных на качество моделей классификации при несбалансированности данных.

Цель исследования – провести экспериментальное исследование влияния методов балансировки данных на эффективность классических моделей классификации. Разработать приложение, рекомендующее алгоритмы балансировки в зависимости от размера набора данных и величины дисбаланса, и дающее возможность провести балансировку данных.

Методы исследования: сравнительный анализ алгоритмов для работы с несбалансированными данными, постановка и реализация эксперимента, проектирование модели ранжирования для рекомендации алгоритмов балансировки данных, проектирование и разработка приложения.

Полученные результаты и их новизна: оценено влияние методов балансировки данных и подбора гиперпараметров на качество модели в задачах классификации. Разработано веб-приложение с моделью рекомендаций методов балансировки. Новизна работы заключается в интеграции алгоритма подбора методов балансировки с практической реализацией программного инструмента, позволяющего применять стандартные методы балансировки данных.

Достоверность материалов и результатов дипломной работы. Автор работы подтверждает, что приведенный в ней расчетно-аналитический материал правильно и объективно отражает состояние объекта исследования. Все заимствованные из источников теоретические, методологические и методические положения и концепции сопровождаются ссылками на их авторов.

Область практического применения: оптимизация выбора подходящих методов работы с несбалансированными данными для конкретных практических задач в сферах, где данная проблема является критической: медицине, биоинформатике, инженерии, безопасности, бизнесе и других.

_____

(подпись студента)

SUMMARY

Structure and scope of the diploma work
49 pages, 16 figures, 8 tables, 5 appendixes, 26 references
Keywords: IMBALANCED DATA, DATA BALANCING METHODS, MACHINE LEARNING, CLASSIFICATION, EXPERIMENTAL STUDY, RECOMMENDATION SYSTEM, WEB APPLICATION

The object of the research – algorithms and methods of machine learning designed to work with imbalanced datasets.

The subject of the research – the influence of data balancing methods on the quality of classification models trained with imbalanced datasets.

The aim of the research is to conduct an experimental study of the influence of data balancing methods on the quality of classical classification models, and to develop a software tool that recommends balancing algorithms depending on the size of the dataset and the imbalance ratio, while also enabling data balancing.

Research methods – comparative analysis of algorithms for working with imbalanced data, setup and implementation of the experiment, results interpretation, design of a machine learning model for recommending data balancing algorithms, design and development of an application.

The results of the work and their novelty – the study assessed the impact of data balancing methods and hyperparameter tuning on the quality of classification models. A web application with a model, which recommends balancing methods, has been developed. The novelty of the work lies in integrating an algorithm for selecting balancing methods with the practical implementation of a software tool that allows the use of standard data balancing methods.

Authenticity of the materials and results of the diploma work. The author confirms that the analytical and computational material presented in this work it correctly and objectively reflects the state of the research object. All theoretical, methodological and methodological provisions and concepts borrowed from literary and other sources are accompanied by references to their authors.

Recommendations on the usage*:* optimization of the process of selecting suitable methods and algorithms for working with imbalanced data for specific practical problems in areas where this problem is critical such as medicine, bioinformatics, engineering, security, business, and others.

_____

(student's signature)

ВВЕДЕНИЕ

В практических задачах машинного обучения в настоящее время часто встречаются несбалансированные наборы данных – такие, где количество объектов одного класса существенно преобладает над количеством объектов остальных классов [2].

Наиболее характерна эта проблема для задач классификации. При этом классические алгоритмы машинного обучения рассчитаны на работу со сбалансированными данными, из-за чего на несбалансированном наборе работают гораздо менее эффективно. Это становится особенно важным в тех задачах, где крайне важно получить результаты без смещения в пользу большего класса – например, не пропустить мошеннические действия или серьезное заболевание у пациента [2].

Цель данной дипломной работы – выполнить экспериментальное исследование влияния методов балансировки данных на эффективность классических моделей классификации. По результатам исследования необходимо разработать приложение, которое будет рекомендовать алгоритмы балансировки в зависимости от размера набора данных и величины дисбаланса, а также давать возможность провести балансировку данных.

Для достижения цели работы необходимо решить следующие задачи:

1) провести обзор методов работы с несбалансированными данными;
2) определить корректные метрики оценки качества моделей классификации в условиях дисбаланса классов;
3) провести экспериментальные исследования и проанализировать полученные результаты;
4) разработать программное средство, используя результаты исследования.

Необходимость разработки темы обусловлена широким разнообразием методов балансировки данных. Результат их сравнения и разработанное приложение может применяться для экономии ресурсов при выборе подходящего метода для конкретных практических задач.

В главе 1 приводится описание основных понятий предметной области, анализ литературы и научных публикаций предметной области. Глава 2 посвящена экспериментальным исследованиям влияния балансировки данных и подбора гиперпараметров на эффективность моделей, а также подбору оптимального соотношения классов после балансировки. В главе 3 описана разработка приложения с использованием полученных результатов исследования.

ГЛАВА 1
ПРОБЛЕМА НЕСБАЛАНСИРОВАННЫХ ДАННЫХ В МАШИННОМ ОБУЧЕНИИ

В данной главе рассмотрены основные теоретические аспекты проблемы несбалансированных данных: обзорно описываются основные понятия, причины возникновения несбалансированности в наборах данных и задачи, для которых такой дисбаланс характерен в силу особенностей предметной области. Также проведена систематизация алгоритмов и методов работы с несбалансированными данными и подобраны корректные метрики для оценки качества моделей.

Полученные результаты будут использованы для дальнейшего экспериментального исследования, в котором анализируется, как методы балансировки данных влияют на итоговое качество классификации.

1.1 Актуальность проблемы несбалансированных данных

Проблема несбалансированных данных не является новой: первые публикации на эту тематику датируются 1997 годом [17]. Однако она не является и до конца изученной: ее актуальность подтверждается растущим количеством публикаций в этой области в последние годы.

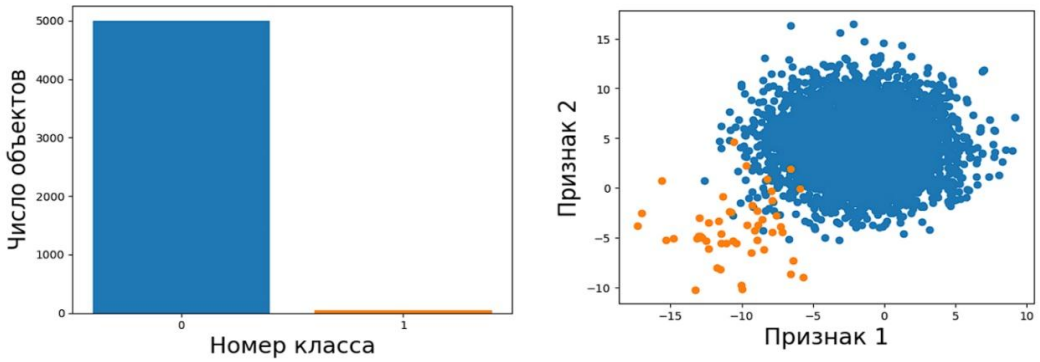
1.1.1 Основные понятия

Набор данных называют несбалансированным, если «распределение объектов по известным классам является неравномерным» [7, с. 2]. Несмотря на то, что в научной литературе отсутствует единое соглашение по поводу критериев классификации степени дисбаланса, курс машинного обучения от Google [12] предлагает следующий подход: если объекты меньшего класса составляют от 20% до 40% от общего числа, то дисбаланс считается слабым; от 1% до 20% – умеренным; менее 1% – сильным.

Обучение моделей с применением классических алгоритмов машинного обучения на несбалансированных данных связано со значительными трудностями: дисбаланс классов приводит к существенному ухудшению качества моделей, поскольку классические алгоритмы рассчитаны на приблизительное равномерное распределение объектов по классам [19].

Определить наличие дисбаланса в наборе данных можно различными способами. Универсальным является подсчет соотношения между наименьшим и наибольшим классом. Более наглядным будет метод построения гистограммы,

на которой визуально отображается количество объектов для каждого класса [7]. Также при условии небольшой размерности пространства признаков можно использовать графическую визуализацию распределения объектов. На рисунке 1.1 представлена визуализация вторым и третьим методом искусственно сгенерированного набора данных с соотношением объектов различных классов 1:100.



а б

а – гистограмма с количеством объектов, б – график с визуализацией объектов

Рисунок 1.1 – Визуализация набора данных с балансом классов 1:100

Множество литературы посвящено проблеме бинарной классификации для несбалансированных данных, однако несбалансированность характерна и для мультиклассовых наборов данных. В таком случае количество объектов одного или нескольких классов значительно меньше, чем остальных.

1.1.2 Причины несбалансированности данных

Причины возникновения дисбаланса в распределении классов в наборе данных, согласно [7, 9], можно разбить на следующие категории.

- *специфика предметной области* – некоторые события или состояния по своей природе являются редкими, к примеру, случаи редких заболеваний или мошеннических действий в сети;
- *ошибки сбора данных и неправильная разметка классов* – подобные ошибки могут возникнуть из-за человеческого фактора, неправильной настройки или несовершенства оборудования;
- *смещенная выборка* – к примеру, данные были собраны из узкого географического региона или отрезка времени, что приводит к искажению распределения классов.

Проблемы сбора данных или смещенной выборки могут быть решены повторным сбором данных и разметкой. Однако в некоторых случаях такой способ требует временных, финансовых и ресурсных затрат, что является нецелесообразным и приводит к необходимости работать с имеющимся набором данных.

Это же справедливо и для тех наборов данных, где несбалансированность возникла из-за специфики предметной области.

1.1.3 Примеры задач с несбалансированными данными

Первый пример задачи с несбалансированными данными в научной литературе приводится в работе [17], опубликованной в 1997 году, и связанных с ней публикациях, где авторами рассматривалась проблема распознавания разливов нефти по спутниковым снимкам.

С момента выявления проблемы несбалансированности данных ей было уделено значительное внимание в научных исследованиях. Согласно статистике, приведенной в публикации [9], большинство исследовательских работ в данной области, направленных на решение практических задач, охватывают такие сферы как медицина, биоинформатика, информационные технологии, безопасность, бизнес и инженерия.

В *медицине* наиболее характерными являются задачи по контролю качества – прогнозированию состояния пациентов после проводимого лечения или оперативного вмешательства; а также диагностированию и прогнозированию заболеваний. В *биоинформатике* наиболее изучаемой областью является идентификация микро-РНК, прогнозирование локализации белка, классификация белков и митотических клеток [9].

Для *сферы безопасности* характерны задачи биометрической аутентификации, обнаружения мошеннических действий и транзакций, подозрительных рассылок на электронные почты. Смежные проблемы для *сферы безопасности* и *бизнеса* включают обнаружение мошеннических действий, принятие решения по обслуживанию в банке и по выдаче кредитных средств [9]. Тем не менее, стоит отметить, что ситуации, в которых может выявиться несбалансированность данных, не ограничиваются перечисленными направлениями.

1.2 Методы и алгоритмы работы с несбалансированными данными

Вскоре после выявления проблемы несбалансированности наборов данных появился ряд исследований на эту тематику. Одними из первопроходцев можно назвать публикации [13, 24]. При этом многие методы возникали в процессе работы над реальными задачами, а не в процессе теоретического моделирования, но значительное внимание было уделено и систематизации уже разработанных методов и алгоритмов [11, 15, 23, 25]. В том числе появились и публикации, нацеленные на экспериментальное сравнение методов и алгоритмов работы с несбалансированными данными в зависимости от размера набора данных и величины

дисбаланса [8]. Классификация методов и алгоритмов работы с несбалансированными данными, основанная на работах [7, 16], приводилась в опубликованной мною ранее статье [2].

Предобработка данных. Такой подход направлен на обработку исходного набора данных (увеличение меньшего класса или уменьшение большего класса) для того, чтобы снизить дисбаланс до обучения модели.

Модификация алгоритмов. Этот подход нацелен на адаптацию классических алгоритмов классификации к работе с несбалансированными данными. Примером таких модификаций может служить взвешивание классов или добавление затрат на неправильную классификацию.

Ансамблевые алгоритмы. Применение ансамблей на классических алгоритмах позволяет добиться улучшения качества за счет объединения результата работы нескольких классификаторов, или последовательного исправления ошибок предыдущих классификаторов в ансамбле.

Комбинированные методы. Объединение вышеуказанных подходов позволяет получить модель с наилучшим качеством классификации. Наиболее широко используется комбинация из предобработки данных и использования ансамблевых алгоритмов [7], однако можно объединить и модифицированные алгоритмы в ансамбль.

Каждая из вышеупомянутых категорий работы с несбалансированными данными подробно рассмотрена далее, также включены основные методы, применяемые в каждой из них.

Выбор метода обуславливается оценкой временных и ресурсных затрат на подготовку данных или модификацию алгоритма.

Также для некоторых несбалансированных данных характерны сопутствующие проблемы, которые осложняют задачу классификации и накладывают ограничения на применяемые методы. Одна из проблем – малый набор данных. В таком случае уменьшение количества объектов большего класса может значительно ухудшить результаты классификации. Другая проблема – плохая разделимость или неразделимость классов. Если объекты разных классов смешаны в пространстве признаков, то это осложняет поиск границы решения [7].

1.2.1 Методы балансировки данных

Методы балансировки данных относятся к категории подходов на уровне данных и по сути являются частью предобработки исходного набора данных перед обучением на нем модели. Предобработка заключается в уменьшении дисбаланса тремя основными методами: увеличением количества объектов меньшего класса, уменьшением количества объектов большего класса и комбинированием обоих подходов [2]. Рассмотрим подробнее каждый из этих методов.

1) Увеличение меньшего класса

Методы, основанные на увеличении меньшего класса, особенно актуальны для небольших наборов данных, поскольку уменьшение большего класса может привести к значительной потере информации и снижению качества классификации вследствие этого.

Random Oversampling «случайным образом дублирует объекты меньшего класса» [7, с. 113].

SMOTE «выбирает объекты, которые близки в пространстве признаков, проводит линию между примерами в пространстве признаков и создает новый как точку вдоль этой линии» [7, с. 122]. Этот метод является самым популярным и широко используемым на практике.

Также существуют следующие его модификации: алгоритмы, создающие объекты на границе решения (Borderline-SMOTE, Borderline-SMOTE SVM), основываясь на идее, что дополнительные объекты нужны именно в этой области; и алгоритмы, создающие объекты с учетом плотности объектов меньшего класса (ADASYN).

Borderline-SMOTE «выбирает те объекты меньшего класса, которые неправильно классифицированы, например, моделью k-ближайших соседей, и затем создает новые только для этих объектов» [7, с. 130].

Borderline-SMOTE SVM использует «SVM вместо алгоритма k-ближайших соседей для выявления неправильно классифицированных примеров на границе решения» [7, с. 132].

ADASYN «предназначен для создания большего числа синтетических объектов в областях пространства признаков, где плотность объектов меньшего класса низкая, и меньше или нисколько там, где плотность высокая» [7, с. 134].

Следует отметить, что применение методов увеличения меньшего класса может привести к переобучению модели. Кроме того, оптимальной стратегией не всегда является достижение равенства между классами – в некоторых задачах может быть достаточно уменьшить дисбаланс до умеренного или легкого, чтобы избежать переобучения [2].

2) Уменьшение большего класса

Методы, основанные на уменьшении большего класса, не всегда приводят к улучшению качества моделей, и даже могут способствовать ухудшению из-за недообучения модели. Однако их стоит рассмотреть из-за того, что в комбинации с методами увеличения меньшего класса они могут значительно улучшить эффективность модели.

Методы уменьшения большего класса делят на три категории в зависимости направленности: выбор объектов для сохранения, выбор объектов для удаления, и объединение этих подходов [7, с. 108-109].

К алгоритмам, направленным на выбор объектов для сохранения, относят CNN и Near Miss.

CNN «ищет подмножество выборки, которое не приводит к потере качества модели» [7, с. 147]. Алгоритм оставляет все те объекты, которые необходимы для правильной классификации всех объектов в исходном наборе данных.

Методы семейства *Near Miss* «используют алгоритм k-ближайших соседей для выбора объектов большего класса» [7, с. 147]. NearMiss-1 сохраняет объекты с минимальным расстоянием к трем ближайшим объектам меньшего класса, NearMiss-2 – с минимальным расстоянием к трем наиболее удаленным объектам меньшего класса, а NearMiss-3 – с минимальным расстоянием к каждому объекту меньшего класса [7, с. 143].

К алгоритмам, направленным на выбор объектов для удаления, относят Random Undersampling, TomekLinks и ENN.

*Random Undersampling «*случайным образом удаляет объекты большего класса» [7, с. 113].

Tomek Links находит «пару объектов, которые являются взаимными ближайшими соседями (то есть между ними наблюдается минимальное расстояние в пространстве признаков) и принадлежат разным классам» [7, с. 150]. Предполагается, что объекты большего класса с таким расстоянием являются или граничными, или шумом, и их следует удалить [13].

ENN «подразумевает использование метода *3* ближайших соседей для поиска тех объектов в наборе данных, которые классифицированы неправильно, и затем удаляются» [7, с. 152].

К алгоритмам, направленным на выбор как объектов для удаления, так и объектов для сохранения, относят OSS и NCR.

OSS комбинирует алгоритмы Tomek Links для удаления шума и граничных объектов большего класса, и CNN для удаления объектов, далеких от границы решения [7, с. 155].

NCR комбинирует алгоритмы CNN для удаления избыточных объектов большего класса и ENN для удаления шума [7, с. 158].

3) Комбинированные методы предобработки

Сочетание алгоритмов увеличения меньшего класса и уменьшения большего класса может привести к улучшению качества модели по сравнению с тем, как если бы использовался только один из алгоритмов.

Для комбинированных методов зачастую выбирают те алгоритмы, которые показывают наилучшие результаты для модели изолированно. На практике [7, с. 109] широко используются сочетания SMOTE с Random Undersampling, TomekLinks и ENN.

При этом синтетическое увеличение данных проводится не до количества объектов в большем классе, а до половины, чтобы после второго этапа

комбинированного метода не получить дисбаланс в другую сторону. Однако для целей конкретной задачи может понадобиться и другое соотношение между классами или же другая комбинация методов по уменьшению большей и увеличению меньшей выборки, которые показывают лучшие результаты на наборе данных.

Подбор и очередность методов должны быть индивидуальны в зависимости от зашумленности набора, линейной разделимости или неразделимости классов, количества объектов разных классов на границе и общего количества объектов в наборе данных.

1.2.2 Модификация классических алгоритмов классификации

Изменения на уровне алгоритмов не затрагивают набор данных и не вызывают сдвиги в их распределении. Это позволяет назвать подход более адаптированным к разным типам дисбаланса данных, поскольку изменения затрагивают только классификатор в момент обучения модели.

К наиболее чувствительным к дисбалансу классов алгоритмам относят SVM, наивный байесовский классификатор, решающие деревья и алгоритм k-ближайших соседей [7].

Некоторые из модификаций применимы только для конкретных алгоритмов: к примеру, преобразование ядра для SVM, чтобы перейти к пространству, в котором признаки могут быть более разделимыми; или подбор функции разделения для решающих деревьев.

Однако некоторые из методов не зависят от типа классификатора, а потому могут применяться для многих из них. Среди таких – обучение с учетом издержек классификации. Оно «сосредоточено на том, чтобы сначала назначить различные затраты типам ошибок неправильной классификации, которые могут быть допущены, а затем использовать специализированные методы для учета этих затрат» [7, с. 178]. Ярким примером метода из этого класса будет взвешивание классов – присвоение большего веса редким классам и меньшего веса преобладающим классам при обучении модели. В таком случае «весовые коэффициенты могут наказывать модель меньше за ошибки, допущенные в примерах из класса большинства, и больше за ошибки, допущенные в примерах из класса меньшинства» [7, с. 189].

1.2.3 Ансамблевые методы

Идея ансамблевых методов заключается в объединении прогнозов базовых классификаторов, входящих в состав ансамбля, для получения более

обобщенной и надежной модели. Качество модели напрямую зависит от видов классификаторов, которые объединены в ансамбль, а также от метода объединения.

К наиболее широко используемым на практике ансамблевым методам [13, 16] относятся следующие.

Случайный лес (Random Forest) – это «алгоритм, который применяет метод бэггинга для построения нескольких деревьев решений с использованием бутстрэп-выборок. Техника бэггинга заключается в генерации случайных выборок из исходных данных, на которых обучаются отдельные деревья решений» [26].

Градиентный бустинг зачастую основан на последовательном обучении моделей с использованием решающих деревьев в качестве базового алгоритма, а «его основная идея заключается в формировании базовых алгоритмов, которые демонстрируют сильную корреляцию с отрицательным градиентом функции потерь, соответствующей всему ансамблю» [26].

AdaBoost является реализацией градиентного бустинга над решающими деревьями. Идея алгоритма заключается в том, что «веса выборок корректируются в соответствии с прогнозами классификатора, а скорректированные выборки используются для обучения последующего классификатора. Таким образом, неправильно классифицированным образцам назначаются большие веса, а правильно классифицированным экземплярам — меньшие веса, гарантируя, что последующие классификаторы уделяют больше внимания неправильно классифицированным образцам» [26].

XGBoost также является реализацией градиентного бустинга над решающими деревьями. Отличием от классического градиентного бустинга является то, что «функция потерь XGBoost содержит член регуляризации, препятствующий переобучению» [26].

LightGBM является еще одной реализацией алгоритма градиентного бустинга над решающими деревьями. Алгоритм «использует два новых метода: одностороннюю выборку на основе градиента (GOSS) и эксклюзивное объединение признаков (EFB), что обеспечивает более быстрое обучение алгоритма и достижение высокой точности» [26]. Метод GOSS удаляет объекты с малыми градиентами, поскольку объекты с большими градиентами более полезны для вычисления прироста информации, а метод EFB объединяет разреженные взаимоисключающие признаки, уменьшая их число [26].

И хотя такие ансамблевые методы повышают точность модели по сравнению с изолированными алгоритмами, следует учесть, что они не всегда решают напрямую проблему дисбаланса классов. Однако позволяют делать гибкую настройку благодаря гиперпараметрам и адаптировать алгоритм к несбалансированным данным.

1.2.4 Комбинированные методы

Комбинированные методы заключаются в подборе техник из других групп методов и их объединения для того, чтобы задача решалась комплексно и на всех возможных уровнях – как данных, так и алгоритмов.

Одним из распространенных методов является объединение преобразований на уровне данных с обучением с учетом издержек классификации [9]. Также в [13] приводится большое количество примеров, где ансамбли объединяются с обучением с учетом издержек или балансировкой данных, основанные на бэггинге (UnderBagging, OverBagging, SMOTEBagging, Balanced Random Forests), бустинге (SMOTEBoost, RUSBoost, DataBoost-IM) и гибридные (EasyEnsemble и BalancedCascade). Библиотека imbalanced-learn [20] предоставляет реализации некоторых из них – а именно EasyEnsemble, RUSBoost, UnderBagging и Balanced Random Forests.

Гибридные методы на практике могут использоваться реже других, потому что они требуют тщательного анализа качества моделей на стандартных алгоритмах, настройки гиперпараметров, а также связаны с увеличением вычислительных затрат и времени на обучение. А, как упомянуто в [1], часто на практике важнее выбрать алгоритм, который будет давать высокую точность предсказаний, но также окажется подходящим по производительности и времени, затраченному на предобработку и обучение модели.

Однако в случаях, когда практическая задача позволяет временные затраты и требует крайне высокой точности, подбор методов и экспериментальное исследование по их комбинированию оправданы.

1.2.5 Оценка качества моделей

Существуют стандартные метрики [3, 10, 24], которые широко используются для оценки моделей в задачах бинарной и мультиклассовой классификации. Однако следует отметить, что часть этих метрик корректно отображает ситуацию только при условии сбалансированности набора данных. При использовании же в задачах с несбалансированными наборами данных они показывают чрезмерно оптимистичную оценку модели, и искажают объективную картину ее работы [2].

Рассмотрим эти метрики конкретнее и укажем, какие из них корректные для использования в задачи классификации с несбалансированным набором данных.

1) Матрица ошибок

Матрица ошибок представляет собой матрицу размерности $N \times N$, где N – число классов в задаче. Для заполнения матрицы в случае бинарной

классификации используются обозначения TP, TN, FP и FN. Примеры матриц ошибок для случая бинарной и мультиклассовой классификации приводятся на рисунке 1.2.

Строки матрицы ошибок для мультиклассовой классификации соответствуют истинным классам объектов, а вертикали – предсказанным. Таким образом, на пересечении первой строки и второго столбца в матрице ошибок находится число объектов первого класса, которым присвоены метки второго класса.

Матрица ошибок наглядно демонстрирует количество объектов, которые были верно и неверно классифицированы, и помогает предварительно оценить качество работы модели.

а б

а – матрица ошибок 2 × 2, б – матрица ошибок 3 × 3

Рисунок 1.2 – Матрицы ошибок для бинарной и мультиклассовой классификации

2) Классические метрики

Матрица ошибок является наглядным инструментом, однако классические метрики предлагают более точные возможности для оценки моделей. Они позволяют сосредоточиться на важных аспектах конкретной задачи, например, когда критически важно избегать пропуска представителей положительного класса.

Такие метрики можно разделить на три категории: пороговые, метрики ранжирования и метрики вероятности. Пороговые лучше подходят для задач, где требуется однозначно назначить каждому объекту метку конкретного класса. А метрики ранжирования и вероятности – для задач, где необходимо оценить вероятность принадлежности объекта к определенному классу.

2.1) Пороговые метрики.

Для вычисления пороговых метрик используются значения TP, TN, FP и FN, полученные ранее из матрицы ошибок. Формулы и описания основных широко используемых на практике метрик для бинарной классификации приводятся в таблице 1.1.

Таблица 1.1 – Классические метрики для задач классификации

Метрика	Формула	Описание
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Доля верно классифицированных объектов
Specifity (TNR, true negative rate)	$\frac{TN}{TN+FP}$	Специфичность – доля истинноотрицательных результатов
Sensitivity (TPR, true positive rate)	$\frac{TP}{TP+FN}$	Чувствительность – доля истинноположительных результатов
FPR (false positive rate)	$\frac{FP}{TN+FP}$	Доля ложноположительных результатов
FNR (false negative rate)	$\frac{FN}{TP+FN}$	Доля ложноотрицательных результатов
Precision	$\frac{TP}{TP+FP}$	Точность
Recall	$\frac{TP}{TP+FN}$	Полнота
F-мера	$(1+\beta^2)\frac{precision \cdot recall}{(\beta^2 \cdot precision)+recall}$	Среднее гармоническое точности и полноты
G-мера	$\sqrt{precision \cdot recall}$	Среднее геометрическое точности и полноты
G-среднее	$\sqrt{TPR \cdot TNR}$	Среднее геометрическое специфичности и чувствительности

Однако сравнительный анализ метрик, приведенный в работе [6] показывает, что все классические метрики за исключением FPR и FNR чувствительны к дисбалансу классов, что может привести к необъективности оценки модели.

В качестве иллюстрации можно привести случай с использованием показателя доли правильно классифицированных объектов (accuracy). При наличии выраженного дисбаланса, когда модель правильно предсказывает исключительно больший класс, этот показатель может стремиться к 100%, несмотря на ошибочную классификацию всех объектов меньшего класса.

Согласно выводам, представленным [5, 4] для задачи классификации с несбалансированным набором данных целесообразнее использовать сбалансированную (balanced accuracy), определяемую по формуле (1.1).

$$BalancedAccuracy = \frac{sensitivity+specifity}{2} \quad (1.1)$$

Также вполне корректно использовать метрики FPR и FNR – доля ложноположительных и ложноотрицательных результатов соответственно благодаря их нечувствительности к дисбалансу классов.

Следует отметить, что для задачи мультиклассовой классификации существуют следующие адаптированные метрики.

Микро: вычисляется общее количество TP, TN, FP, FN вместо вычисления индивидуальных метрик в каждом классе;

Макро: вычисляются значения матрицы ошибок для каждого класса, а затем – их невзвешенное среднее.

Взвешенное среднее: вычисляются значения матрицы ошибок для каждого класса, а затем – их взвешенное среднее с опорой на процентное соотношение объектов различных классов.

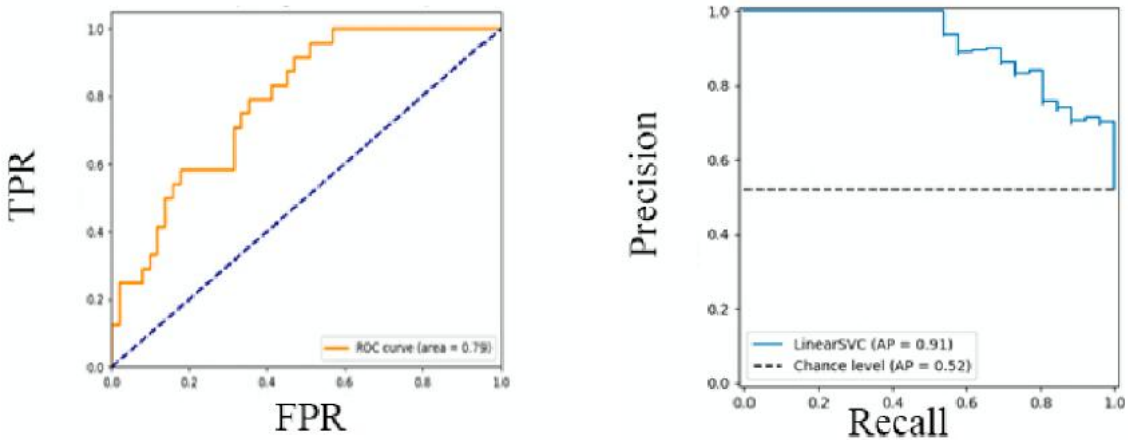
Как следует из определения методов, микро- и макрометрики не подходят для задач с несбалансированным набором данных, поскольку не учитывают дисбаланс классов. Вместо этого наиболее подходящим будет подсчет взвешенного среднего или определение метрик для отдельных классов.

2.2) Метрики ранжирования

Данные метрики применяются в том случае, когда для модели в первую очередь важно предсказание вероятностей принадлежности объекта к конкретному классу. В зависимости от величины порога объект классифицируется в положительный или отрицательный класс, а на основе величины порога и значений TPR и FPR для него можно построить график ROC-кривой. Аналогично по значениям порога, precision и recall можно построить Precision-Recall кривую. Пример этих кривых приводится на рисунке 1.3.

Для задач мультиклассовой классификации данные метрики можно адаптировать тем же способом, что и для пороговых метрик: вычисляя микро, макро или взвешенные средние значения.

Для задачи несбалансированной классификации по аналогии будут больше подходить кривые с вычисленными взвешенными средними – это поможет избежать необъективной оценки модели. Также в случае, когда один из классов имеет гораздо большую важность для классификации, чем остальные, метрики можно вычислять именно для него.



а б

а – ROC-кривая, б – кривая Precision-Recall

Рисунок 1.3 – Метрики ранжирования

2.3) Метрики вероятности

В то время как метрики ранжирования дают более наглядное представление о качестве решения задачи, основанной на предсказании вероятностей, метрики вероятности дают количественные показатели качества модели.

В качестве таких метрик широко используются логистическая функция потерь (LogLoss) и оценка Брайера (BrierScore) [18].

Логистическая функция потерь суммирует среднюю разницу между двумя распределениями вероятностей и вычисляется по формуле (1.2).

$$LogLoss = -\big((1-y)\big) \times \log\big(1-y_p\big) + y \times \log\big(y_p\big) \quad (1.2)$$

где y – истинный класс объекта;

y_p – предсказанный класс объекта.

Для мультиклассовой классификации формула логистической функции потерь (1.3) выглядит следующим образом:

$$LogLoss = -\sum_{i=1}^{n} y_i \times \log\big(y_{pi}\big) \quad (1.3)$$

где n – количество объектов набора данных;

y_i – истинный класс объекта;

y_{pi} – предсказанный класс объекта.

Оценка Брайера вычисляется как среднеквадратическая ошибка между ожидаемыми и прогнозируемыми вероятностями для некоторого класса по формуле (1.4).

$$BrierScore = \frac{1}{N} \times \sum_{i=1}^{N} \big(y_{pi} - y_i\big)^2 \quad (1.4)$$

где N – количество объектов соответствующего класса из набора данных.

Оценку Брайера для задач несбалансированной классификации можно назвать более предпочтительной по сравнению с логистической функцией потерь, потому что она ориентирована на вычисление погрешности для конкретного класса без учета всего распределения вероятностей.

1.3 Выводы по первой главе

Несбалансированность набора данных часто возникает в практических задачах в области медицины, бизнеса, безопасности, инженерии и смежных

сферах, и представляет собой серьезную проблему в машинном обучении, так как большинство стандартных алгоритмов плохо справляются с задачами классификации, когда один или несколько классов значительно преобладают над остальными.

Среди причин возникновения дисбаланса классов можно выделить специфику предметной области, ошибки сбора данных, смещенность выборки. При этом зачастую целесообразнее работать с имеющимся набором данных ввиду финансовых, ресурсных и временных ограничений на получение нового набора данных.

Классические алгоритмы классификации, такие как SVM, наивный байесовский классификатор, решающие деревья, алгоритм k-ближайших соседей чувствительны к дисбалансу классов и показывают недостаточную эффективность при работе с несбалансированными данными.

Для оценки качества моделей в задачах классификации с несбалансированным набором данных важно подбирать подходящие метрики: либо нечувствительные к дисбалансу классов (FPR и FNR), либо основанные на взвешенных средних (balanced accuracy, взвешенное среднее классических метрик).

Для задач предсказания вероятностей более оптимальной будет оценка Брайера (Brier Score), а также ROC и Precision-Recall кривые для нужного класса или со взвешенными оценками по классам.

Основные методы работы с несбалансированными данными можно разделить на четыре категории: балансировку данных, модификацию классических алгоритмов, ансамблевые методы и комбинированные. Их преимущества и недостатки приводятся в таблице 1.2.

Таблица 1.2 – Оценка подходов работы с несбалансированными данными

Метод	Преимущества	Недостатки
Балансировка данных	низкие временные затраты	небольшое улучшение возможность переобучения
Модификация классических алгоритмов	гибкость универсальность относительно наборов данных	сложность необходимы экспертные знания в области конкретных алгоритмов
Ансамблевые методы	достаточно высокое качество	временные затраты на побор гиперпараметров
Комбинированные методы	высокое качество тонкая настройка под нужды задачи	большие временные затраты

Анализ методов демонстрирует, что оптимизация качества классификации требует экспериментальной проверки различных методов и их комбинаций на реальных данных, что позволит выбрать наилучший подход для решения конкретной задачи.

ГЛАВА 2
ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ АЛГОРИТМОВ РАБОТЫ С НЕСБАЛАНСИРОВАННЫМИ ДАННЫМИ

В предыдущей главе рассмотрены теоретические аспекты проблемы несбалансированных данных, причины возникновения дисбаланса и влияние на работу классических алгоритмов машинного обучения. Также проведена систематизация методов балансировки, выделены основные подходы и выбраны корректные метрики для оценки качества моделей.

В данной главе описывается постановка задачи на экспериментальное исследование. Указываются используемые методы и средства работы с данными, описываются выбранные наборы данных и техники их предобработки. Комплексно анализируется влияние методов балансировки данных, применяемых изолированно и в комбинации, соотношения классов после балансировки и подбора гиперпараметров на качество моделей.

Полученные результаты будут использованы при разработке программного средства для обучения модели, рекомендующей алгоритмы балансировки в зависимости от размера набора данных и величины дисбаланса.

2.1 Постановка задачи

Для проведения объективного сравнительного анализа необходимо подобрать несколько наборов данных разной величины и разной степени дисбаланса среди реальных или синтетических наборов. Это позволит избежать узконаправленного анализа, ограниченного особенностями конкретного набора данных, и обеспечить более универсальные выводы, применимые к широкому спектру практических задач.

Чтобы условия эксперимента максимально соответствовали реальным и имели практическую ценность, необходимо предварительно выполнить разведочный анализ данных и предобработку.

Исходя из существующих методов и алгоритмов работы с несбалансированными данными, целесообразно в рамках экспериментальной части оценить влияние на качество моделей следующих методов:

- увеличение меньшего класса алгоритмами Random Oversampling, SMOTE, B-SMOTE, B-SMOTE SVM, ADASYN;
- уменьшение большего класса алгоритмами Random Undersampling, NearMiss, TomekLinks, CNN, ENN, OSS, NCR.

– применение ансамблевых методов: случайного леса, градиентного бустинга, XGBoost, AdaBoost и LightGBM.

Для эксперимента в случаях, когда не применяются ансамблевые методы, следует взять наиболее распространенные классические модели: дерево решений, MLP, k-ближайших соседей, наивный байесовский классификатор.

В качестве метрик для сравнения качества моделей наиболее подходящими будут матрицы ошибок и сбалансированная точность.

Для каждого набора данных необходимо отобрать методы, наилучшим образом показавшие себя на предыдущем этапе, и оценить качество модели при их комбинации. Целесообразно сравнить полученные комбинации со встроенными сбалансированными ансамблями. Также необходимо определить оптимальное соотношение классов, выполнив балансировку не только с соотношением 1:1, а с постепенным изменением его от 1:10 до 100:10.

Также целесообразно провести подбор гиперпараметров, который является завершающим этапом работы над моделями машинного обучения, чтобы можно было сделать выводы о том, насколько значительно каждый из инструментов работы с данными влияет на итоговое качество модели.

2.2 Выбор средств и инструментов работы с данными

2.2.1 Язык программирования и среда разработки

В данной работе основным языком программирования выбран Python, что обусловлено наличием специализированной библиотеки imbalanced-learn [20]. Эта библиотека предоставляет широкий спектр алгоритмов для обработки несбалансированных данных, что значительно расширяет возможности исследования и избавляет от необходимости выполнять программную реализацию каждого метода вручную.

Помимо Python для машинного обучения широко используется язык программирования R, однако преимущество было отдано именно Python благодаря отсутствию необходимости изучать его с нуля, большому количеству примеров кода для машинного обучения в целом и несбалансированных данных в частности, и простоты использования.

В качестве основных сред разработки для выполнения кода на Python в настоящее время можно выделить Google Colab и Jupyter Notebook. Первая имеет преимущество благодаря возможности использовать мощные вычислительные ресурсы серверного ускорителя tensor processing units (TPU) без необходимости ручной настройки. Вторая же позволяет использовать ресурсы компьютера без ограничений в то время, как ресурсы серверного ускорителя в бесплатной версии

имеют ограничения и не подходят для масштабных исследований В процессе экспериментального исследования выбрана использовались обе среды.

2.2.2 Используемые библиотеки

Для реализации экспериментального исследования использовались библиотеки языка Python, предоставляющие удобные инструменты для обработки наборов данных, обучения моделей, балансировки данных и визуализации.

На всех этапах исследования использовались библиотеки Pandas и Numpy для работы с табличными данными.

Во время разведочного анализа данных использовались Matplotlib и Seaborn для визуализации наборов данных и корреляционных матриц.

На этапе предобработки наборов данных использовались Mlxtend и Scikit-learn для отбора признаков, Scikit-learn для масштабирования данных, кодирования категориальных признаков, разбиения на тестовую и обучающую выборку.

При обучении моделей и балансировки данных использовались Scikit-learn для создания пайплайнов, использования встроенных алгоритмов и ансамблей и расчета метрик, XGBoost и LightGBM для использования соответствующих ансамблей, imblearn для балансировки данных, Optuna для автоматизированного подбора гиперпараметров.

2.3 Подготовка исходных данных

2.3.1 Выбор наборов данных

Наборы данных выбирались на платформе Kaggle [14] являющейся наиболее известной среди специалистов в области машинного обучения благодаря доступу к большому числу наборов данных различной тематики. Выбор наборов данных выполнялся по следующим критериям [2]:

- величина наборов данных: должны быть представлены как сравнительно небольшие наборы, включающие до 10 тысяч записей, так и средние и большие – до и свыше 100 тысяч записей соответственно;
- величина дисбаланса: в наборах данных одного размера должен быть представлен легкий, умеренный и сильный дисбаланс классов.
- величина признакового пространства;
- тематика наборов данных.

По данным критериям были отобраны пять различных наборов данных, данные о них приводятся в таблице 2.1.

Таблица 2.1 – Описание выбранных наборов данных

Название	Размер набора данных	Тип классификации	Соотношение классов
Классификация здоровья плода	~2 тысячи записей	мультиклассовая	8:100 и 13:100
Прогнозирование церебрального инсульта	~43 тысячи записей	бинарная	2:100
Кредитный риск	~32 тысячи записей	бинарная	22:100
Классификация кредитного рейтинга	100 тысяч записей	мультиклассовая	17:100 и 29:100
Показатели здоровья при диабете	~254 тысячи записей	мультиклассовая	2:100 и 15:100

2.3.2 Разведочный анализ и предобработка

Для обеспечения корректности и надежности результатов экспериментального исследования на этапе разведочного анализа и предобработки были применены методы визуализации, обработки данных и генерации признаков.

На первом этапе изучена структура наборов и используемые типы данных в них, построены диаграммы для визуализации распределения признаков (приложение А) и корреляционные матрицы для оценки зависимости между признаками.

Исходные наборы данных проверялись на наличие пропущенных значений, дубликатов и выбросов. Для устранения пропущенных данных использовались стратегии заполнения медианным значением и удаление экземпляров, также удалялись дубликаты. Выбросы корректировались заменой на значения соседних точек или удалением.

Поскольку ряд признаков носил категориальный характер, они были преобразованы в числовой формат, для этого использовался метод LabelEncoder из библиотеки Scikit-learn. Данный подход позволил обеспечить компактное представление данных при сохранении их значимости для модели.

Для определения наиболее значимых признаков была применен метод mutual_info_classif из библиотеки Scikit-learn, позволяющий оценить степень зависимости между признаками и целевой переменной. Полученные значения использовались для последующего отбора наиболее значимых признаков.

Для приведения числовых признаков к единому масштабу и устранения влияния разнородных диапазонов значений использовались нормализация при помощи метода MinMaxScaler библиотеки Scikit-learn.

Для того, чтобы можно было оценить качество моделей во время классификации, исходные наборы данных делились на тренировочную и тестовую выборки в соотношении 70% и 30% при помощи метода train_test_split библиотеки Scikit-learn.

В результате разведочного анализа данных и предобработки были получены более чистые наборы данных, адаптированные для балансировки и применения алгоритмов машинного обучения.

2.4 Исследование влияния алгоритмов балансировки

Как и ожидалось после выполненного в первой главе анализа литературы, наихудшие результаты до балансировки данных по сбалансированной точности получили классификаторы, обученные на исходных наборах данных с большим дисбалансом между классами.

При легком и умеренном дисбалансе ансамблевые алгоритмы превосходят классические модели машинного обучения на 5-10%. Однако при крайне выраженном дисбалансе ансамблевые модели дают результаты значительно хуже классических: ухудшение может достигать 22% и даже больше. Причем при таком дисбалансе в целом и те, и другие модели показали очень плохие результаты: 0.5 для набора данных меньшего размера и 0.39 для набора данных большего размера. Единственный алгоритм, который выделился в этом случае – наивный байесовский, который показал результаты в 0.65 и 0.45 соответственно.

2.4.1 Изолированное применение алгоритмов

Таблицы с результатами приводятся в приложении Б. Также результаты исследований приводятся в опубликованной мною статье [2]: «в большинстве случаев классификаторы, обученные после балансировки набора данных изолированно каждым из выбранных методов, показывали результаты не хуже, чем на исходном наборе».

«Наиболее противоречивым можно назвать алгоритм уменьшения большей выборки Near Miss. Наиболее чувствительным к нему оказался средний набор данных с крайне выраженным дисбалансом, причем для него вариант NearMiss-1 показал ухудшение сбалансированной точности вплоть до 24 процентных пунктов, а вот вариант NearMiss-3 улучшил сбалансированную точность в среднем на 13 процентных пунктов – с 0.52 до 0.65. Для остальных же классификаторов и наборов данных применение этого метода дало результаты не лучше или даже существенно хуже, чем при обучении на исходном наборе» [2].

«Особенно хорошо показали себя методы балансировки данных на небольшом наборе данных и на наборе данных среднего размера, но с ярко выраженных дисбалансом классов. Для них удалось добиться значительного повышения качества моделей. А вот для больших наборов данных положительный эффект от балансирования класса оказался менее выраженным. При этом использование

алгоритмов увеличения меньшего класса дало больший прирост сбалансированной точности, особенно для не-ансамблевых алгоритмов, чем использование алгоритмов уменьшения большего класса» [2].

«Ансамблевые модели оказались менее подвержены влиянию балансировки данных в случае легкого и умеренного дисбаланса, однако и для них удалось добиться небольшого повышения точности» [2]. Следует отметить, что повышение точности наблюдалось в пределах 5%.

Для наборов с крайне выраженным дисбалансом добавление балансировки к ансамблевому методу привело к большому скачку сбалансированной точности. Если до применения балансировки ухудшение относительно классических алгоритмов могло составлять 15-20%, то после балансировки ансамбли давали уже улучшение результатов относительно классических алгоритмов на 13-18%.

Как уже было сказано в опубликованной статье, «для больших наборов данных со значительным дисбалансом результат классификации все равно является крайне низким – значение сбалансированной точности близко к 0.5, что не позволяет применять полученную модель в реальной практике» [2].

2.4.2 Комбинированное применение алгоритмов

Табличное представление результатов экспериментов по комбинированному применению методов балансировки отображено в приложении В.

Несмотря на то, что в литературе [7] упоминаются некоторые сочетания алгоритмов увеличения меньшего класса и уменьшения большего класса в качестве эталонных, широко распространенных и тех, которые дают неплохие результаты на большинстве наборов данных, нельзя рекомендовать какие-то комбинации алгоритмов в качестве универсальных.

На первом наборе данных комбинация с OSS показала хорошее улучшение качества модели (до 3-5%), а на втором наборе данных – значительное ухудшение вплоть до -21%. Объяснить это можно тем, что для первого набора данных изолированное применение OSS улучшило или по крайней мере оставило таким же качество модели, а для второго – ухудшило, причем для некоторых моделей достаточно сильно.

В целом прирост сбалансированной точности при комбинации алгоритмов оказался не очень высоким. Иногда он мог быть и отрицательным, когда выбирался алгоритм уменьшения большего класса, который не ухудшал, но и не сильно улучшал результаты модели при изолированном применении.

Исходя из этого можно сделать вывод о том, что комбинирование алгоритмов целесообразно в тех случаях, когда есть временные и вычислительные ресурсы, и требуется как можно большая точность модели. Причем выбирать для комбинации следует те алгоритмы, которые изолированно дают наилучший

прирост качества модели. В ином случае даже использование широко распространенных и известных сочетаний может обернуться ухудшением точности модели, если изолированно эти алгоритмы из сочетания не дают значимого положительного результата.

2.4.3 Применение встроенных сбалансированных ансамблей

Для исследования были выбраны реализованные в библиотеке imbalanced-learn сбалансированные ансамбли:

- *EasyEnsemble* – ансамбль использует бэггинг сбалансированных при помощи Random Undersampling классификаторов AdaBoost [13];
- *RUSBoost Random* – ансамбль использует сбалансированный при помощи Random Undersampling один классификатор AdaBoost [13];
- *Balanced Bagging* – ансамбль на основе бэггинга (над решающими деревьями при использовании параметров по умолчанию) и Random Undersampling [13];
- *BalancedRF* – ансамбль на основе случайного леса и Random Undersampling [13].

Из-за того, что алгоритм случайного удаления на некоторых наборах данных и моделях показывал не очень хорошие результаты, выдвинуто предположение, что и в ансамблях сбалансированная точность модели будет не очень хорошей. Однако анализ результатов, приведенный в таблицах 2.2-2.4, показывает, что в некоторых случаях такие сбалансированные ансамбли показывали лучшие результаты, чем последовательное ручное применение балансировки и ансамбля. А по сравнению с использованием исключительно ансамбля без балансировки результаты сбалансированных ансамблей оказались строго лучше.

В таблицах приводятся значения сбалансированной точности для сбалансированных ансамблей, для базовых ансамблей без балансировки, и для комбинации из балансировки и применения затем базового ансамбля.

Таблица 2.2 – Результаты для Easy Ensemble и RUSBoost

Набор данных	Easy Ensemble	RUSBoost	AdaBoost	Random Undersampling + AdaBoost
«Классификация здоровья плода»	0,911	0,794	0,818	0,777
«Прогнозирование церебрального инсульта»	0,771	0,718	0,500	0,774
«Кредитный риск»	0,824	0,827	0,806	0,825
«Классификация кредитного рейтинга»	0,684	0,697	0,632	0,715
«Показатели здоровья при диабете»	0,508	0,508	0,391	0,410

Таблица 2.3 – Результаты для BalancedBagging

Набор данных	BalancedBagging	Бэггинг	Random Undersampling + бэггинг
«Классификация здоровья плода»	0,919	0,872	0,901
«Прогнозирование церебрального инсульта»	0,727	0,508	0,719
«Кредитный риск»	0,854	0,853	0,847
«Классификация кредитного рейтинга»	0,823	0,800	0,810
«Показатели здоровья при диабете»	0,465	0,387	0,418

Таблица 2.4 – Результаты для BalancedRF

Набор данных	BalancedRF	Случайный лес	Random Undersampling + Случайный лес
«Классификация здоровья плода»	0,918	0,887	0,912
«Прогнозирование церебрального инсульта»	0,769	0,502	0,760
«Кредитный риск»	0,854	0,845	0,845
«Классификация кредитного рейтинга»	0,847	0,805	0,825
«Показатели здоровья при диабете»	0,494	0,386	0,413

При использовании сбалансированных ансамблей не получилось добиться результатов, по качеству превышающих модели с применением методов увеличения меньшего класса и комбинирования. Но, несмотря на это, результаты получились сравнительно хорошими, поэтому использование сбалансированных ансамблей можно рекомендовать в условиях ограниченных временных ресурсов на обучение модели. Или же для предварительной оценки результатов, которые можно получить при объединении методов балансировки с ансамблями.

2.5 Определение оптимального соотношения классов

Алгоритмы балансировки в библиотеке imbalanced-learn позволяют задавать стратегию сэмплирования в виде гиперпараметра, определяющего итоговое соотношение классов после балансировки. Соотношение по умолчанию в этих алгоритмах 1:1, однако для полноценного анализа влияния балансировки необходимо изучить, как изменяется качество модели в зависимости от изменения соотношения классов. Это позволит либо определить оптимальное значение, либо установить целесообразность подбора гиперпараметров в определенном в ходе исследования диапазоне.

Методология исследования была сформирована следующим образом.

Для задачи мультиклассовой классификации в качестве начального соотношения меньшего класса к большему выбиралось наибольшее из доступных значений. К примеру, для соотношения классов 8:100 и 13:100 в качестве отправной точки выбиралось 13:100. В бинарной классификации этот параметр определяется единственным возможным образом.

После выбора начального соотношения оно округлялось в большую сторону до ближайшего десятка, и полученное значение использовалось в алгоритме в качестве стартового.

До соотношения 100:100 меньшее число увеличивалось с шагом в 10. После этого шаг принимался равным 100, пока не будет получено соотношение 1000:100.

Графическое представление результатов отображено в приложении Г.

По результатам анализа графиков можно сделать вывод о том, что для методов балансировки, давших наилучшие результаты, оптимальное соотношение классов (пик на графике) располагается между 100:100 и 200:100. Для методов, показавших не такие хорошие результаты, пик может находиться и дальше – возле 600:100 или даже правее. Однако несмотря на подобное поведение графиков, такие результаты едва ли можно считать показательными для определения оптимального соотношения, потому что итоговое качество моделей на этих смещенных вправо пиках оказалось сравнительно не очень хорошим.

Можно заметить также и то, что результаты для соотношений меньше 100:100 не являются пиковыми, из чего можно сделать вывод, что нецелесообразно балансировать классы до меньшего соотношения, чем один к одному.

Показательно также и то, что набор данных маленького размера реагировал на малое изменение соотношения балансировки большими скачками качества модели при немонотонности графика, а вот для набора данных большего размера графики оказались более сглаженными.

В то же время получить единственное универсальное значение соотношения классов не удалось, оно оказалось вариативным в зависимости от набора данных. Поэтому и рекомендовать балансировать классы до соотношения 1:1 или к 2:1 пока нельзя – но можно говорить о целесообразности подбора соотношения в диапазоне от 1:1 до 2:1.

При этом нет и явной очевидной зависимости между выраженностью исходного дисбаланса набора и смещением оптимального соотношения правее, чем 1:1. Поэтому нельзя рекомендовать создание искусственного дисбаланса в обратную сторону для наборов данных с сильно выраженным дисбалансом, как и не обязательно превышение отношения 1:1 негативно скажется на наборе данных с легким или умеренным дисбалансом.

Дополнительно была изучена эффективность двух стратегий балансировки в задаче мультиклассовой классификации: изолированная балансировка одного из классов и одновременная балансировка обоих классов.

Результаты анализа показали, что при балансировке одного класса наилучшее качество моделей наблюдаются в случае увеличения численности того из меньших классов, который изначально содержал больше объектов. Однако одновременная балансировка обоих классов обеспечивает значительно лучшие результаты по сравнению с балансировкой только одного из них.

На основании полученных результатов можно рекомендовать стратегию балансировки сразу обоих классов, поскольку в этом случае качество моделей оказывается гарантированно не хуже, чем при балансировке только одного из них.

2.6 Анализ влияния подбора гиперпараметров

Табличное представление результатов экспериментального исследования по влиянию подбора гиперпараметров отображено в приложении Д.

Для набора данных небольшого размера с умеренным дисбалансом («Классификация здоровья плода») подбор гиперпараметров дал не очень выраженный эффект. Наибольшим он был на тех моделях, где использовался только один метод балансировки, а не их комбинация. Для таких случаев улучшение составило в среднем 2-4%. В тех случаях, когда использовалась комбинация из алгоритма увеличения меньшего класса и алгоритма уменьшения большего класса, улучшение составило в среднем 1-2%.

Для набора данных среднего размера с чрезвычайным дисбалансом («Прогнозирование церебрального инсульта») подбор гиперпараметров существенно улучшил работу моделей с балансировкой методом OSS (в среднем на 10-20%) и для алгоритма градиентного бустинга (12-25%). Крайне выраженными оказались результаты подбора гиперпараметров для градиентного бустинга, когда балансировка не выполнялась – улучшение составило 47%. Также выраженным оказался эффект от подбора гиперпараметров для AdaBoost, но он сильнее проявился на моделях с балансировкой: здесь улучшение в среднем на 5-15%.

Для набора данных среднего размера с умеренным дисбалансом («Кредитный риск») подбор гиперпараметров дал небольшой прирост качества моделей. Как и в предыдущем наборе данных, больше всего от подбора гиперпараметров улучшился алгоритм градиентного бустинга (4-5%). Еще менее выраженные результаты для LightGBM и AdaBoost: порядка 2-3%. Случайный лес и XGBoost практически не изменили показателей.

Для набора данных большого размера с легким дисбалансом («Классификация кредитного рейтинга») результаты оказались средними. Больше всего

выиграл от подбора гиперпараметров LightGBM: у него улучшение составило 8-9%, чуть меньше – XGBoost (5-7%). Эффект на случайном лесе оказался практически не выражен.

Для набора данных большого размера с чрезвычайным дисбалансом («Показатели здоровья при диабете») отличные результаты дал подбор параметров для XGBoost (в среднем 26-27%) и LightGBM (14-21%). Улучшение для градиентного бустинга и AdaBoost оказался средним, около 4-6%.

Для всех наборов данных характерно то, что подбор гиперпараметров не смог скомпенсировать отсутствие балансировки данных – результаты моделей оказались существенно хуже даже тех, где выполнялась только балансировка без последующего подбора гиперпараметров.

Однако качество моделей при комбинировании методов балансировки данных, ансамблевых алгоритмов и подбора гиперпараметров оказалось наилучшим.

Также можно сделать вывод о том, что чувствительность моделей к подбору гиперпараметров зависит от исходного уровня дисбаланса данных – чем сильнее выражен дисбаланс, тем более значимым оказывается эффект от настройки гиперпараметров, особенно для алгоритмов градиентного бустинга.

Алгоритмы балансировки данных за исключением OSS в целом оказались не очень чувствительны к подбору гиперпараметров кроме как на наименьшем наборе данных. Поэтому в условиях ограниченного времени можно рекомендовать подбор гиперпараметров для алгоритмов балансировки только в том случае, когда используется OSS, или когда набор данных очень мал – особенно если балансировка выполняется только одним методом.

Следует отметить также и наблюдение, что если модель изначально показывала худшие результаты по сравнению с другими, то даже после подбора гиперпараметров её результаты не становились лучше, чем у тех моделей, которые изначально были сильнее и тоже проходили подбор гиперпараметров.

2.7 Анализ влияния этапов настройки модели на итоговый результат

Результаты экспериментального исследования обобщены в таблице 2.5. Во избежание ее загромождения для наборов данных приводилось только соотношение классов, а для каждой категории – наилучшее значение сбалансированной точности и изменение в процентах относительно результатов сбалансированной точности классического алгоритма.

Таблица 2.5 – Результаты по категориям для наборов данных

Соотношение классов	Классический алгоритм	Ансамбль	Одна балансировка + ансамбль	Комбинация + ансамбль	Комбинация + ансамбль + гиперпараметры
8:100 и 13:100	0,873	0,920 +5,38%	0,939 +7,56%	0,950 +8,82%	0,954 +9,28%
2:100	0,650	0,502 -22,77%	0,773 +18,92%	0,780 +20,00%	0,782 +20,31%
22:100	0,839	0,865 +3,10%	0,874 +4,17%	0,874 +4,17%	0,882 +4,67%
17:100 и 29:100	0,732	0,805 +9,97%	0,826 +12,84%	0,836 +14,21%	0,840 +14,75%
2:100 и 15:100	0,453	0,391 -13,69%	0,513 +13,25%	0,516 +13,91%	0,517 +14,13%

По табличным данным наглядно видно ухудшение качества при применении ансамбля без балансировки на наборах данных с чрезвычайным дисбалансом, и скачок точности при добавлении балансировки. Изменение сбалансированной точности с 0.502 до 0.773 и с 0.391 до 0.513 соответствует улучшению на 54% и 31% соответственно.

Наибольший вклад в итоговую модель давало применение одного метода балансировки, а с учетом того, что пиковые значения достигались на методах увеличения меньшего класса – для решения практических задач может быть достаточно объединения одной балансировки с ансамблем.

Вклад добавления еще одного метода балансировки и подбора гиперпараметров оказался сравнительно низким. Причем закономерности, что именно добавление второго метода или именно подбор гиперпараметров дает строго больший вклад, на рассмотренных данных нет.

Для наглядности и визуальной оценки вклада каждого из этапов работы с набором данных была построена гистограмма. Для подсчета вклада использовались результаты, полученные в результате применения следующих сочетаний:

1) классический алгоритм;
2) балансировка и классический алгоритм;
3) балансировка и ансамбль;
4) комбинации балансировок и ансамбль;
5) комбинация балансировок и ансамбля с подбором гиперпараметров.

На каждом этапе выбирались наилучшие полученные значения сбалансированной точности на выбранном наборе данных, определялась разница между текущим и предыдущим сочетанием, начальное значение принималось равным нулю. После этого строилась гистограмма с накоплением, приведенная на рисунке 2.1.

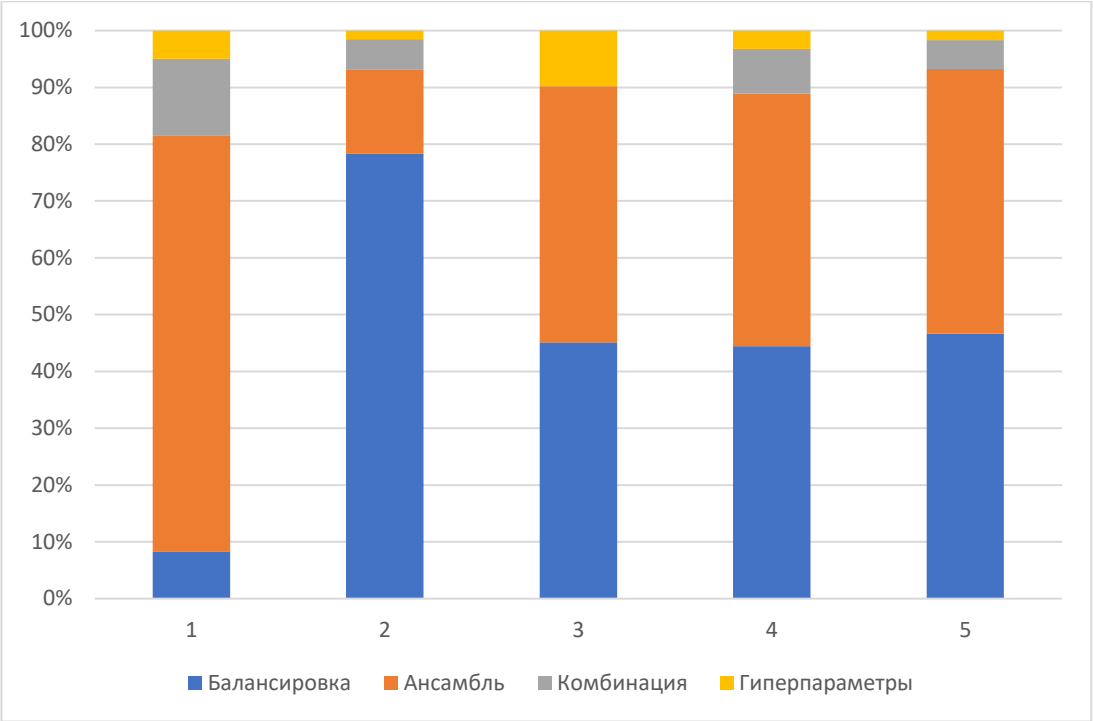


Рисунок 2.1 – Вклад этапов работы с набором данных в итоговый результат

2.8 Выводы по второй главе

В ходе экспериментального исследования были выбраны инструменты для работы с данными: язык программирования Python и сопутствующие библиотеки, библиотека для работы с несбалансированными данными imbalanced-learn, фреймворк для подбора гиперпараметров Optuna.

Подобраны пять наборов данных с платформы Kaggle с учетом объема, размерности признакового пространства, степени дисбаланса классов и тематической направленности, проведены разведочный анализ данных и предобработка.

Модели балансировки данных в большинстве случаев улучшали качество классификации, особенно для небольших и средних наборов данных с ярко выраженным дисбалансом классов. Из всех методов увеличение меньшего класса дало наиболее значительный прирост точности.

Ансамблевые методы при умеренном дисбалансе давали небольшой прирост качества модели, но при чрезвычайном дисбалансе требовалась балансировка для того, чтобы результаты оказались не хуже, чем у классических алгоритмов.

Комбинирование различных подходов балансировки дало сравнительно небольшой прирост качества модели. Для комбинирования целесообразно выбирать те алгоритмы, которые при изолированном применении дали наилучший результат, иначе качество модели может ухудшиться.

Встроенные сбалансированные ансамбли давали результаты лучше, чем ансамбли без балансировки, однако они уступают в сбалансированной точности моделям, где объединены несколько алгоритмов балансировки и ансамбль.

Наиболее эффективной является балансировка одновременно всех меньших классов, если стоит задача мультиклассовой классификации, и подбор соотношения классов в диапазоне между 1:1 и 2:1, где первое число соотносится с количеством объектов изначально меньшего класса.

Оптимизация гиперпараметров не компенсирует отсутствие предварительной балансировки данных: модели без балансировки демонстрируют существенно более низкие показатели, чем модели, где применялись методы балансировки (даже без последующего подбора гиперпараметров).

Наилучшие результаты достигаются при комплексном подходе, объединяющем балансировку данных разными методами, использование ансамблей и настройку гиперпараметров. Влияние подбора гиперпараметров коррелирует с уровнем дисбаланса: чем он сильнее выражен, тем более существенным оказывается эффект, особенно для градиентного бустинга. Но в целом алгоритмы балансировки демонстрирую достаточно низкую чувствительность к настройке гиперпараметров, за исключением метода OSS или небольших наборов данных.

Наибольший вклад в качество моделей дало применение одного метода балансировки вместе с обучением модели при помощи ансамбля. Поэтому в условиях ограниченных временных и вычислительных ресурсов можно рекомендовать применение метода увеличения меньшего класса вместе с ансамблем. А объединение с методом уменьшения большего класса и подбор гиперпараметров целесообразно проводить при достаточном количестве ресурсов и высоких требованиях к качеству модели.

Также необходимо отметить, что модели с изначально низкими показателями не смогли превзойти более эффективные модели даже после подбора гиперпараметров. Поэтому для подбора целесообразно выбирать те модели, которые дали наилучшие результаты на предварительном этапе.

Если обобщить результаты экспериментального исследования, то для балансировки можно выдвинуть гипотезу о справедливости принципа оптимальности, созвучного принципу оптимальности Беллмана: если на каждом этапе (балансировка данных, выбор модели, подбор гиперпараметров) выбирать оптимальное решение для конкретного набора данных, то в результате этого будет выбрано глобально оптимальное решение. При этом если на каком-то этапе было выбрано неоптимальное решение, то последствия этого не могут быть исправлены в будущем.

ГЛАВА 3
РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА

В предыдущей главе описано проведение экспериментального исследования, в котором анализировалось, как этапы балансировки данных, обучения модели и подбора гиперпараметров влияют на итоговое качество модели в задаче классификации.

В этой главе будет рассмотрена разработка программного средства, которое интегрирует результаты проведенного экспериментального исследования и позволяет автоматизировать процесс выбора оптимальных методов балансировки данных. Основное внимание уделено проектированию системы и модели ранжирования, рекомендующей методы балансировки.

Полученные результаты позволят продемонстрировать практическую применимость результатов исследования в задачах машинного обучения.

3.1 Постановка задачи

Необходимо разработать программное средство, позволяющее на основе характеристик входного набора данных (таких как размер выборки, соотношение классов и тип классификации) автоматически определять и рекомендовать оптимальные методы балансировки данных, и проводить балансировку данных встроенными методами.

Целью разработки программного средства является применение полученных результатов экспериментального исследования.

Программное средство должно обеспечивать удобный интерфейс для анализа, обработки и визуализации набора данных, а также способствовать повышению качества классификационных моделей за счет корректной предобработки.

Программное средство должно обеспечивать следующие функциональные возможности:

1) загрузка пользовательского набора данных либо использование встроенного демонстрационного примера;
2) разведочный анализ данных: показ первых строк набора данных;
3) обработка пропущенных значений: удаление или заполнение модой, медианой либо средним значением по выбору пользователя;
4) рекомендация подходящих моделей балансировки и их сочетаний;
5) балансировка данных методами увеличения меньшего класса и уменьшения большего класса по выбору пользователя вне зависимости от рекомендованных на предыдущем этапе;

6) визуализация: построение круговой и столбчатой диаграмм для сравнения распределения классов до и после применения методов балансировки;
7) сохранение итогового набора данных после применения всех этапов обработки в формате csv.

В результате будет получено удобное приложение, которое автоматизирует процессы обработки данных, включая обработку пропущенных значений, балансировку и визуализацию результатов. Итоговое программное средство станет функциональным и практическим решением, способствующим улучшению качества классификации за счет корректной предобработки данных.

3.2 Выбор средств и методов реализации проекта

Для реализации программного средства выбран язык Python, как и для проведения экспериментального исследования, благодаря возможностям его библиотек. В качестве среды разработки используется VS Code, для удобного сохранения изменений – система контроля версий Git и веб-сервис для хостинга репозиториев GitHub. Для работы с наборами данных и балансировки выбраны библиотеки Pandas и Plotly.

Принято проектное решение разработать веб-приложение, что объясняется некоторыми факторами.

Веб-приложение не требует установки на локальные машины, обеспечивая удобный доступ с любого устройства, имеющего подключение к интернету – это ускоряет процесс работы с приложением на первичном этапе.

Пользователь может загружать, анализировать и балансировать данные в интерактивной среде, не привязываясь к конкретной операционной системе или вычислительным ресурсам устройства. Также веб-приложение легко интегрируется с облачными решениями и API, что позволит легко масштабировать обработку данных и работать с большими наборами в дальнейшем, если будет принято решение продолжить работу над разработанным программным средством.

Поддержка работы веб-приложения в браузере устраняет необходимость установки зависимостей в виде других версий библиотек, чем пользователь обычно использует в работе.

Таким образом, проектирование веб-архитектуры обусловлено её преимуществами в доступности, гибкости и удобстве взаимодействия.

В качестве основного фреймворка для разработки веб-интерфейса принято решение использовать Streamlit.

Основное преимущество фреймворка заключается в том, что можно провести бесплатное развертывание и запуск разработанного с его помощью веб-приложения в облачной среде Streamlit Community Cloud.

Реализация пользовательского интерфейса осуществляется на языке Python, что убирает необходимость интегрировать различные языки программирования в проект. Также к его преимуществам относится то, что он позволяет создавать интерактивные веб-приложения без необходимости написания сложного клиент-серверного модуля. Это, в свою очередь, позволяет сосредоточиться на разработке ключевых алгоритмов и модулей программного средства, меньше отвлекаясь на детали реализации интерфейса и коммуникации с пользователем.

В то же время фреймворк предоставляет достаточно широкие возможности по добавлению интерактивных и визуальных элементов, а динамическое обновление страницы позволяет мгновенно отображать результаты работы. Все вышеперечисленное делает Streamlit оптимальным решением для реализации проекта.

3.3 Проектирование и разработка программного средства

Логическое моделирование системы выполним, разработав функциональную модель и модель ранжирования для рекомендации алгоритмов балансировки.

3.3.1 Функциональная модель

Функциональную модель процесса взаимодействия пользователя с приложением разработаем в нотации IDEF0. Диаграмма процесса приводится на рисунке 3.1.

Рисунок 3.1 – Функциональная модель программного средства

На вход процесса подается путь к файлу с набором данных, а также обученная модель ранжирования для рекомендации алгоритмов балансировки. На выходе функция формирует обработанный набор данных. Управляющее

воздействие на процесс оказывается на основе существующих реализаций методов балансировки данных и запросы пользователя. Процесс обеспечивается механизмами, к которым относятся программное средство и пользователь.

В основу построения функциональной модели, согласно поставленным задачам, выбраны следующие функции:

- загрузка данных;
- разведочный анализ данных и обработка пропущенных значений;
- рекомендация методов балансировки;
- балансировка данных и визуализация результатов.

Диаграмма декомпозиции процесса взаимодействия пользователя с приложением приводится на рисунке 3.2.

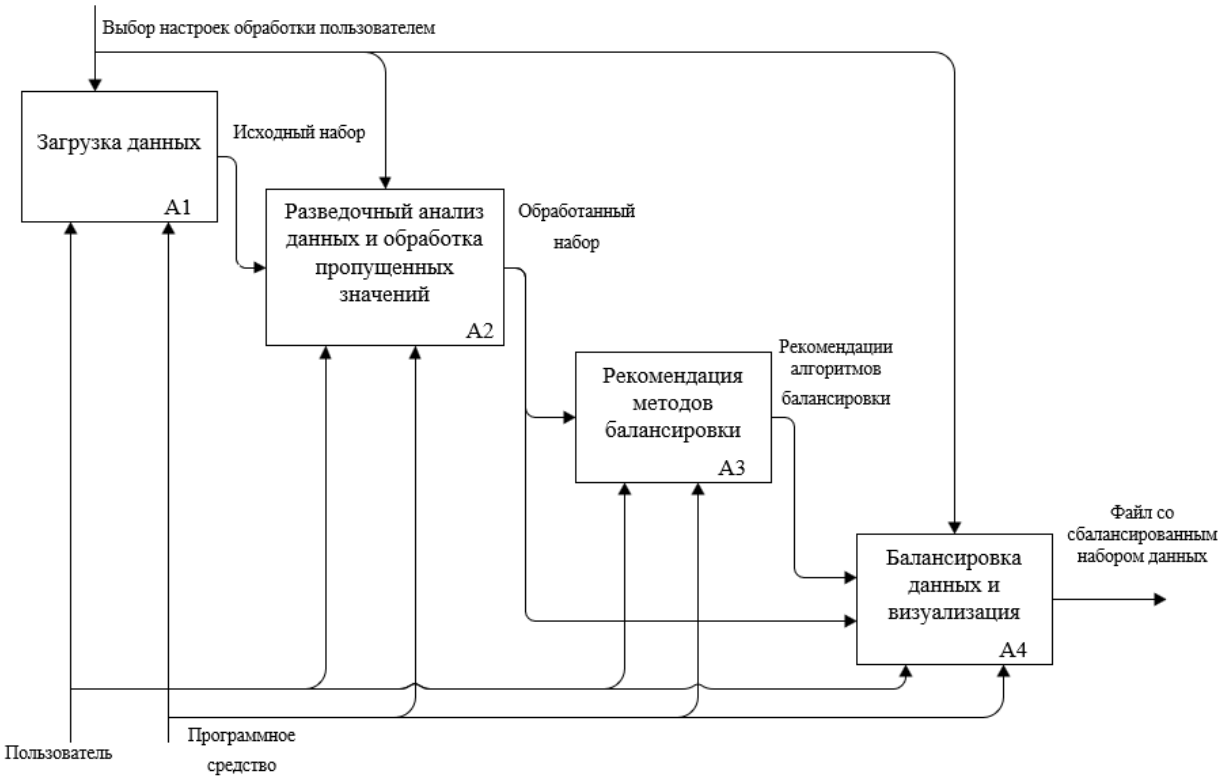


Рисунок 3.2 – Декомпозиция функциональной модели программного средства

3.3.2 Разработка модели ранжирования для рекомендации алгоритмов балансировки

1) Подготовка обучающего набора данных и предобработка

По результатам экспериментального исследования, описанного в главе 2, были получены значения сбалансированной точности для каждого из исходных наборов данных после применения к ним балансировок одним методом и их комбинацией – как при обучении классическими алгоритмами, так и при использовании ансамблей.

Выбор конкретной модели (классический алгоритм или ансамбль) при работе с набором данных остается на усмотрение специалиста по машинному обучению. Поэтому выбирались максимальные значения сбалансированной точности среди полученных, сгруппированные по сочетанию метода увеличения меньшего класса и уменьшению большего класса: без учета того, были полученные наилучшие результаты получены на классической или ансамблевой модели.

В результате был получен следующий обучающий набор. В качестве признаков в нем используются следующие:

- количество объектов набора данных, указанный в тысячах (Size);
- отношение наименьшего класса к большему – рациональное число (Imbalance Ratio);
- тип классификации: бинарная или мультиклассовая (Type);
- метод увеличения меньшего класса (Oversampling), при его отсутствии – значение No;
- метод уменьшения большего класса (Undersampling), при его отсутствии – значение No;
- значение сбалансированной точности (Balanced Accuracy).

Поскольку диапазон значений сбалансированной точности отличался в зависимости от исследуемого набора данных, была проведена минимаксная нормализация сбалансированной точности отдельно для каждого из наборов.

Также в обучающем наборе данных имеются категориальные признаки «Type», «Oversampling» и «Undersampling», которые были преобразованы в числовой формат при помощи LabelEncoder.

Набор данных разбивался на обучающую и тестовую выборку с использованием стратификации.

2) Обучение модели

Для повышения качества и устойчивости модели выбрана стратегия оптимизации гиперпараметров с использованием библиотеки Optuna. Протестированы пять различных алгоритмов регрессии: XGB, GradientBoosting, ExtraTrees, RandomForest и LGBM. Они были выбраны с опорой на литературные источники [21, 22] по задаче ранжирования.

При подборе гиперпараметров проводилась 5-кратная кросс-валидация с использование KFold с перемешиванием, модель оценивалась по метрике R^2 (коэффициенту детерминации). Значения средней оценки инвертировались, чтобы оптимизация шла в направлении минимизации.

По результатам подбора гиперпараметров выбраны три алгоритма регрессии, показавшие наилучший результат: XGB, GradientBoosting и ExtraTrees.

Затем проверялось качество модели при объединении выбранных алгоритмов регрессии в ансамбль с использованием бэггинга (Bagging Regressor), стекинга (Stacking Regressor) и голосующего ансамбля (Voting Regressor). По

результатам 5-кратной кросс-валидации и значениям коэффициентов детерминации выбран голосующий ансамбль.

3) Оценка качества модели

Для визуальной оценки качества модели построен график, сравнивающий действительное и предсказанное значение ранга на тестовой выборке. График приводится на рисунке 3.3. Точки находятся вблизи диагонали, что свидетельствует о близости действительных и предсказанных значений и неплохом качестве модели.

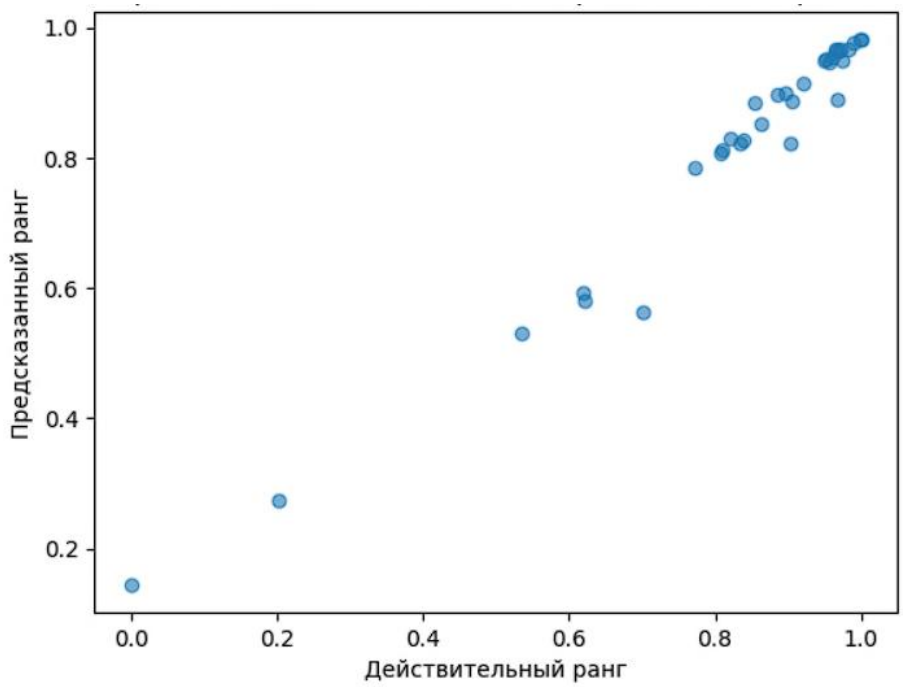


Рисунок 3.3 – Сравнение действительного и предсказанного ранга

Для более точной оценки производительности построенного ансамблевого регрессора применялась 5-кратная кросс-валидация с расчётом среднеквадратической ошибки (MSE, Mean Squared Error), средней абсолютной ошибки (MAE, Mean Absolute Error) и коэффициента детерминации.

Полученные значения приводятся в таблице 3.1.

Таблица 3.1 – Обобщение результатов экспериментальных исследований

Номер блока	MSE	MAE	Коэффициент детерминации
1	-0.01135477	-0.07120671	0.69812307
2	-0.02629933	-0.10227519	0.60890527
3	-0.00695144	-0.04180257	0.65440132
4	-0.01836801	-0.05555495	0.59469433
5	-0.01021363	-0.0535413	0.79880918

Модель в различных блоках демонстрирует разброс ошибок от наименьшей ошибок (в абсолютном значении: 0.00695144) до наибольшей (0.02629933).

Такое различие может указывать на вариабельность распределения данных между блоками – вероятнее всего, она обусловлена небольшим размером обучающего набора данных.

Приблизительное среднее значение (с учётом модулей ошибок) составляет около 0.015. Это говорит о том, что в среднем модель показывает низкую квадратичную ошибку, и это является хорошим показателем при оценке точности предсказаний.

Разброс значений коэффициента детерминации может свидетельствовать о том, что некоторые блоки содержат более явно выраженные зависимости между признаками и целевой переменной, в то время как в других выборках эти зависимости менее выражены. Это может быть связано с неоднородностью данных и небольшим размером обучающего набора.

В целом, несмотря на некоторую вариативность между блоками, общая статистика ошибок (MSE и MAE) и значений R^2 демонстрируют, что модель в целом обладает хорошей предсказательной способностью и адекватно описывает зависимость между признаками и целевой переменной. Это подтверждает применимость модели для дальнейшего ранжирования сочетаний методов балансировки.

4) Методология применения модели.

После тестирования полученной модели выполнено ее сохранение вместе с кодировщиком меток для того, чтобы можно было восстановить исходные названия методов балансировки данных. Также для удобства сохранены все возможные уникальные сочетания методов балансировки из обучающего набора данных.

Это позволит не обучать модель заново каждый раз при загрузке приложения, а воспользоваться уже обученной и сразу переходить к этапу предсказаний.

При использовании приложения для загруженного набора данных определяется набор признаков: размер, минимальное соотношение классов, тип классификации. Тип проходит аналогичное кодирование, как и на этапе обучения модели. Полученный набор признаков объединяется со всеми строками таблицы с сочетаниями методов балансировки.

С использованием обученной ансамблевой модели для каждого сочетания методов балансировки вычисляется прогноз нормализованной сбалансированной точности. По предсказанным значениям выполняется сортировка вариантов по убыванию, и на основе этого ранжирования выбираются топ-10 сочетаний, которые, согласно модели, дадут наилучшие результаты при балансировке классов.

3.4 Интерфейс программного средства

Веб-приложение не требует установки и становится доступно при переходе по адресу сайта https://resample.streamlit.app/. Интерфейс приложения выполнен на английском языке. Из-за того, что целевая аудитория приложения – это специалисты в области машинного обучения, и большинство из них изучает английский язык, понимание интерфейса не вызовет затруднений.

После открытия веб-приложения пользователю предоставляется выбор использовать демонстрационный набор данных (показан на рисунке 3.4) либо загрузить собственный. Загрузка собственного набора данных показана на рисунке 3.5.

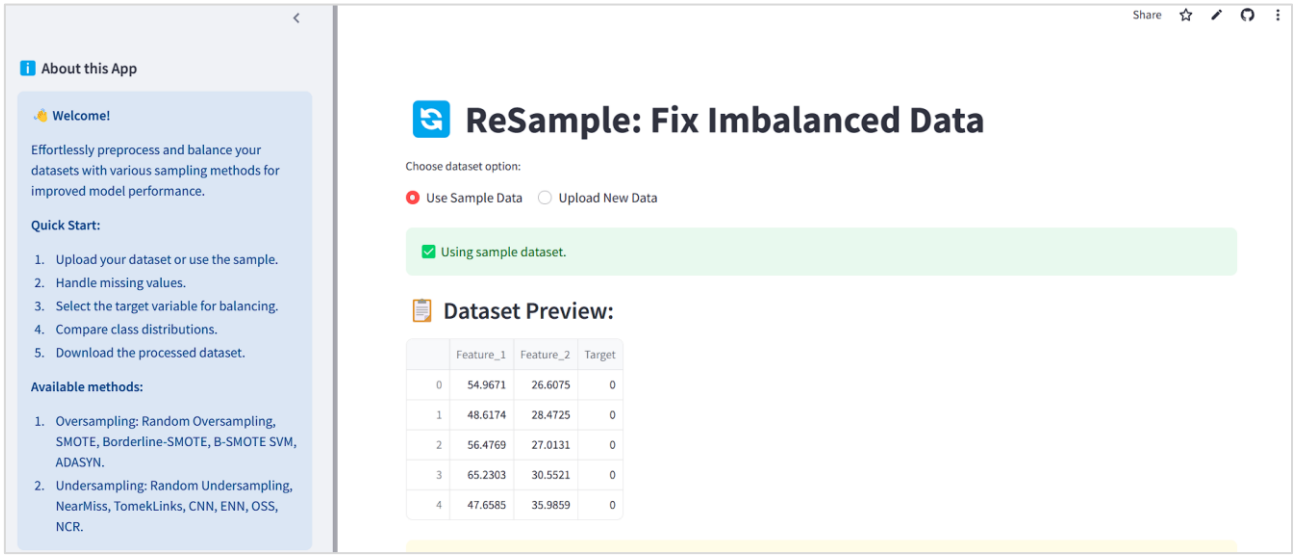


Рисунок 3.4 – Использование демонстрационного набора данных

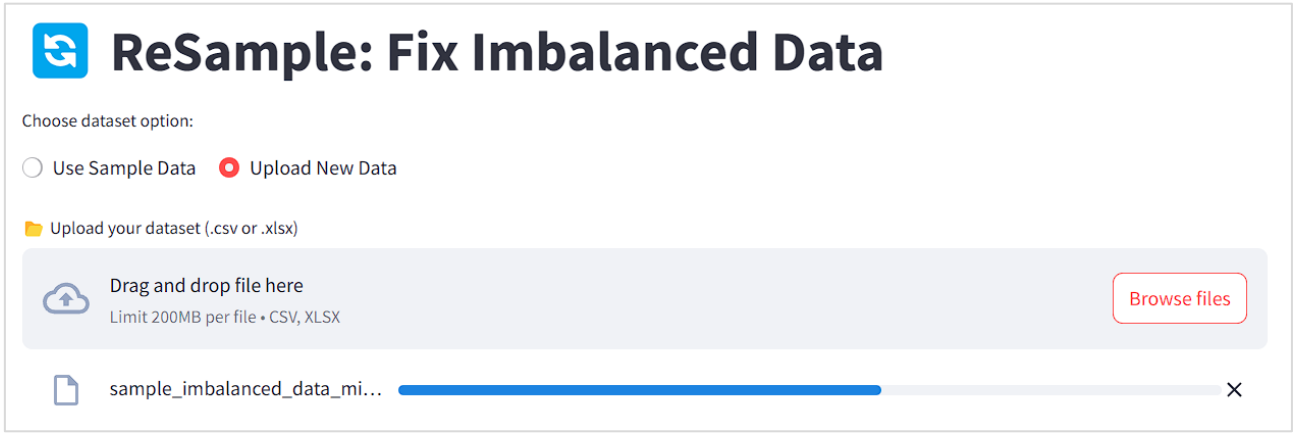


Рисунок 3.5 – Загрузка пользовательского набора данных

После загрузки набора данных выполняется проверка на наличие в нем пропущенных значений. Если такие обнаружены, то пользователю предлагается выбрать стратегию предобработки: удалить строки либо заполнить их модой,

медианой или средним значением. Интерфейс для процесса обработки показан на рисунке 3.6.

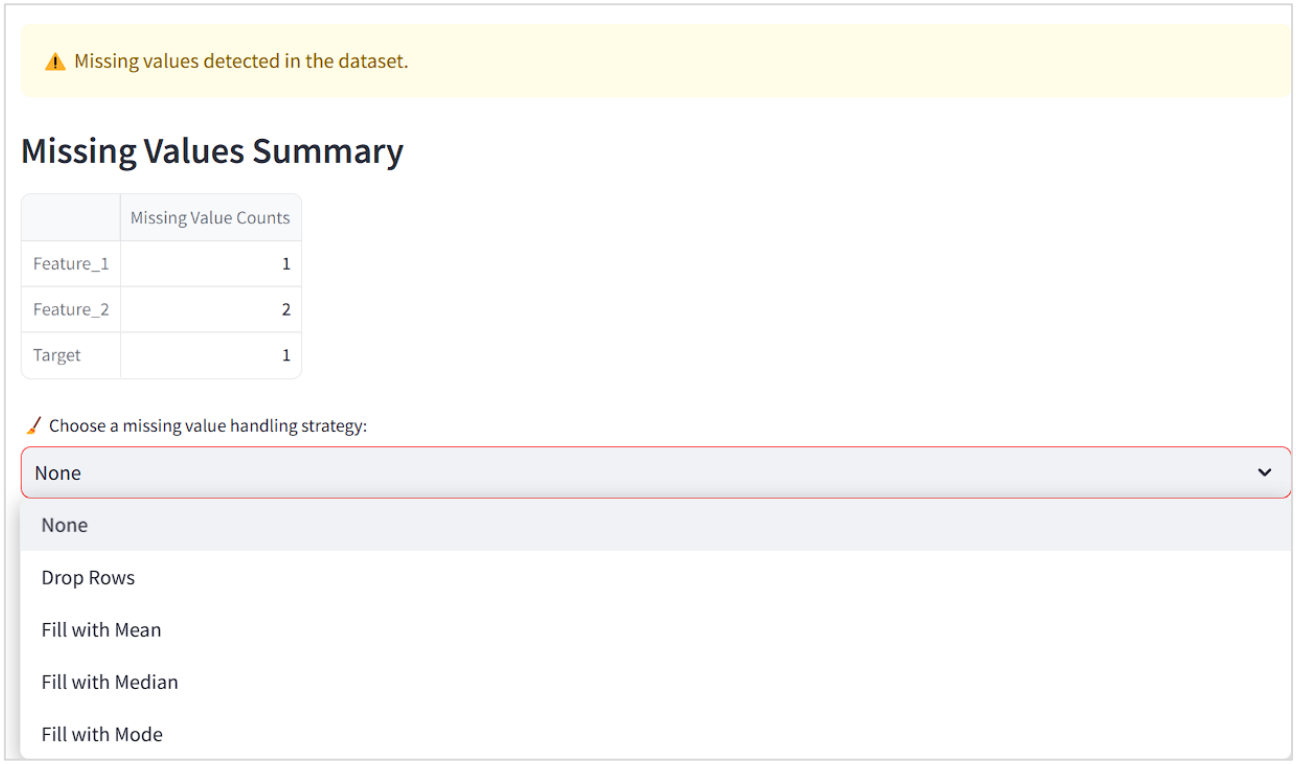


Рисунок 3.6 – Обработка пропущенных значений

При успешной обработке пропущенных значений пользователю будет показано информационное сообщение об этом, уточняющее, какой метод был применен, как на рисунке 3.7.

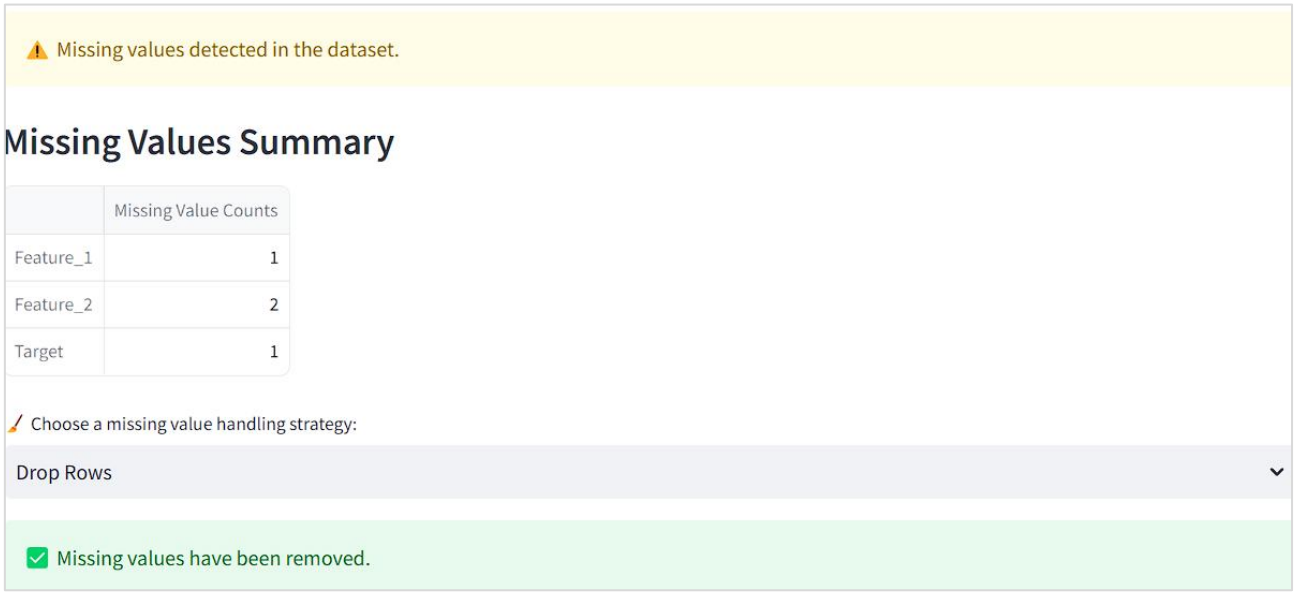


Рисунок 3.7 – Сообщение об успешной обработке пропущенных значений

После выполнения предобработки на экран выводится список рекомендованных методов балансировки данных, основанный на размере набора данных, величине дисбаланса и количестве признаков помимо целевой переменной. Пример такого списка приводится на рисунке 3.8.

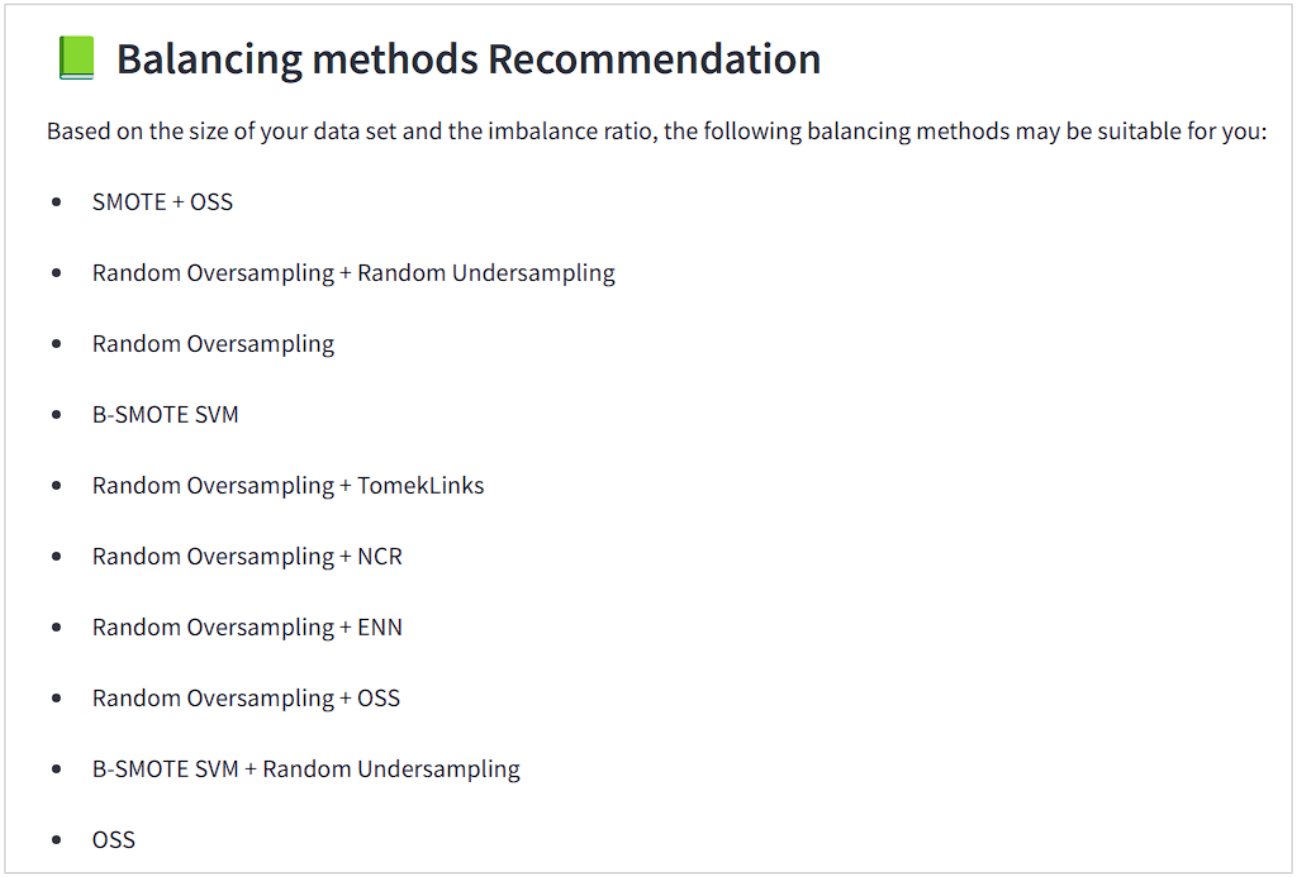


Рисунок 3.8 – Рекомендация методов балансировки

Вне зависимости от того, какие именно методы были рекомендованы пользователю, в дальнейшем он имеет возможность провести балансировку любым из доступных методов. Для этого нужно выбрать желаемое соотношение и метод балансировки, как на рисунке 3.9.

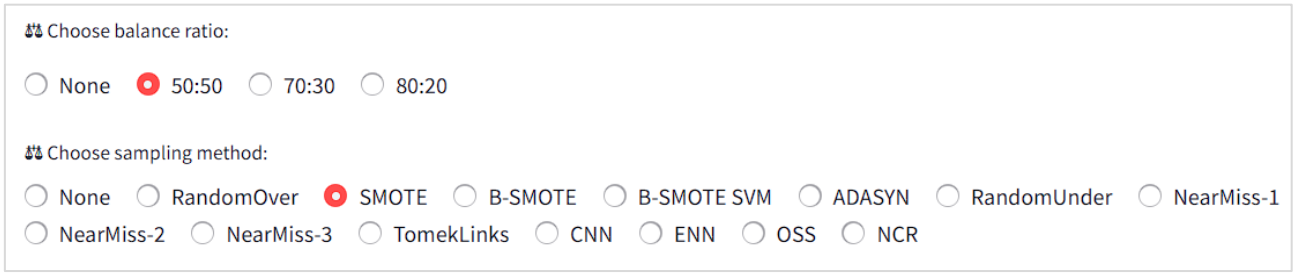


Рисунок 3.9 – Рекомендация методов балансировки

После выполнения балансировки выводятся диаграммы для визуализации соотношения классов до и после балансировки. Пользователь может выбрать как

круговую диаграмму (показанную на рисунке 3.10), так и столбчатую (показанную на рисунке 3.11).

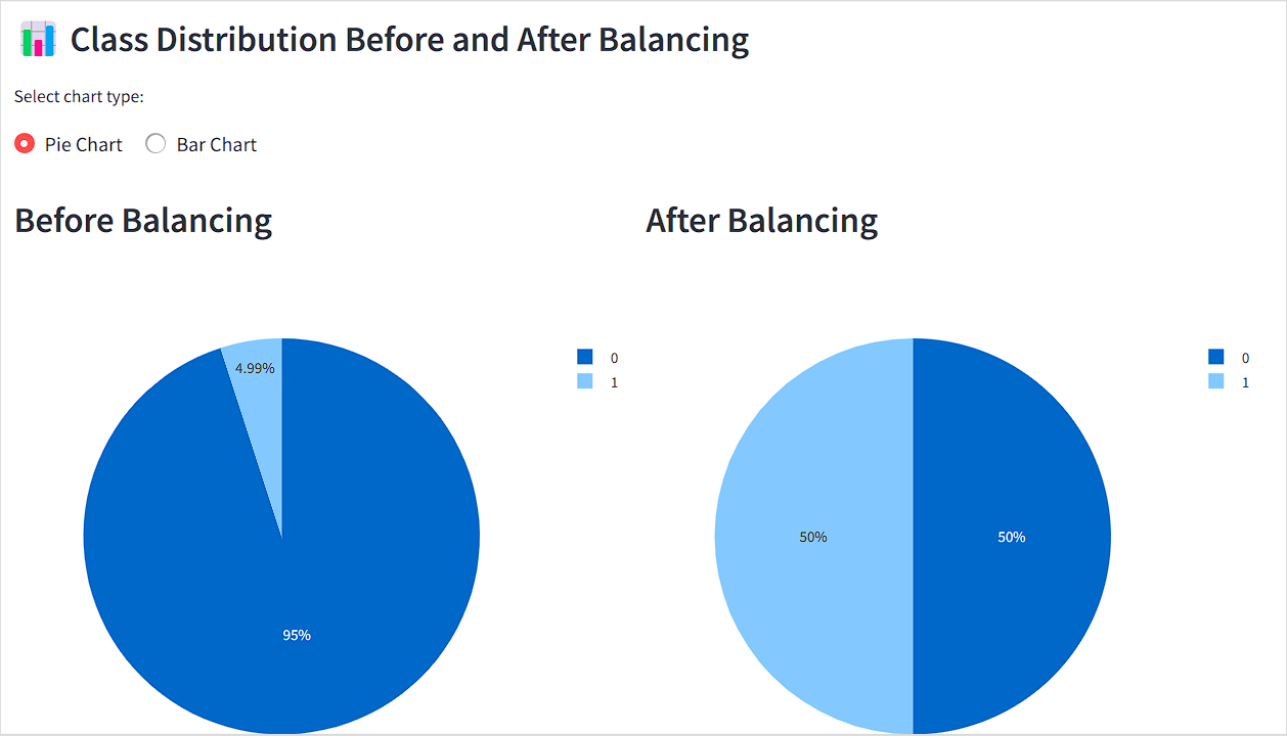


Рисунок 3.10 – Круговая диаграмма с соотношениями классов

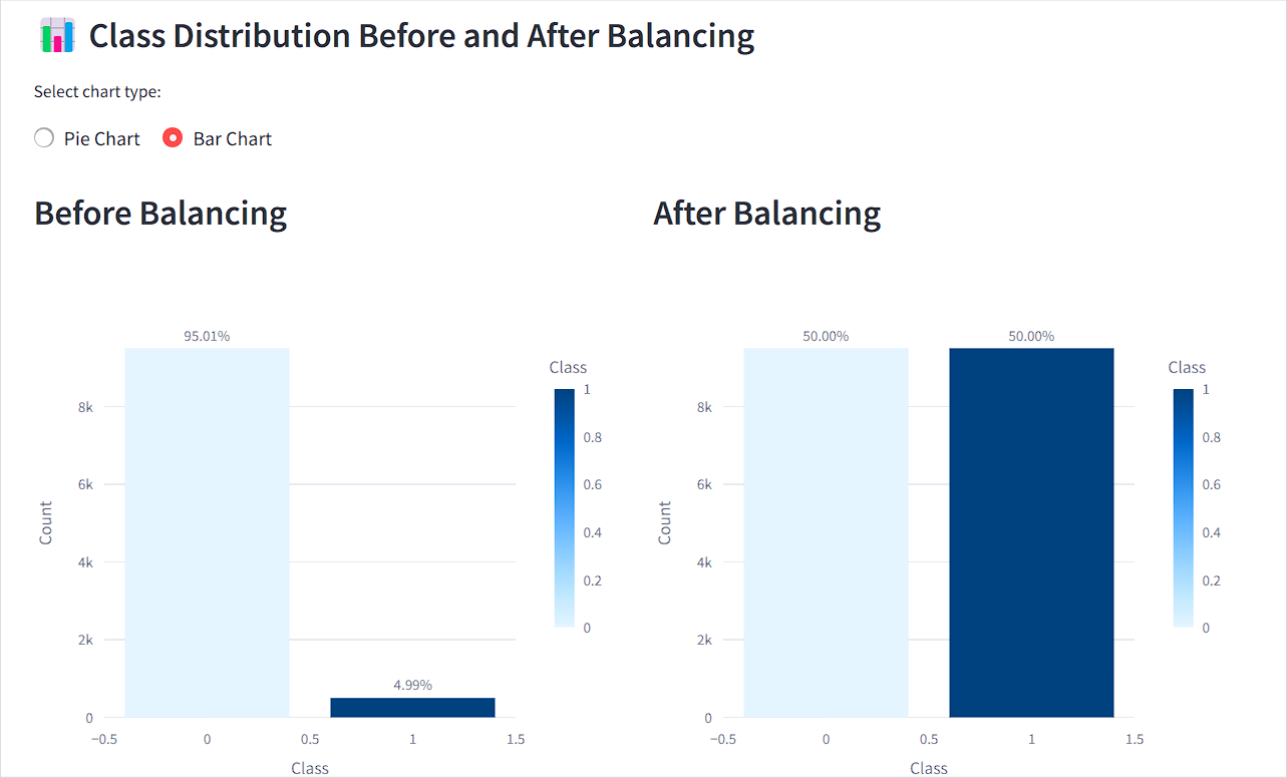


Рисунок 3.11 – Столбчатая диаграмма с соотношениями классов

Также после балансировки выводится основная информация о наборе данных до и после балансировки, и появляется кнопка загрузки сбалансированного набора данных, как на рисунке 3.12. В любой момент работы с веб-приложением можно вернуться на предыдущие этапы и изменить выбор настроек для обработки набора данных.

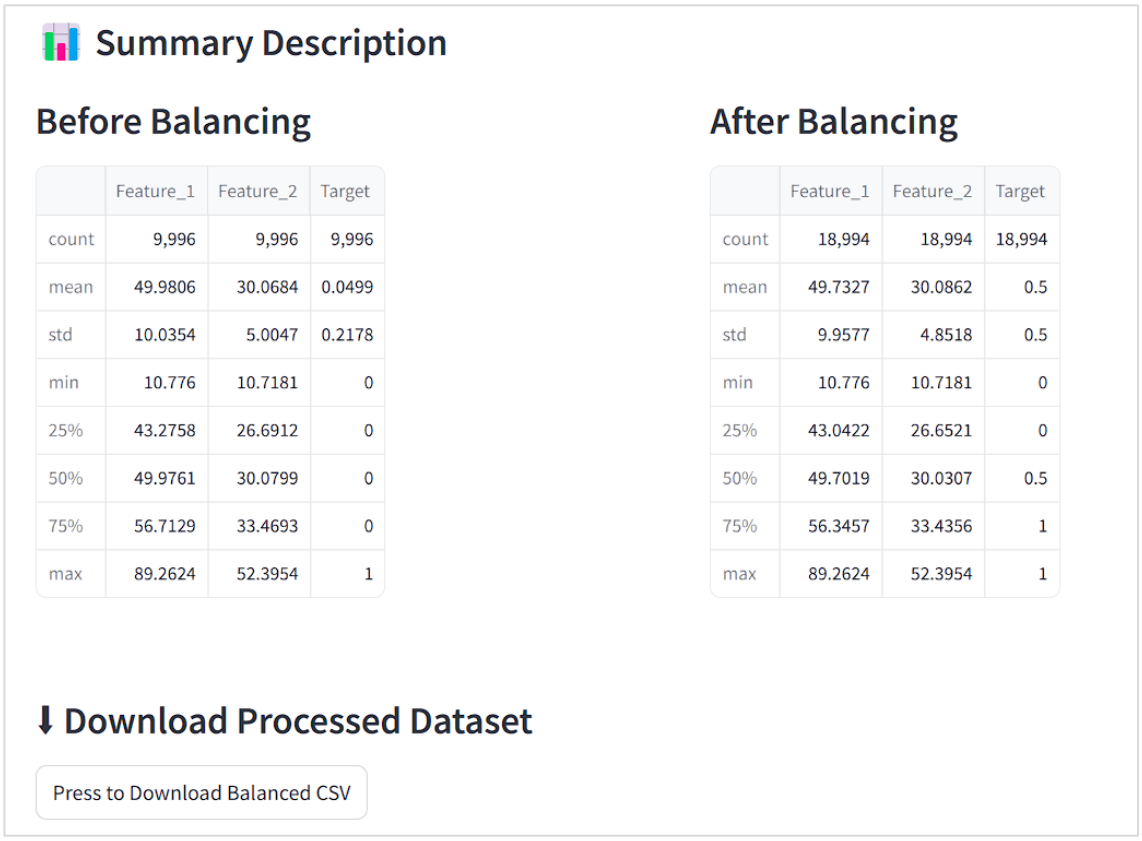


Рисунок 3.12 – Информация о наборе данных и кнопка загрузки

3.5 Выводы по третьей главе

В ходе разработки программного средства выполнена постановка задач, определены основные функции системы: загрузка набора данных, предобработка пропущенных значений, рекомендация методов балансировки, проведение балансировки, визуализация результатов и сохранение сбалансированного набора данных.

Принято проектное решение разработать веб-приложение, поскольку оно дает значительные преимущества в доступности, гибкости и удобстве в контексте решения задач машинного обучения. Для реализации выбран язык программирования Python, для разработки веб-интерфейса – фреймворк Streamlit.

Выполнено логическое моделирование системы: разработана функциональная модель в нотации IDEF0. Спроектирована модель ранжирования для рекомендации алгоритмов балансировки.

Для обучения модели сформирован набор данных с ключевыми признаками: размер набора данных, соотношение классов, тип классификации, используемые методы балансировки и нормализованная сбалансированная точность. Категориальные признаки трансформированы в числовой формат, а разбиение на обучающую и тестовую выборки выполнялось с учетом стратификации, обеспечивающей равномерное распределение классов.

Оптимизация гиперпараметров проводилась с использованием библиотеки Optuna и 5-кратной кросс-валидации. Протестированы и отобраны три лучших алгоритма регрессии: XGB, GradientBoosting и ExtraTrees. Для повышения предсказательной способности модели они объединены в ансамбль с использованием Voting Regressor, который продемонстрировал наилучшие результаты среди протестированных ансамблевых методик.

Анализ среднеквадратической (MSE), средней абсолютной ошибки (MAE), и коэффициента детерминации подтвердил способность модели эффективно описывать зависимости между признаками и целевой переменной. Несмотря на вариативность ошибок между блоками, среднее значение ошибки ≈0.015 свидетельствует о высокой точности предсказаний.

Приложение автоматически формирует признаки загруженного пользователем набора данных и использует обученный ансамблевый регрессор для ранжирования лучших методов балансировки, после чего пользователю предоставляется топ-10 оптимальных комбинаций, которые обеспечивают максимальную сбалансированную точность.

В результате разработан удобный инструмент для обработки набора данных, получения рекомендаций по оптимальным методам балансировки с возможностью выполнить балансировку и сохранить обработанный набор данных.

ЗАКЛЮЧЕНИЕ

В данной дипломной работе проанализированы методы и алгоритмы работы с несбалансированными наборами данных. Из них для экспериментального исследования отобраны широко применяемые методы балансировки данных:

- основанные на увеличении меньшего класса: Random Oversampling, SMOTE, B-SMOTE, B-SMOTE SVM, ADASYN;
- основанные на уменьшении большего класса: Random Undersampling, CNN, Near Miss, Tomek Links, ENN, OSS, NCR.

Определены корректные метрики для оценки моделей в условиях дисбаланса классов, для исследования выбрана сбалансированная точность. Выбраны классические (дерево решений, MLP, k-ближайших соседей, наивный байесовский классификатор) и ансамблевые (случайный лес, градиентный бустинг, XGBoost, AdaBoost и LightGBM) модели для оценки качества классификации.

Выполнен разведочный анализ данных, предобработка и балансировка указанными методами для пяти наборов данных, выбранных с учетом объема, размерности признакового пространства, степени дисбаланса классов и тематической направленности.

Проведено экспериментальное исследование влияния методов балансировки данных на качество классических и ансамблевых моделей как при изолированном применении алгоритмов балансировки, так и при их комбинации. Определено оптимальное соотношение классов после балансировки и оценено влияние подбора гиперпараметров при помощи Optuna.

На основе результатов экспериментов обучена модель ранжирования, позволяющая определить наилучшие алгоритмы балансировки и их комбинации для конкретного набора данных в зависимости от его размера и величины дисбаланса. Модель ранжирования интегрирована в качестве рекомендательного модуля в веб-приложение, предоставляющее возможность выполнить предобработку и балансировку конкретного набора данных.

Результаты работы предназначены для специалистов в области машинного обучения и могут использоваться для оптимизации выбора подходящих методов работы с несбалансированными данными для конкретных практических задач в сферах, где данная проблема является критической: медицине, биоинформатике, инженерии, безопасности, бизнесе и других.

Направлением для дальнейшего исследования может стать расширение исследуемых наборов данных для улучшения эффективности модели рекомендаций; добавление более широких возможностей по предобработке данных, интеграция актуальных моделей машинного обучения и подбора гиперпараметров для того, чтобы веб-приложение представляло собой единую среду с возможностями проведения всех этапов работы с набором данных в задаче классификации.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вертинская А. Е. Методика сравнения методов машинного обучения в зависимости от различных параметров задачи: магистерская диссертация : дис. / Антонина Евгеньевна Вертинская ; БГУ, ФПМИ. – Минск, 2020. – 41 с.
2. Клицунова Е, Сравнительный анализ методов балансировки данных для задач машинного обучения / Е. Клицунова, М. М. Лукашевич // BIG DATA и анализ высокого уровня = BIG DATA and Advanced Analytics : сборник научных статей XI Международной научно-практической конференции, Минск, 23–24 апреля 2025 года / редкол.: В. А. Богуш [и др.]. – Минск : БГУИР, 2025. – С. 74-83
3. Михайличенко, А. А. Аналитический обзор методов оценки качества алгоритмов классификации в задачах машинного обучения / А. А. Михайличенко // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. – 2022. – Т. 311, №4. – С. 52–59. DOI: 10.53598/2410-3225-2022-4-311-52-59
4. Старовойтов, В. В. Как оценивать результаты классификации несбалансированных больших данных? / В. В. Старовойтов, Ю. И. Голуб // BIG DATA and Advanced Analytics = BIG DATA и анализ высокого уровня : сборник научных статей VII Международной научно-практической конференции, Минск, 19-20 мая 2021 года / редкол.: В. А. Богуш [и др.]. – Минск : Бестпринт, 2021. – С. 272–283.
5. Старовойтов, В. В. Об оценке результатов классификации несбалансированных данных по матрице ошибок / В. В. Старовойтов, Ю. И. Голуб // Информатика. – 2021. – Т. 18, № 1. – С. 61–71. DOI: 10.37661/1816-0301-2021-18-1-61-71
6. Старовойтов, В. В. Сравнительный анализ оценок качества бинарной классификации / В. В. Старовойтов, Ю. И. Голуб // Информатика. – 2020. Т. 17, №1. – С. 87–101. DOI: 10.37661/1816-0301-2020-17-1-87-101
7. Brownlee, J. Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning / J. Brownlee – Machine Learning Mastery, 2020. – 446 p.
8. Buda, M. A systematic study of the class imbalance problem in convolutional neural networks / M. Buda, A. Maki, M. A. Mazurowski // Neural networks. – 2018. – Т. 106. – P. 249–259. DOI: 10.48550/arXiv.1710.05381
9. Fernández, A Learning from imbalanced data sets / A. Fernández, S. García, M. Galar [et al.] // Cham: Springer – Vol. 10, № 2018. – 385 p. DOI: 10.1007/978-3-319-98074-4
10. Ferri, C. An experimental comparison of performance measures for classification / C. Ferri, J. Hernández-Orallo, R. Modroiu // Pattern recognition letters. – 2009. – Т. 30, № 1. – P. 27–38. DOI: 10.1016/j.patrec.2008.08.010

11. Ganganwar, V. An overview of classification algorithms for imbalanced datasets / V. Ganganwar // International Journal of Emerging Technology and Advanced Engineering. – 2012. – Vol. 2, №. 4. – P. 42–47.

12. Google Developers, Datasets: Imbalanced datasets // Machine Learning Crash Course. Mode of access: https://developers.google.com/machine-learning/crash-course/overfitting/imbalanced-datasets – Date of access: 24.05.2025

13. He, H. Imbalanced learning: foundations, algorithms, and applications / H. He, Y. Ma – John Wiley & Sons, 2013. – 224 p. DOI: 10.1002/9781118646106

14. Kaggle: Your Home for Data Science // Mode of access: https://www.kaggle.com – Date of access: 19.05.2025

15. Kotsiantis, S. Handling imbalanced datasets: A review / S. Kotsiantis, D. Kanellopoulos, P. Pintelas // GESTS international transactions on computer science and engineering. – 2006. – Vol. 30, №. 1. – P. 25–36.

16. Krawczyk, B. Learning from imbalanced data: open challenges and future directions / B. Krawczyk // Progress in artificial intelligence. – 2016. – Vol. 5, №. 4. – P. 221–232. DOI: 10.1007/s13748-016-0094-0

17. Kubat, M. Addressing the curse of imbalanced training sets: one-sided selection / M. Kubat, S. Matwin // Icml. – 1997. – Vol. 97, №. 1. – P. 179.

18. Kuhn, M. Applied predictive modeling / M. Kuhn, K. Johnson – Springer Science & Business Media, 2013. – 615 p. DOI: 10.1007/978-1-4614-6849-3

19. Kumar P. et al. Classification of imbalanced data: review of methods and applications / P. Kumar, R. Bhatnagar, K. Gaur [et al] // IOP conference series: materials science and engineering. – IOP Publishing, 2021. – Vol. 1099, №. 1. – P. 012077. DOI: 10.1088/1757-899X/1099/1/012077

20. LemaÃŽtre, G. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning / G. LemaÃŽtre, F. Nogueira, C. K. Aridas // Journal of machine learning research. – 2017. – Vol. 18, №. 17. – P. 1–5. DOI: 10.48550/arXiv.1609.06570

21. Li H. A short introduction to learning to rank // IEICE TRANSACTIONS on Information and Systems. – 2011. – Vol. 94, №. 10. – P. 1854–1862. DOI: 10.1587/transinf.E94.D.1854

22. Liu T. Y. et al. Learning to rank for information retrieval // Foundations and Trends in Information Retrieval. – 2009. – Vol. 3, №. 3. – P. 225–331. DOI: 10.1561/1500000016

23. Paula Branco, L. T. A survey of predictive modelling under imbalanced distributions / L. T. Paula Branco, R. Ribeiro // ACM Comput Surveys. – 2016. – Vol. 49. – P. 31–48. DOI: 10.48550/arXiv.1505.01658

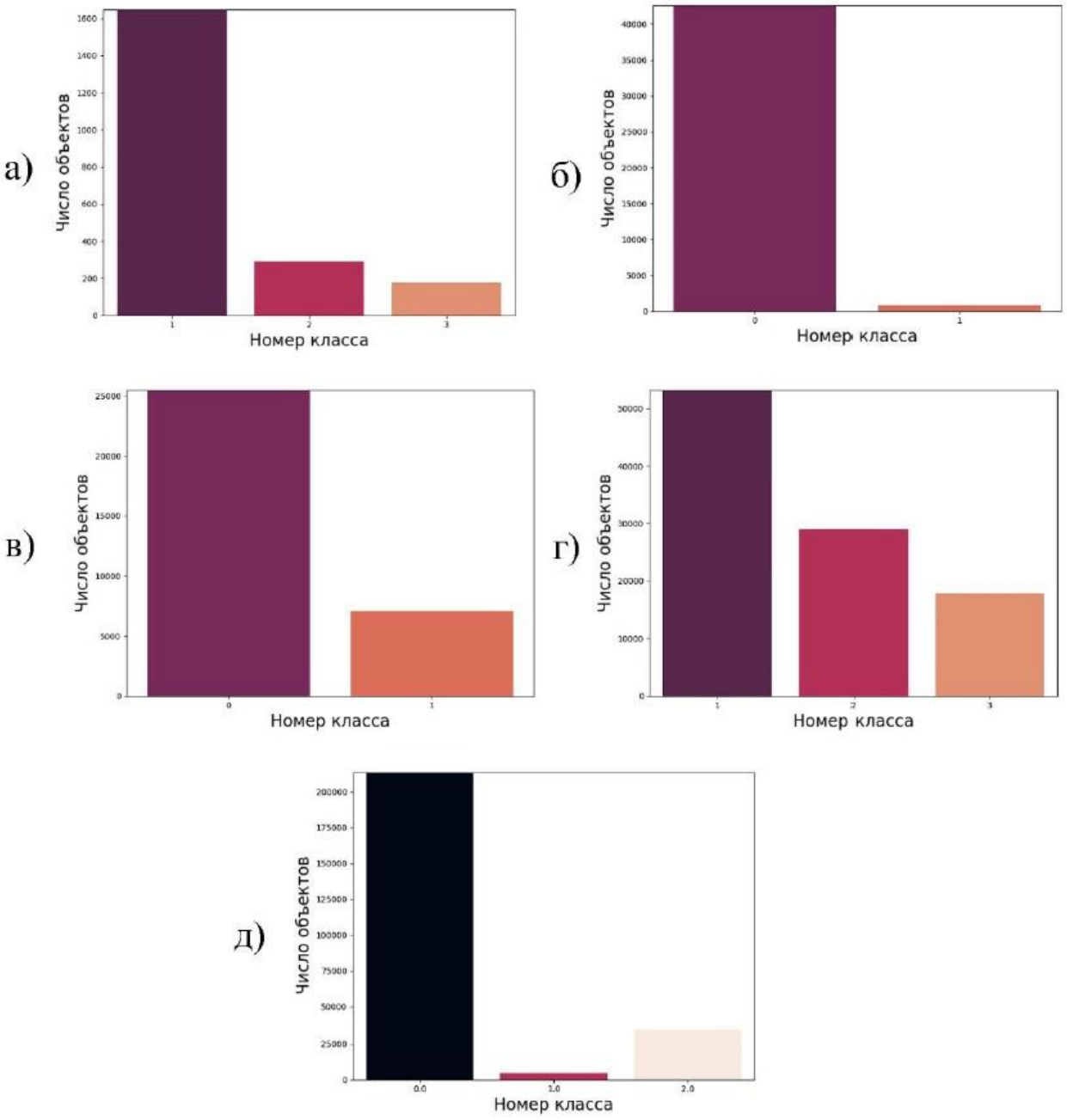
24. Sokolova, M. A systematic analysis of performance measures for classification tasks / M. A. Sokolova, G. Lapalme // Information processing & management. – 2009. – Vol. 45, №. 4. – P. 427–437. DOI: 10.1016/j.ipm.2009.03.002

25. Sun, Y. Classification of imbalanced data: A review / Y. Sun, A. K. C. Wong, M. S. Kamel // International journal of pattern recognition and artificial intelligence. – 2009. – Vol. 23, №. 04. – P. 687–719. DOI: 10.1142/S0218001409007326

26. Yang, Y. A survey on ensemble learning under the era of deep learning / Y. Yang, H. Lv, N. Chen // Artificial Intelligence Review. – 2023. – Vol. 56, №. 6. – P. 5545-5589. DOI: 10.1109/ACCESS.2022.3207287

ПРИЛОЖЕНИЕ А

СООТНОШЕНИЕ КЛАССОВ В ВЫБРАННЫХ НАБОРАХ ДАННЫХ



а) набор «Классификация здоровья плода» б) набор «Прогнозирование церебрального инсульта» в) набор «Кредитный риск» г) набор «Классификация кредитного рейтинга» д) набор «Показатели здоровья при диабете»

Рисунок А.1 – Соотношение классов в наборах данных

ПРИЛОЖЕНИЕ Б

РЕЗУЛЬТАТЫ БАЛАНСИРОВКИ ДАННЫХ ОДНИМ МЕТОДОМ

Полужирным шрифтом обозначены максимальные значения точности для набора в целом. Более бледный цвет шрифта использовался в тех случаях, когда наблюдалось отсутствие повышения точности классификации по сравнению с точностью для исходного набора данных.

Таблица Б.1 – Результаты экспериментов для набора данных «Классификация здоровья плода» (увеличение меньшего класса)

Модель	Исходный набор	Random Oversampling	SMOTE	B-SMOTE	B-SMOTE SVM	ADASYN
kNN	0,79	0,81	0,85	0,85	0,85	0,83
DT	0,87	0,88	0,86	0,86	0,86	0,87
MLP	0,67	0,69	0,85	0,80	0,85	0,82
GaussianNB	0,78	0,79	0,79	0,83	0,77	0,84
RF	0,89	**0,93**	**0,93**	**0,93**	0,91	**0,92**
GB	0,91	0,86	0,90	0,89	0,88	0,86
AdaBoost	0,82	0,91	0,92	0,90	0,91	**0,92**
XGBoost	0,89	**0,92**	**0,92**	**0,92**	**0,92**	**0,92**
LightGBM	**0,92**	**0,92**	**0,92**	**0,92**	**0,92**	**0,92**

Таблица Б.2 – Результаты экспериментов для набора данных «Классификация здоровья плода» (уменьшение большего класса)

Модель	Исходный набор	Random Undersampling	NM-1	NM-3	CNN	Tomek Links	OSS
kNN	0,79	0,85	0,75	0,71	0,75	0,79	0,83
DT	0,87	0,87	0,85	0,83	0,80	0,84	0,87
MLP	0,67	0,82	0,77	0,76	0,77	0,67	0,84
GaussianNB	0,78	0,79	0,69	0,75	0,77	0,78	0,77
RF	0,89	0,91	0,90	0,88	0,87	0,88	0,91
GB	0,91	**0,93**	0,92	0,90	0,90	0,90	**0,93**
AdaBoost	0,82	0,78	0,70	0,84	0,79	0,81	0,88
XGBoost	0,89	**0,93**	0,91	0,90	0,87	0,89	**0,94**
LightGBM	**0,92**	**0,92**	0,88	0,87	0,86	**0,93**	**0,93**

Таблица Б.3 – Результаты экспериментов для набора данных «Прогнозирование церебрального инсульта» (увеличение меньшего класса)

Модель	Исходный набор	Random Oversampling	SMOTE	B-SMOTE	B-SMOTE SVM	ADASYN
kNN	0,50	0,54	0,61	0,57	0,56	0,60
DT	0,51	0,52	0,55	0,54	0,54	0,54
MLP	0,50	0,73	0,65	0,66	0,56	0,64

Продолжение таблицы Б.3

Модель	Исходный набор	Random Oversampling	SMOTE	B-SMOTE	B-SMOTE SVM	ADASYN
GaussianNB	0,65	**0,74**	0,72	0,72	0,71	0,72
RF	0,50	0,51	0,53	0,52	0,53	0,53
GB	0,50	**0,77**	0,65	0,63	0,60	0,67
AdaBoost	0,50	**0,78**	0,67	0,67	0,62	0,67
XGBoost	0,50	0,60	0,56	0,54	0,53	0,57
LightGBM	0,50	0,66	0,57	0,55	0,53	0,56

Таблица Б.4 – Результаты экспериментов для набора данных «Прогнозирование церебрального инсульта» (уменьшение большего класса)

Модель	Исходный набор	Random Undersampling	NM-2	NM-3	Tomek Links	ENN	OSS	NCR
kNN	0,50	**0,74**	0,51	0,65	0,50	0,50	0,50	0,50
DT	0,51	0,69	0,50	0,54	0,52	0,52	0,53	0,53
MLP	0,50	**0,75**	0,67	0,66	0,50	0,50	0,50	0,50
GaussianNB	0,65	**0,74**	0,52	0,71	0,65	0,65	0,59	0,66
RF	0,50	**0,76**	0,51	0,66	0,50	0,50	0,50	0,50
GB	0,50	**0,77**	0,50	0,70	0,50	0,50	0,50	0,50
AdaBoost	0,50	**0,77**	0,51	0,63	0,50	0,50	0,50	0,50
XGBoost	0,50	**0,74**	0,50	0,66	0,50	0,50	0,50	0,51
LightGBM	0,50	**0,74**	0,50	0,64	0,50	0,50	0,50	0,50

Таблица Б.5 – Результаты экспериментов для набора данных «Кредитный риск» (увеличение меньшего класса)

Модель	Исходный набор	Random Oversampling	SMOTE	B-SMOTE	B-SMOTE SVM	ADASYN
kNN	0,72	0,74	0,73	0,74	0,74	0,73
DT	0,84	0,84	0,81	0,81	0,82	0,80
MLP	0,50	0,72	0,73	0,73	0,72	0,75
GaussianNB	0,67	0,72	0,69	0,68	0,71	0,68
RF	**0,85**	**0,86**	0,85	0,84	0,85	0,84
GB	0,84	0,84	0,83	0,84	0,84	0,84
AdaBoost	0,81	0,83	0,79	0,78	0,80	0,78
XGBoost	**0,86**	**0,88**	**0,86**	**0,86**	**0,86**	**0,87**
LightGBM	**0,86**	**0,87**	**0,86**	**0,86**	**0,86**	**0,86**

Таблица Б.6 – Результаты экспериментов для набора данных «Кредитный риск» (уменьшение большего класса)

Модель	Исходный набор	Random Undersampling	NM-3	CNN	Tomek Links	ENN	OSS	NCR
kNN	0,72	0,74	0,70	0,73	0,73	0,75	0,72	0,74
DT	0,84	0,82	0,79	0,78	0,84	0,83	0,83	0,83
MLP	0,50	0,52	0,50	0,50	0,69	0,69	0,50	0,50
GaussianNB	0,67	0,71	0,71	0,71	0,68	0,70	0,69	0,70
RF	**0,85**	0,84	0,85	0,84	**0,85**	0,84	**0,85**	0,84

Продолжение таблицы Б.6

Модель	Исходный набор	Random Undersampling	NM-3	CNN	Tomek Links	ENN	OSS	NCR
GB	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84
AdaBoost	0,81	0,82	0,82	0,83	0,82	0,82	0,82	0,82
XGBoost	**0,86**	**0,87**	**0,86**	**0,86**	**0,87**	**0,87**	**0,87**	**0,87**
LightGBM	**0,86**	**0,86**	**0,86**	**0,86**	**0,86**	**0,86**	**0,86**	**0,86**

Таблица Б.7 – Результаты экспериментов для набора данных «Классификация кредитного рейтинга» (увеличение меньшего класса)

Модель	Исходный набор	Random Oversampling	SMOTE	B-SMOTE	B-SMOTE SVM	ADASYN
kNN	0,71	0,75	0,76	0,76	0,75	0,77
DT	0,73	0,73	0,75	0,74	0,75	0,74
MLP	0,59	0,50	0,53	0,59	0,45	0,67
GaussianNB	0,64	0,63	0,63	0,63	0,63	0,63
RF	**0,80**	**0,82**	**0,82**	**0,82**	**0,82**	**0,82**
GB	0,71	0,72	0,73	0,73	0,73	0,73
AdaBoost	0,63	0,68	0,70	0,69	0,70	0,69
XGBoost	0,76	**0,77**	**0,78**	**0,78**	**0,77**	**0,78**
LightGBM	0,74	0,75	0,76	0,76	0,76	0,75

Таблица Б.8 – Результаты экспериментов для набора данных «Классификация кредитного рейтинга» (уменьшение большего класса)

Модель	Исход-ный набор	Random Undersampling	NM-3	CNN	Tomek Links	ENN	OSS	NCR
kNN	0,71	0,68	0,58	0,56	0,71	0,67	0,64	0,71
DT	0,73	0,75	0,66	0,64	0,76	0,76	0,72	0,78
MLP	0,59	0,58	0,39	0,50	0,59	0,57	0,65	0,45
GaussianNB	0,64	0,63	0,67	0,65	0,64	0,64	0,64	0,64
RF	**0,80**	**0,83**	**0,78**	0,76	0,82	0,78	**0,78**	**0,81**
GB	0,71	0,73	0,71	0,70	0,72	0,72	0,71	0,73
AdaBoost	0,63	0,72	0,67	0,64	0,68	0,69	0,66	0,70
XGBoost	0,76	0,78	0,74	0,72	0,78	0,77	0,75	0,79
LightGBM	0,74	0,76	0,73	0,72	0,75	0,75	0,74	0,76

Таблица Б.9 – Результаты экспериментов для набора данных «Показатели здоровья при диабете» (увеличение меньшего класса)

Модель	Исходный набор	Random Oversampling	SMOTE	B-SMOTE	B-SMOTE SVM	ADASYN
kNN	0,38	0,38	0,44	**0,45**	0,44	0,44
DT	0,40	0,39	0,40	0,40	0,39	0,40
MLP	0,39	0,43	0,40	0,44	**0,47**	0,42
GaussianNB	**0,45**	**0,47**	**0,49**	**0,48**	**0,49**	**0,49**
RF	0,39	0,38	0,39	0,39	0,39	0,39

Продолжение таблицы Б.9

Модель	Исходный набор	Random Oversampling	SMOTE	B-SMOTE	B-SMOTE SVM	ADASYN
GB	0,39	**0,45**	**0,45**	**0,45**	**0,45**	0,44
AdaBoost	0,39	**0,45**	**0,46**	**0,47**	**0,46**	**0,46**
XGBoost	0,39	0,44	0,40	0,40	0,40	0,40
LightGBM	0,38	0,45	0,41	0,40	0,41	0,40

Таблица Б.10 – Результаты экспериментов для набора данных «Показатели здоровья при диабете» (уменьшение большего класса)

Модель	Исходный набор	Random Undersampling	CNN	Tomek Links	ENN	OSS	NCR
kNN	0,38	0,43	0,34	0,38	0,35	0,36	0,37
DT	0,40	0,40	0,36	0,40	0,39	0,39	0,39
MLP	0,39	0,43	0,39	0,38	0,37	0,38	0,37
GaussianNB	0,45	**0,47**	0,45	0,45	0,44	0,45	0,44
RF	0,39	0,41	0,38	0,38	0,36	0,38	0,37
GB	0,39	0,42	0,39	0,38	0,37	0,38	0,37
AdaBoost	0,39	0,41	0,39	0,38	0,38	0,38	0,37
XGBoost	0,39	0,42	0,38	0,38	0,38	0,38	0,37
LightGBM	0,38	0,42	0,37	0,38	0,38	0,38	0,37

ПРИЛОЖЕНИЕ В

РЕЗУЛЬТАТЫ БАЛАНСИРОВКИ ПРИ КОМБИНИРОВАНИИ МЕТОДОВ

Полужирным шрифтом обозначены максимальные значения точности для набора в целом. Более бледный цвет шрифта использовался в тех случаях, когда наблюдалось отсутствие повышения точности классификации по сравнению с точностью для исходного набора данных. Сокращениями RF и GB обозначаются случайный лес и градиентный бустинг соответственно.

Таблица В.1 – Результаты экспериментов для набора данных «Классификация здоровья плода»

Метод увеличения меньшего класса	Метод уменьшения большего класса	RF	GB	XGBoost	LightGBM
-		0,885	0,908	0,911	0,920
Random Oversampling	-	0,914	0,929	**0,939**	0,918
	Random Undersampling	0,927	**0,940**	**0,939**	**0,944**
	NearMiss-1	0,919	**0,941**	**0,945**	0,936
	TomekLinks	0,917	**0,947**	0,935	0,927
	ENN	0,929	**0,943**	0,937	**0,938**
	OSS	0,937	0,936	**0,947**	**0,938**
SMOTE	-	0,907	0,912	0,927	0,927
	Random Undersampling	0,929	0,933	**0,939**	**0,943**
	NearMiss-1	0,923	0,936	0,931	**0,945**
	TomekLinks	0,919	**0,938**	0,934	**0,939**
	ENN	0,935	**0,943**	0,935	**0,941**
	OSS	**0,949**	0,935	0,931	**0,950**
B-SMOTE	-	0,903	0,904	0,908	0,928
	Random Undersampling	0,929	**0,939**	0,932	0,936
	NearMiss-1	0,918	0,930	0,932	0,928
	TomekLinks	0,914	0,925	0,929	0,932
	ENN	0,924	0,934	0,934	0,932
	OSS	0,920	0,922	0,922	0,916
B-SMOTE SVM	-	0,894	0,921	0,919	0,913
	Random Undersampling	0,914	0,931	**0,940**	**0,941**
	NearMiss-1	0,922	**0,939**	0,929	**0,939**
	TomekLinks	0,916	0,933	0,929	0,931
	ENN	0,927	0,933	0,935	0,930
	OSS	0,925	0,927	0,925	0,931
ADASYN	-	0,907	0,911	0,905	0,910
	Random Undersampling	0,926	0,931	0,935	**0,938**
	NearMiss-1	0,924	0,931	0,927	0,934
	TomekLinks	0,920	0,934	0,931	**0,946**
	ENN	0,923	0,933	0,932	0,935
	OSS	0,919	0,932	0,927	0,924

Таблица В.2 – Результаты экспериментов для набора данных «Прогнозирование церебрального инсульта»

Метод увеличения меньшего класса	Метод уменьшения большего класса	MLP	GaussianNB	GB	AdaBoost
-		0,500	0,650	0,502	0,500
Random Oversampling	-	**0,737**	**0,749**	**0,766**	**0,773**
	Random Undersampling	**0,762**	**0,748**	**0,776**	**0,780**
	ENN	**0,752**	**0,753**	**0,767**	**0,777**
SMOTE	-	0,650	0,716	0,665	0,663
	Random Undersampling	0,691	0,721	0,669	0,679
	ENN	0,686	**0,739**	0,685	0,706
	OSS	0,636	0,566	0,643	0,685
B-SMOTE	-	0,651	0,726	0,641	0,671
	Random Undersampling	0,702	**0,729**	0,649	0,685
	ENN	0,675	**0,733**	0,660	0,695
	OSS	0,688	0,643	0,678	**0,737**
B-SMOTE SVM	-	0,649	0,712	0,600	0,620
	Random Undersampling	0,710	**0,735**	0,672	0,700
	ENN	0,659	**0,729**	0,637	0,677
	OSS	0,685	0,641	0,668	0,706
ADASYN	-	0,658	0,713	0,650	0,679
	Random Undersampling	0,690	0,721	0,678	0,681
	ENN	0,707	**0,738**	0,682	0,707
	OSS	0,624	0,567	0,644	0,686

Таблица В.3 – Результаты экспериментов для набора данных «Кредитный риск»

Метод увеличения меньшего класса	Метод уменьшения большего класса	RF	GB	AdaBoost	XG Boost	LightGBM
-		0,847	0,839	0,805	**0,859**	**0,854**
Random Oversampling	-	**0,854**	0,844	0,826	**0,874**	**0,863**
	Random Undersampling	0,845	0,840	0,813	**0,854**	**0,853**
	TomekLinks	**0,856**	0,846	0,830	**0,873**	**0,864**
	ENN	0,848	0,839	0,829	**0,868**	**0,862**
	OSS	**0,854**	0,847	0,829	**0,871**	**0,865**
	NCR	**0,852**	0,841	0,826	**0,872**	**0,865**
SMOTE	-	0,847	0,831	0,797	**0,864**	**0,858**
	Random Undersampling	0,849	0,837	0,820	**0,867**	**0,859**
	TomekLinks	**0,854**	0,843	0,810	**0,867**	**0,860**
	ENN	0,845	0,833	0,816	**0,862**	**0,855**
	OSS	**0,852**	0,841	0,814	**0,865**	**0,859**
	NCR	0,845	0,836	0,818	**0,862**	**0,859**
B-SMOTE	-	0,842	0,832	0,787	**0,861**	**0,855**
	Random Undersampling	0,847	0,838	0,817	**0,868**	**0,858**
	TomekLinks	**0,851**	0,844	0,811	**0,865**	**0,857**
	ENN	0,842	0,836	0,813	**0,860**	0,852
	OSS	**0,851**	0,845	0,815	**0,865**	**0,858**
	NCR	0,842	0,839	0,818	**0,864**	**0,855**
B-SMOTE SVM	-	**0,854**	0,844	0,826	**0,874**	**0,863**
	Random Undersampling	**0,852**	0,840	0,821	**0,870**	**0,861**
	TomekLinks	**0,854**	0,843	0,819	**0,864**	**0,863**

Продолжение таблицы В.3

Метод увеличения меньшего класса	Метод уменьшения большего класса	RF	GB	AdaBoost	XG Boost	LightGBM
B-SMOTE SVM	ENN	0,843	0,834	0,820	**0,861**	**0,855**
	OSS	**0,854**	0,841	0,821	**0,866**	**0,858**
	NCR	0,849	0,837	0,822	**0,867**	**0,857**
ADASYN	-	**0,852**	0,836	0,808	**0,860**	**0,857**
	Random Undersampling	0,845	0,836	0,814	**0,867**	**0,860**
	TomekLinks	**0,852**	0,841	0,804	**0,867**	**0,858**
	ENN	0,843	0,835	0,813	**0,856**	**0,853**
	OSS	**0,850**	0,840	0,811	**0,866**	**0,857**
	NCR	0,842	0,837	0,814	**0,860**	**0,857**

Таблица В.4 – Результаты экспериментов для набора данных «Классификация кредитного рейтинга»

Метод увеличения меньшего класса	Метод уменьшения большего класса	RF	XGBoost	LightGBM
-		**0,808**	0,762	0,744
Random Oversampling	-	**0,826**	**0,788**	0,761
	Random Undersampling	**0,834**	**0,788**	0,763
	TomekLinks	**0,825**	**0,787**	0,764
	ENN	**0,813**	0,782	0,760
	NCR	**0,831**	**0,791**	0,767
SMOTE	-	**0,822**	0,779	0,757
	Random Undersampling	**0,830**	0,783	0,760
	TomekLinks	**0,824**	0,782	0,760
	ENN	**0,809**	0,781	0,760
	NCR	**0,827**	**0,787**	0,766
B-SMOTE	-	**0,819**	0,780	0,759
	Random Undersampling	**0,831**	0,783	0,761
	TomekLinks	**0,826**	0,783	0,761
	ENN	**0,812**	0,782	0,763
	NCR	**0,831**	**0,793**	0,766
B-SMOTE SVM	-	**0,822**	0,770	0,759
	Random Undersampling	**0,829**	0,777	0,760
	TomekLinks	**0,824**	0,777	0,758
	ENN	**0,807**	0,774	0,758
	NCR	**0,825**	0,780	0,765
ADASYN	-	**0,822**	0,776	0,755
	Random Undersampling	**0,833**	**0,785**	0,759
	TomekLinks	**0,826**	0,784	0,759
	ENN	**0,821**	**0,789**	0,769
	OSS	**0,821**	0,782	0,764
	NCR	**0,836**	**0,794**	0,768

Таблица В.5 – Результаты экспериментов для набора данных «Показатели здоровья при диабете» (увеличение меньшего класса)

Метод увеличения меньшего класса	Метод уменьшения большего класса	MLP	Gaussian-NB	GB	Ada-Boost	XG-Boost	Light-GBM
-		0,385	0,453	0,387	0,391	0,387	0,385
Random Oversampling	-	**0,500**	**0,488**	**0,513**	**0,512**	**0,500**	**0,510**
	TomekLinks	**0,491**	**0,487**	**0,516**	**0,505**	**0,498**	**0,509**
	ENN	**0,495**	**0,489**	**0,514**	**0,509**	**0,499**	**0,513**
SMOTE	-	0,430	**0,489**	0,447	**0,465**	0,400	0,407
	TomekLinks	0,438	**0,490**	0,441	**0,462**	0,399	0,405
	ENN	**0,492**	**0,494**	**0,496**	**0,491**	**0,490**	**0,491**
B-SMOTE	-	0,452	**0,486**	0,452	0,463	0,399	0,402
	TomekLinks	0,454	**0,487**	0,448	0,459	0,400	0,403
B-SMOTE SVM	-	**0,469**	**0,487**	0,448	**0,467**	0,402	0,407
	TomekLinks	**0,474**	**0,486**	0,447	**0,470**	0,400	0,405
	OSS	**0,466**	**0,486**	0,441	**0,465**	0,395	0,399
ADASYN	-	**0,475**	**0,489**	0,442	0,460	0,398	0,400
	TomekLinks	0,444	**0,490**	0,436	0,460	0,397	0,398
	OSS	0,446	**0,491**	0,435	0,462	0,396	0,397

ПРИЛОЖЕНИЕ Г

РЕЗУЛЬТАТЫ ПОДБОРА СООТНОШЕНИЯ КЛАССОВ

Г1. Набор данных «Классификация здоровья плода»

Рисунок Г1.1 – Результаты для алгоритма «Random Oversampling»

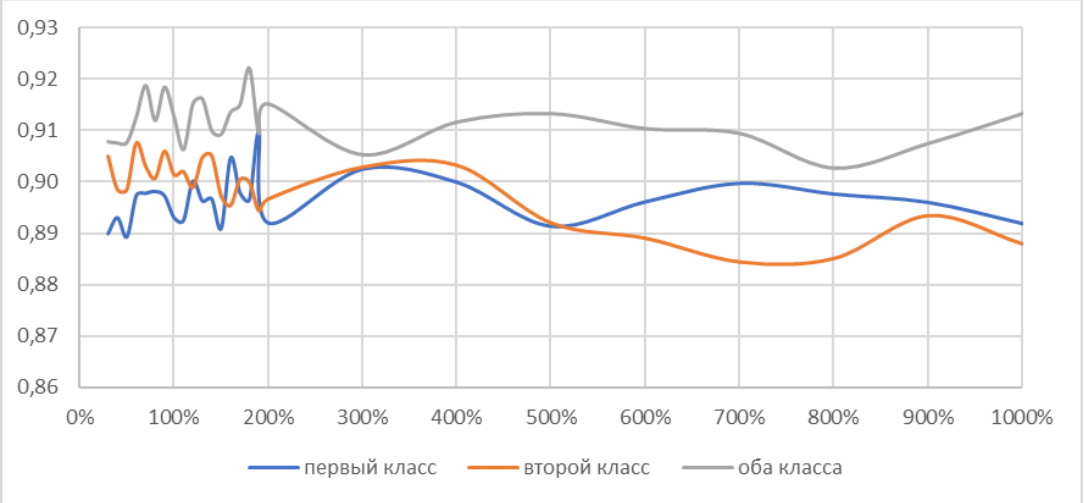


Рисунок Г1.2 – Результаты для алгоритма «SMOTE»

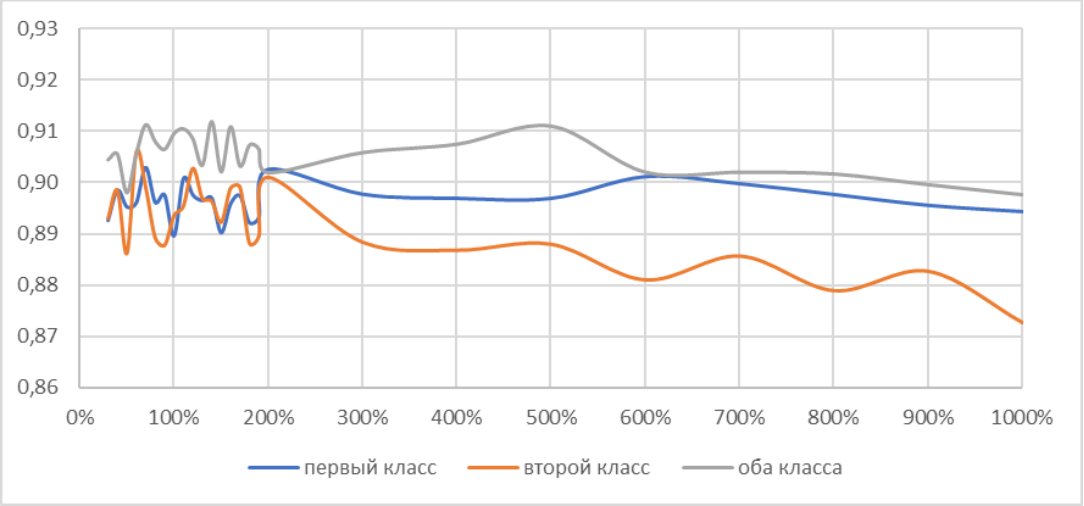


Рисунок Г1.3 – Результаты для алгоритма «B-SMOTE»

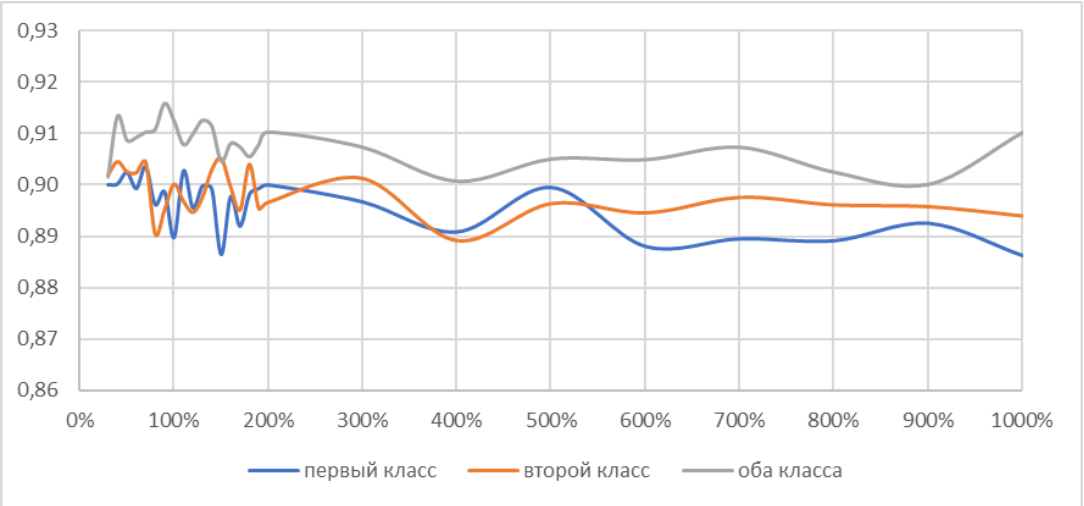


Рисунок Г1.4 – Результаты для алгоритма «B-SMOTE SVM»

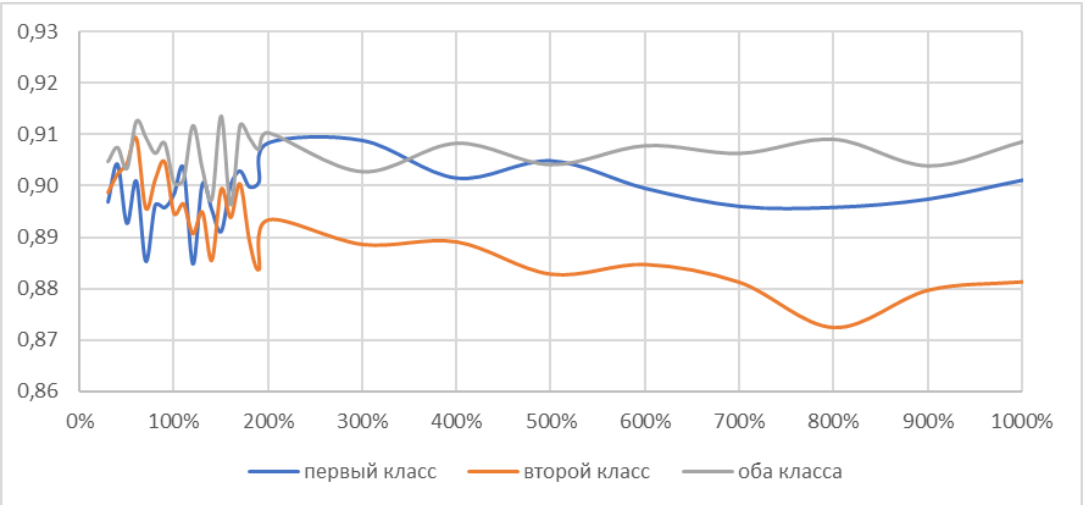


Рисунок Г1.5 – Результаты для алгоритма «ADASYN»

Г2. Набор данных «Прогнозирование церебрального инсульта»

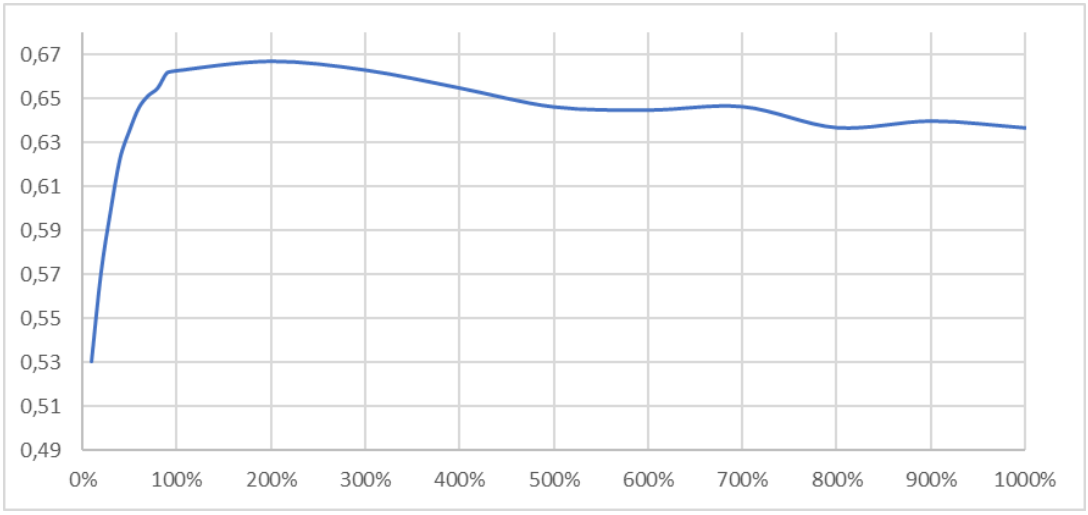


Рисунок Г2.1 – Результаты для алгоритма «Random Oversampling»

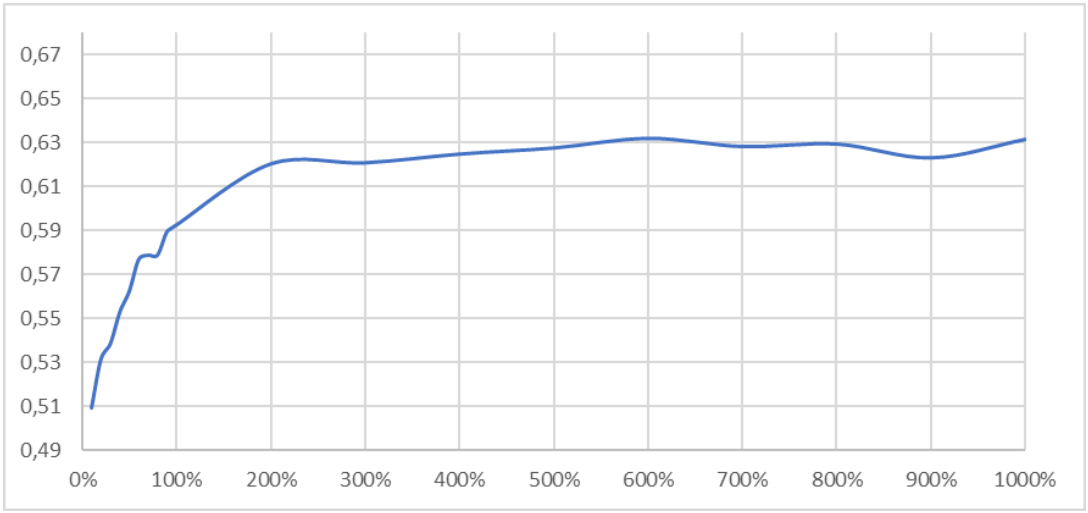


Рисунок Г2.2 – Результаты для алгоритма «SMOTE»

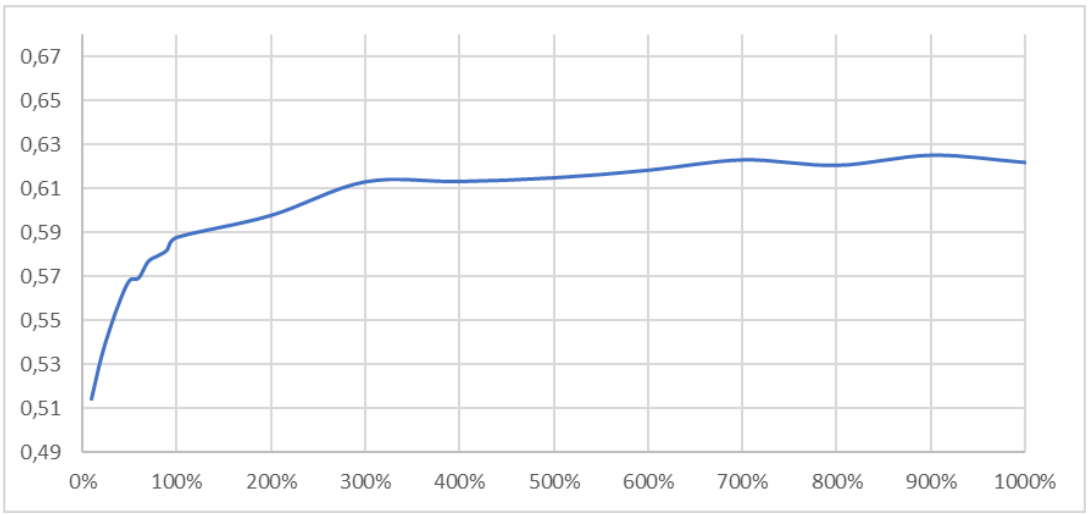


Рисунок Г2.3 – Результаты для алгоритма «B-SMOTE»

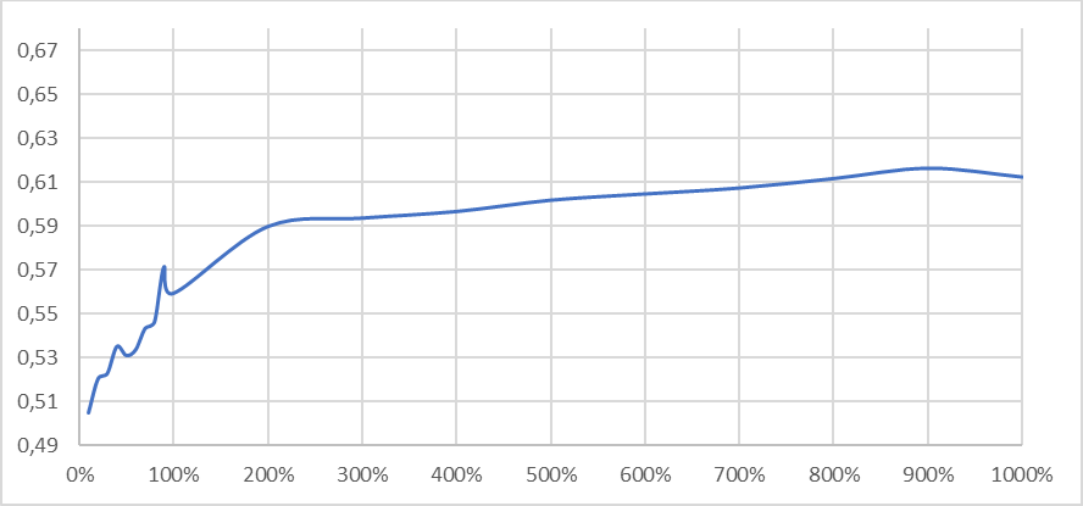


Рисунок Г2.4 – Результаты для алгоритма «B-SMOTE SVM»

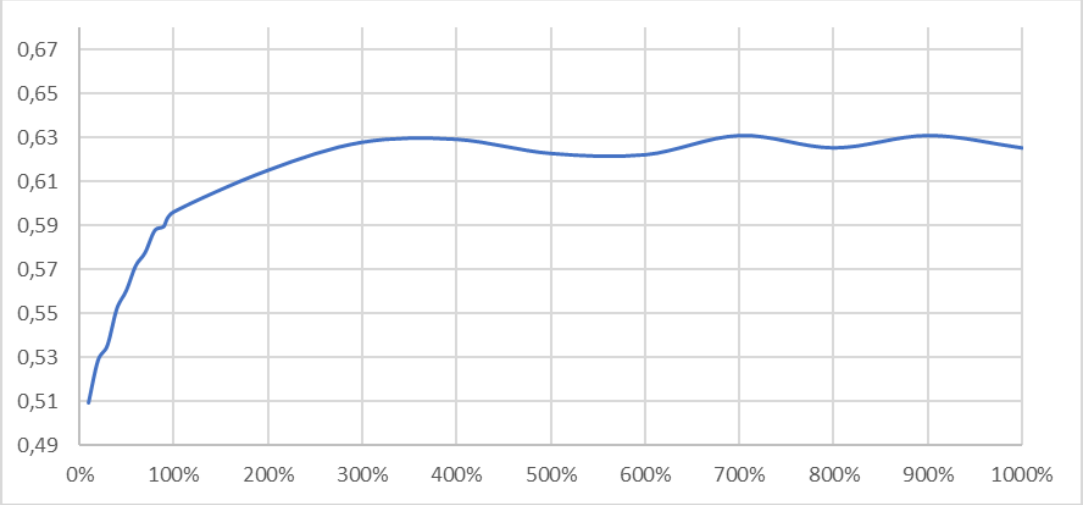


Рисунок Г2.5 – Результаты для алгоритма «ADASYN»

Г3. Набор данных «Кредитный риск»

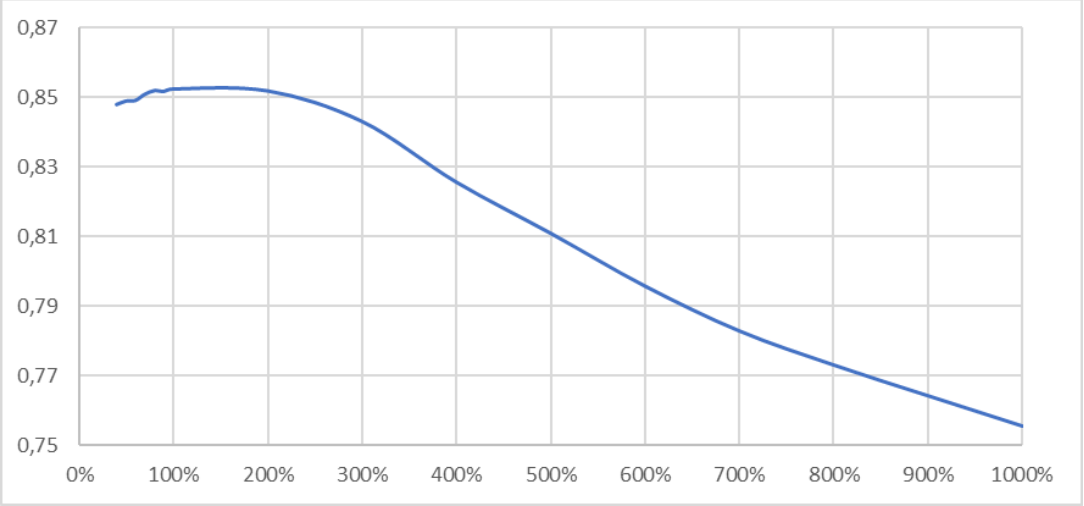


Рисунок Г3.1 – Результаты для алгоритма «Random Oversampling»

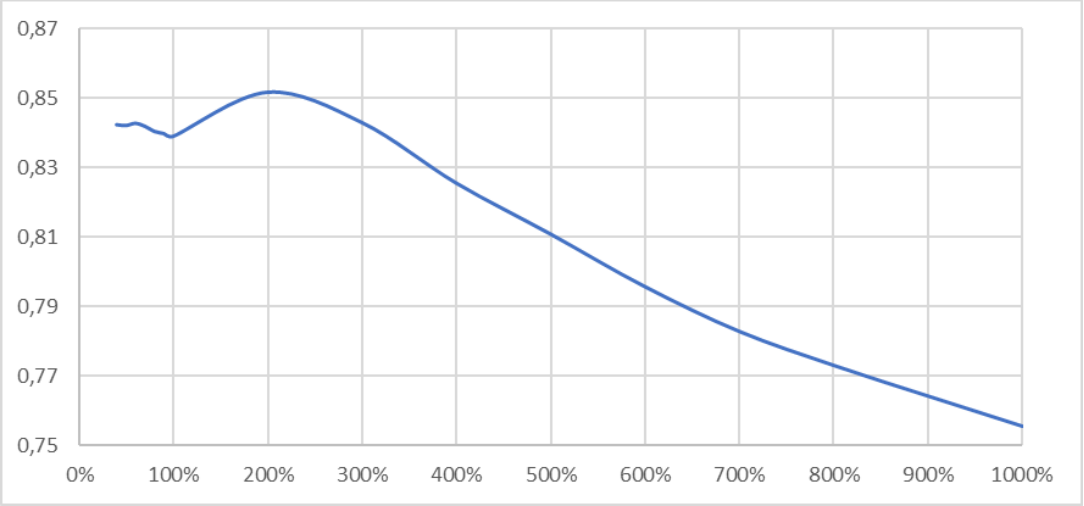


Рисунок Г3.2 – Результаты для алгоритма «SMOTE»

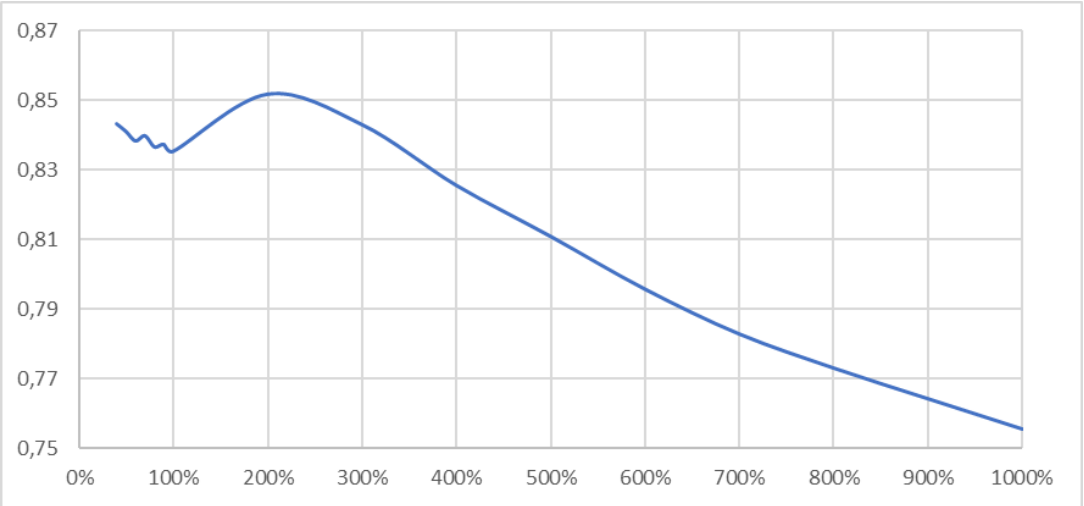


Рисунок Г3.3 – Результаты для алгоритма «B-SMOTE»

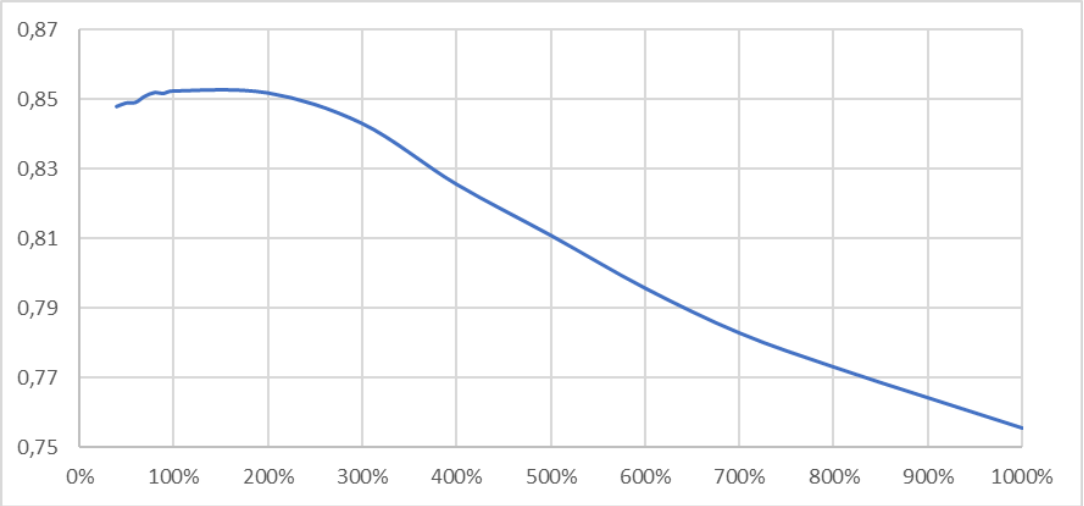


Рисунок Г3.4 – Результаты для алгоритма «B-SMOTE SVM»

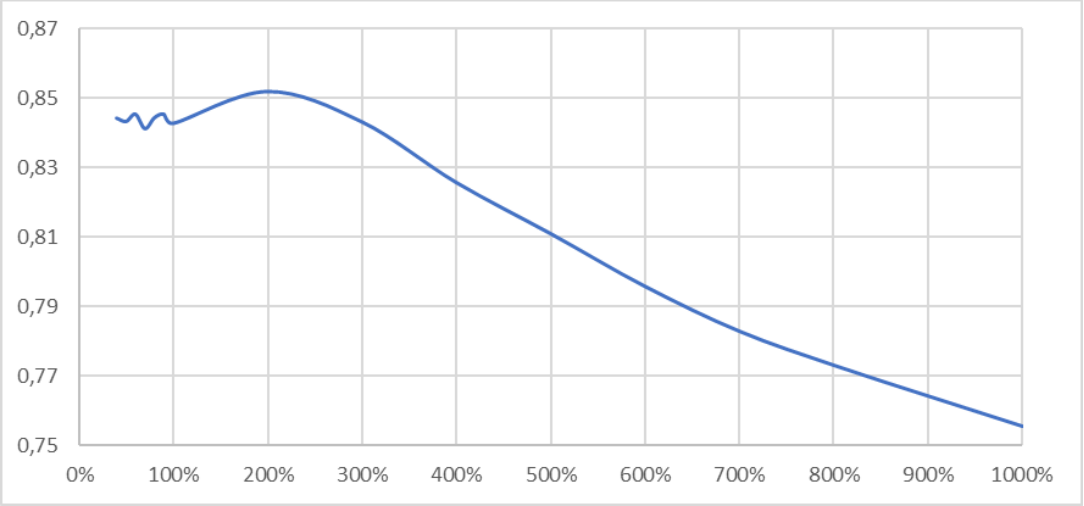


Рисунок Г3.5 – Результаты для алгоритма «ADASYN»

Г4. Набор данных «Классификация кредитного рейтинга»

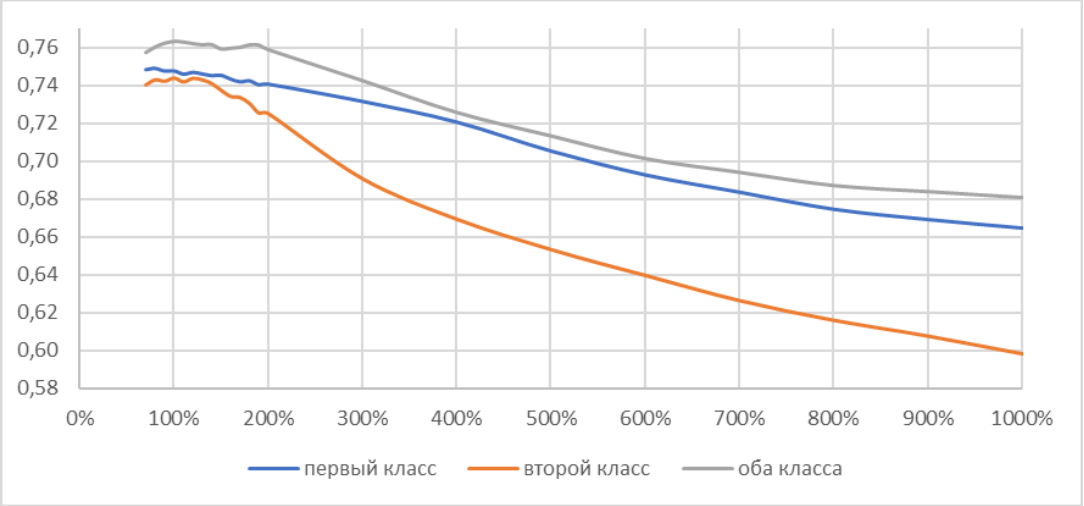


Рисунок Г4.1 – Результаты для алгоритма «Random Oversampling»

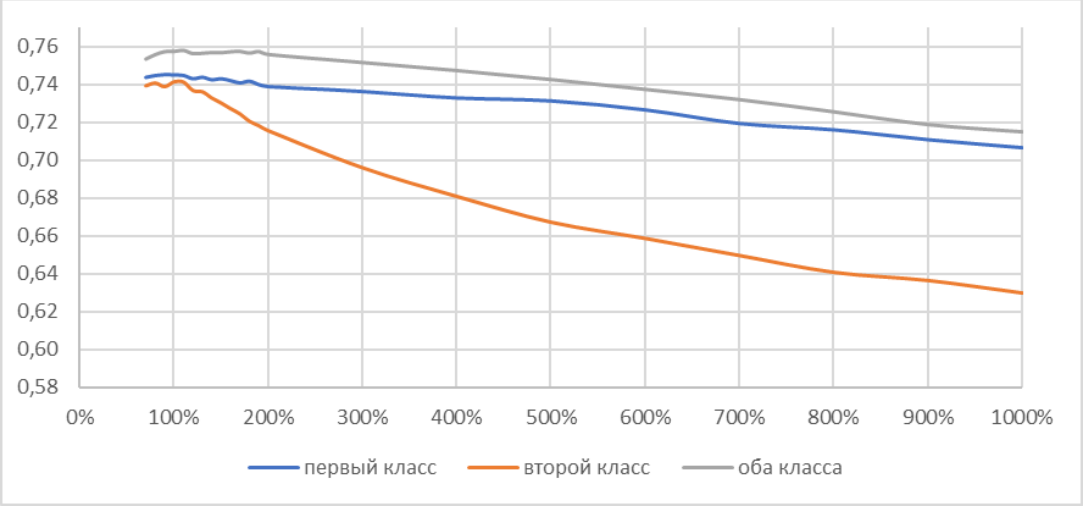


Рисунок Г4.2 – Результаты для алгоритма «SMOTE»

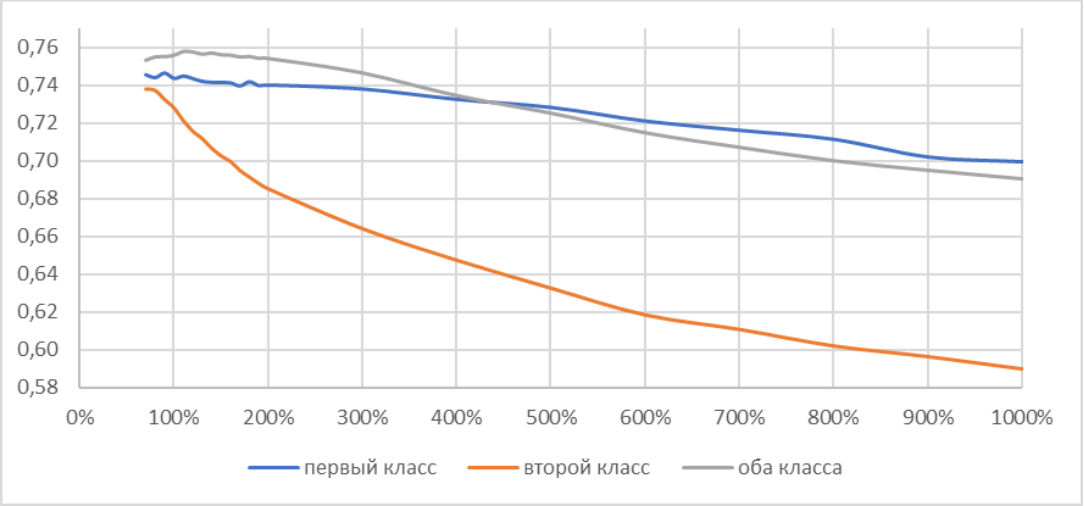


Рисунок Г4.3 – Результаты для алгоритма «B-SMOTE»

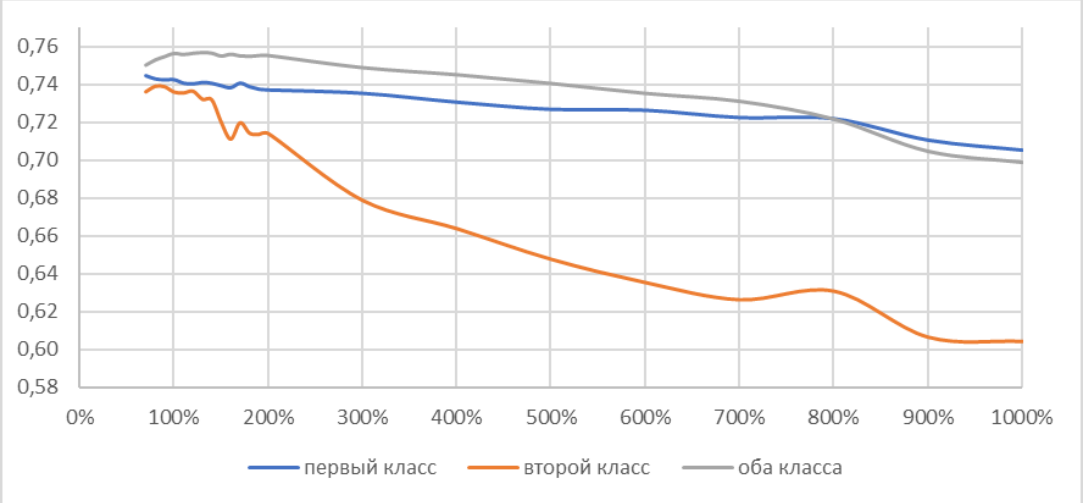


Рисунок Г4.4 – Результаты для алгоритма «B-SMOTE SVM»

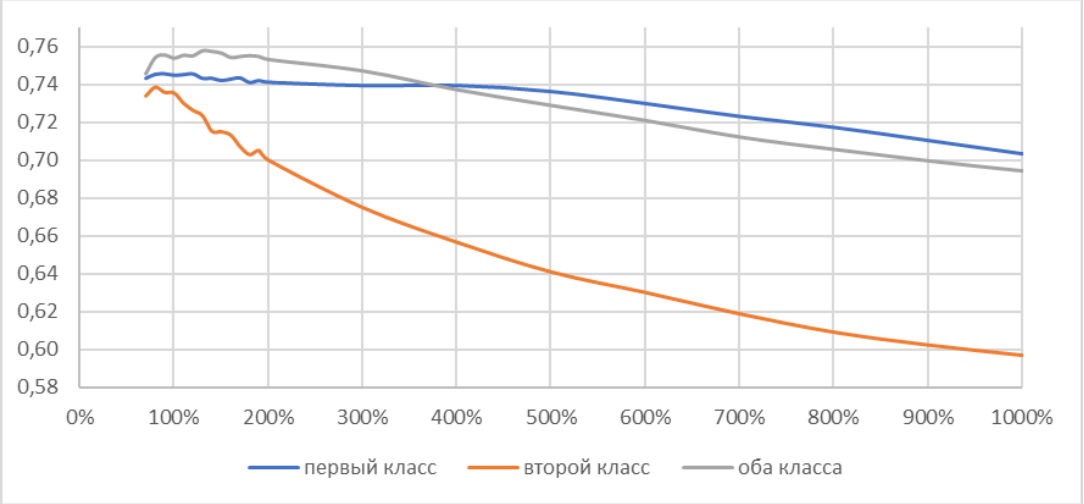


Рисунок Г4.5 – Результаты для алгоритма «ADASYN»

Г5. Набор данных «Показатели здоровья при диабете»

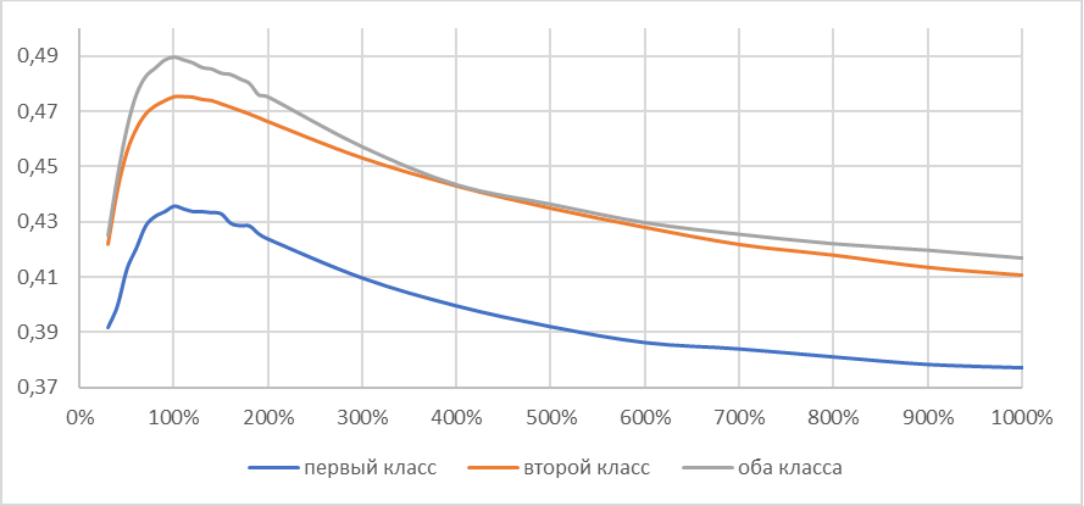


Рисунок Г5.1 – Результаты для алгоритма «Random Oversampling»

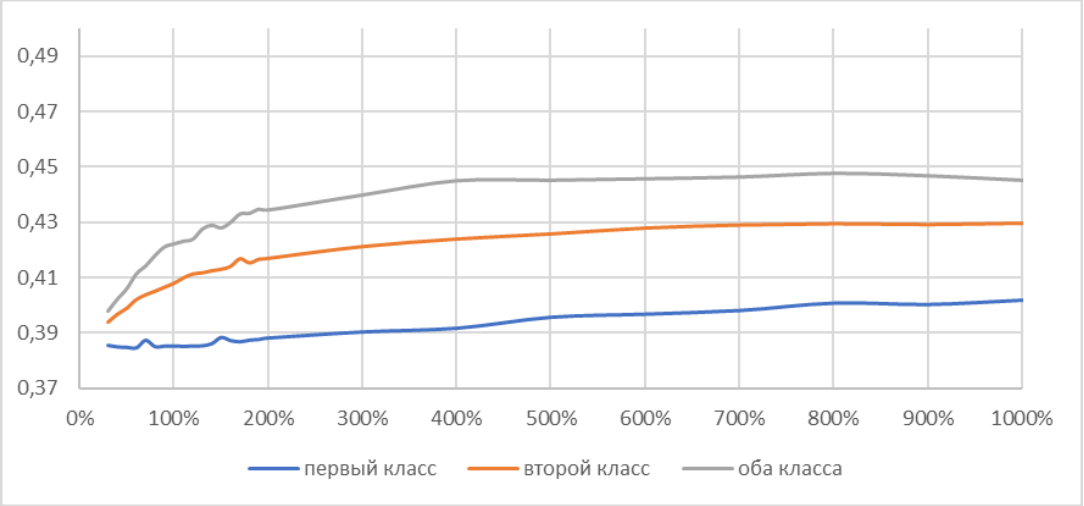


Рисунок Г5.2 – Результаты для алгоритма «SMOTE»

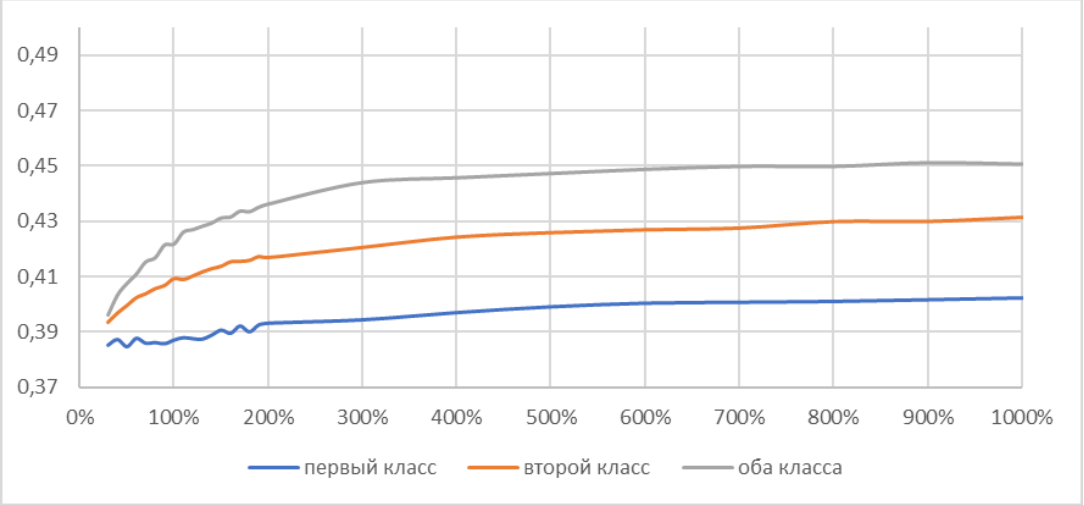


Рисунок Г5.3 – Результаты для алгоритма «B-SMOTE»

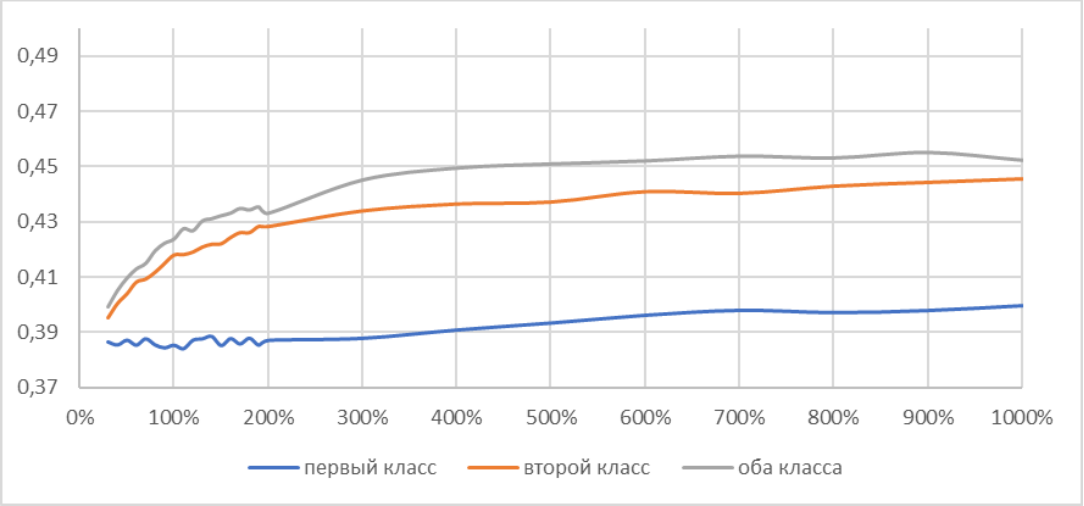


Рисунок Г5.4 – Результаты для алгоритма «B-SMOTE SVM»

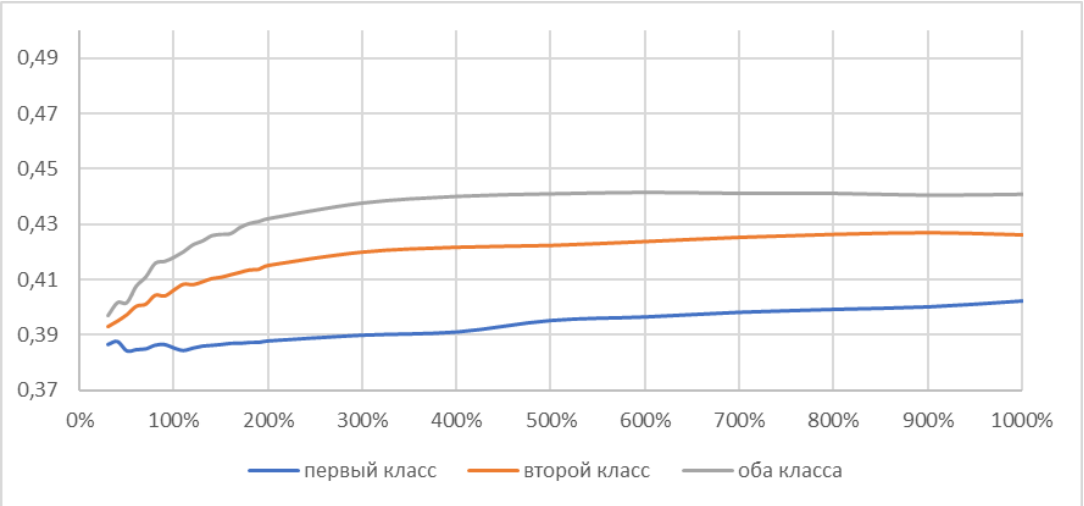


Рисунок Г5.5 – Результаты для алгоритма «ADASYN»

ПРИЛОЖЕНИЕ Д

РЕЗУЛЬТАТЫ ПОДБОРА ГИПЕРПАРАМЕТРОВ

Полужирным шрифтом обозначены максимальные значения точности для набора в целом. Более бледный цвет шрифта использовался в тех случаях, когда наблюдалось отсутствие повышения точности классификации по сравнению с точностью для исходного набора данных. Сокращениями RF и GB обозначаются случайный лес и градиентный бустинг соответственно.

Таблица Д.1 – Результаты экспериментов для набора данных «Классификация здоровья плода»

Метод увеличения меньшего класса	Метод уменьшения большего класса	RF	GB	XGBoost	LightGBM
-		0,921	0,919	0,921	0,928
Random Oversampling	-	0,942	0,945	**0,948**	0,940
	Random Undersampling	0,938	0,944	**0,948**	**0,946**
	NearMiss-1	**0,946**	**0,948**	**0,951**	0,941
	TomekLinks	0,943	**0,946**	**0,947**	0,942
	ENN	**0,947**	**0,952**	**0,949**	**0,950**
	OSS	0,939	0,940	**0,946**	**0,946**
SMOTE	-	0,937	0,936	0,943	**0,945**
	Random Undersampling	0,939	0,945	**0,949**	**0,951**
	NearMiss-1	0,943	**0,949**	**0,948**	0,944
	TomekLinks	0,942	0,942	**0,954**	**0,952**
	ENN	0,944	**0,952**	**0,950**	**0,951**
	OSS	0,936	0,936	**0,945**	0,944
B-SMOTE	-	0,937	0,935	0,931	0,945
	Random Undersampling	0,941	0,944	**0,949**	**0,952**
	NearMiss-1	0,944	0,931	**0,948**	0,938
	TomekLinks	0,942	0,933	0,944	0,943
	ENN	0,937	0,940	**0,953**	0,944
	OSS	0,938	0,931	0,929	0,938
B-SMOTE SVM	-	0,931	0,935	0,934	0,941
	Random Undersampling	0,936	**0,944**	**0,947**	**0,949**
	NearMiss-1	0,941	0,939	**0,946**	**0,946**
	TomekLinks	0,932	**0,950**	**0,946**	0,939
	ENN	0,940	0,942	**0,946**	**0,949**
	OSS	0,930	0,769	0,934	0,943
ADASYN	-	0,933	0,931	0,942	0,932
	Random Undersampling	0,935	0,941	0,942	**0,952**
	NearMiss-1	0,942	0,933	**0,947**	0,938
	TomekLinks	0,936	0,936	0,940	0,943
	ENN	0,936	0,935	0,940	**0,944**
	OSS	0,931	0,934	**0,945**	0,942

Таблица Д.2 – Результаты экспериментов для набора данных «Прогнозирование церебрального инсульта»

Метод увеличения меньшего класса	Метод уменьшения большего класса	GaussianNB	GB	AdaBoost
-		-	0,736	0,514
Random Oversampling	-	0,753	**0,782**	**0,780**
	Random Undersampling	0,753	**0,782**	**0,779**
	ENN	0,759	**0,772**	**0,782**
SMOTE	-	0,728	0,765	0,766
	Random Undersampling	0,736	0,762	0,765
	ENN	0,747	0,762	0,762
	OSS	0,727	0,762	0,765
B-SMOTE	-	0,755	**0,773**	0,766
	Random Undersampling	0,754	**0,778**	0,766
	ENN	0,767	**0,777**	0,768
	OSS	0,754	**0,777**	0,749
B-SMOTE SVM	-	0,732	0,765	0,749
	Random Undersampling	0,755	**0,774**	0,766
	ENN	0,761	**0,774**	0,754
	OSS	0,736	**0,777**	0,747
ADASYN	-	0,725	0,762	0,762
	Random Undersampling	0,733	0,762	0,762
	ENN	0,748	0,766	0,762
	OSS	0,724	0,758	0,762

Таблица Д.3 – Результаты экспериментов для набора данных «Кредитный риск»

Метод увеличения меньшего класса	Метод уменьшения большего класса	RF	GB	AdaBoost	XG Boost	LightGBM
-		0,856	0,864	0,812	0,869	0,870
Random Oversampling	-	0,860	**0,881**	0,843	**0,877**	**0,876**
	Random Undersampling	0,863	**0,882**	0,845	**0,878**	**0,877**
	TomekLinks	0,861	**0,879**	0,848	**0,875**	**0,875**
	ENN	0,861	0,872	0,845	**0,877**	**0,878**
	OSS	0,860	**0,881**	0,847	**0,877**	**0,875**
	NCR	0,858	**0,877**	0,844	**0,877**	**0,877**
SMOTE	-	0,858	0,869	0,822	0,874	0,873
	Random Undersampling	0,853	**0,875**	0,840	**0,875**	**0,874**
	TomekLinks	0,858	0,873	0,828	**0,874**	**0,874**
	ENN	0,854	**0,875**	0,837	**0,875**	**0,877**
	OSS	0,858	0,873	0,827	**0,874**	**0,876**
	NCR	0,853	0,873	0,836	0,873	0,872
B-SMOTE	-	0,856	0,869	0,820	0,872	0,872
	Random Undersampling	0,851	**0,874**	0,836	**0,876**	**0,875**
	TomekLinks	0,858	**0,876**	0,823	**0,874**	**0,875**
	ENN	0,856	**0,874**	0,835	**0,875**	**0,877**
	OSS	0,857	0,871	0,825	**0,873**	**0,876**
	NCR	0,850	0,870	0,838	**0,873**	0,872
B-SMOTE SVM	-	0,857	0,871	0,829	**0,875**	**0,874**
	Random Undersampling	0,859	**0,875**	0,843	**0,877**	**0,876**
	TomekLinks	0,859	**0,874**	0,829	**0,875**	**0,875**

Продолжение таблицы Д.3

Метод увеличения меньшего класса	Метод уменьшения большего класса	RF	GB	AdaBoost	XG Boost	LightGBM
	ENN	0,857	**0,877**	0,836	**0,877**	**0,877**
	OSS	0,858	**0,875**	0,828	**0,875**	**0,876**
	NCR	0,854	**0,874**	0,838	**0,874**	**0,875**
ADASYN	-	0,856	0,870	0,820	**0,873**	**0,873**
	Random Undersampling	0,852	**0,875**	0,838	**0,875**	**0,874**
	TomekLinks	0,858	0,871	0,824	**0,875**	**0,875**
	ENN	0,858	**0,877**	0,834	**0,877**	**0,877**
	OSS	0,858	0,873	0,823	**0,876**	**0,875**
	NCR	0,854	0,872	0,838	**0,875**	**0,875**

Таблица Д.4 – Результаты экспериментов для набора данных «Классификация кредитного рейтинга»

Метод увеличения меньшего класса	Метод уменьшения большего класса	RF	XGBoost	LightGBM
-		0,802	0,796	0,797
Random Oversampling	-	**0,827**	0,822	0,823
	Random Undersampling	**0,835**	0,823	0,824
	TomekLinks	**0,828**	0,824	0,825
	ENN	**0,835**	**0,833**	0,826
	NCR	**0,836**	**0,835**	**0,834**
SMOTE	-	0,822	0,813	0,814
	Random Undersampling	**0,831**	0,819	0,822
	TomekLinks	0,826	0,819	0,817
	ENN	**0,829**	**0,833**	0,826
	NCR	**0,836**	**0,835**	**0,834**
B-SMOTE	-	0,826	0,818	0,819
	Random Undersampling	**0,832**	0,824	0,824
	TomekLinks	0,826	0,823	0,822
	ENN	**0,834**	**0,835**	**0,831**
	NCR	**0,836**	**0,836**	**0,833**
ADASYN	-	0,826	0,820	0,819
	Random Undersampling	**0,832**	0,824	0,823
	TomekLinks	**0,828**	0,823	0,824
	ENN	**0,840**	**0,839**	**0,837**
	NCR	**0,836**	**0,839**	**0,837**

Таблица Д.5 – Результаты экспериментов для набора данных «Показатели здоровья при диабете» (увеличение меньшего класса)

Метод увеличения меньшего класса	Метод уменьшения большего класса	Gaussian NB	GB	Ada Boost	XG Boost	Light GBM
-		-	0,460	0,453	0,394	0,441
Random Oversampling	-	0,490	**0,515**	**0,515**	**0,509**	**0,506**
	TomekLinks	0,490	0,488	**0,516**	**0,510**	**0,507**
	ENN	0,492	**0,516**	**0,517**	**0,510**	**0,511**

Продолжение таблицы Д.5

Метод увеличения меньшего класса	Метод уменьшения большего класса	Gaussian NB	GB	Ada Boost	XG Boost	Light GBM
SMOTE	-	0,492	0,466	0,470	**0,506**	0,465
	TomekLinks	0,492	0,455	0,489	**0,508**	0,463
	ENN	**0,496**	**0,505**	**0,509**	**0,509**	**0,502**
B-SMOTE	-	**0,496**	0,467	**0,501**	**0,505**	0,486
	TomekLinks	**0,495**	0,473	0,481	**0,503**	0,488
ADASYN	-	0,493	0,464	0,480	**0,505**	0,464
	TomekLinks	0,493	0,462	0,479	**0,505**	0,461
	OSS	0,493	0,459	0,480	**0,505**	0,461