УДК 004.9

МЕТОДЫ РЕШЕНИЯ ПРОБЛЕМЫ ДИСБАЛАНСА КЛАССОВ В ЗАДАЧЕ БИНАРНОЙ КЛАССИФИКАЦИИ

Демидова Л.А., Шаршатов М.А., Шыхыев А.А.

МИРЭА - Российский технологический университет», 119454, Россия, г. Москва, проспект Вернадского, 78, e-mail: sharshatov99@mail.ru

В статье исследуется проблема дисбаланса классов в задаче бинарной классификации и применение различных методов для решения этой проблемы. Были проведены эксперименты на несбалансированных данных, а также сравнение результатов применения трех методов: увеличение выборки, уменьшение выборки и взвешивание классов. Для оценки качества моделей использовались метрики Accuracy, Precision, Recall, F1-score, ROC AUC и PR AUC. Проведено сравнение метрик ROC AUC и PR AUC в случае с несбалансированными данными.

Ключевые слова: машинное обучение, классификация данных, дисбаланс классов.

METHODS FOR SOLVING THE CLASS IMBALANCE PROBLEM IN BINARY CLASSIFICATION TASK

Demidova L.A., Sharshatov M.A., Shykhyev A.A.

MIREA - Russian Technological University", 119454, Moscow, 78 Vernadskogo Avenue, Russia, e-mail: sharshatov99@mail.ru

The article investigates the problem of class imbalance in binary classification tasks and the application of various methods to solve this problem. Experiments were conducted on imbalanced data, and the results of three methods were compared: oversampling, undersampling, and class weighting. Metrics such as Accuracy, Precision, Recall, F1-score, ROC AUC, and PR AUC were used to evaluate the models' performance. A comparison of ROC AUC and PR AUC metrics was conducted for imbalanced data.

Keywords: machine learning, data classification, class imbalance.

Введение

Классификация данных является одной из наиболее распространенных задач в машинном обучении. Она заключается в определении класса объекта на основе набора признаков. Однако, в реальных задачах часто возникает проблема дисбаланса классов, когда размеры классов существенно отличаются. Например, в задаче детектирования мошеннических операций в банковских данных [1], количество мошеннических операций может быть значительно меньше, чем количество обычных операций.

Дисбаланс классов приводит к снижению качества модели, так как происходит смещение в сторону предсказания большего класса и не учитывает меньший класс. В

результате модель может неверно классифицировать объекты меньшего класса. Решение проблемы дисбаланса классов является актуальной областью исследований в машинном обучении. Существуют различные методы балансировки классов, которые позволяют бороться с этой проблемой. В данной статье рассматриваются различные методы балансировки классов и их применение в задачах классификации данных.

Понятие дисбаланса классов

Дисбаланс классов — это проблема, которая возникает при обучении модели классификации на несбалансированных данных, когда один класс представлен значительно меньшим количеством примеров, чем другой класс. Например, в задаче определения мошеннических транзакций в банковском секторе [1], количество нормальных транзакций будет гораздо больше, чем количество мошеннических.

Дисбаланс классов может привести к смещению модели в пользу большего класса, что приведет к низкой точности модели на меньшем классе. Проблема дисбаланса классов возникает по разным причинам, например, в некоторых задачах один класс может быть более редким, чем другой класс. Кроме того, дисбаланс классов может быть вызван ошибками при сборе данных, когда один класс может быть недостаточно представлен из-за ограничений доступа или ошибок сбора. Наконец, дисбаланс классов может быть связан с задачей самой по себе, когда один класс является более важным для задачи, чем другой класс.

Понимание причин дисбаланса классов помогает выбрать подходящие методы для решения проблемы и повышения точности модели.

Дисбаланс классов может приводить к серьезным последствиям для алгоритмов машинного обучения, так как они будут склонны присваивать большую долю объектов классу с большим количеством примеров, игнорируя или плохо распознавая объекты меньшего класса. Это может привести к следующим проблемам.

1. Переобучение: алгоритм машинного обучения может переобучиться на более представленный класс, избегая корректное распознавание менее представленного класса. Это может привести к плохим результатам классификации для меньшего класса.

2. Низкая точность: из-за неравномерного распределения классов, алгоритм может быть более склонен к определенному классу, давая более высокую точность для этого класса, но плохие результаты для остальных классов.

3. Низкая полнота: алгоритм может неправильно классифицировать объекты меньшего класса, что может привести к низкой полноте для этого класса. Это может быть опасно, если важные объекты меньшего класса не будут обнаружены.

4. Невозможность обучения: для некоторых алгоритмов машинного обучения дисбаланс классов может стать причиной невозможности обучения. Если объекты меньшего класса представлены недостаточно, то алгоритм не сможет обучить правильную модель для распознавания этого класса.

5. Неадекватная оценка: использование метрик оценки моделей машинного обучения, которые не учитывают дисбаланс классов, может привести к неадекватной оценке модели. Например, доля правильных ответов может быть высокой для более представленного класса, но быть низкой для менее представленного.

6. Низкая устойчивость: если объекты разных классов распределены неравномерно, то алгоритмы машинного обучения могут быть неустойчивы к изменениям в распределении данных.

Поэтому решение проблемы дисбаланса классов является важной задачей в машинном обучении.

Алгоритмы классификации данных

Алгоритмы классификации данных являются одним из основных инструментов машинного обучения. Они используются для определения принадлежности объектов к определенным классам на основе определенных признаков. В этом разделе будут рассмотрены три распространенных алгоритма классификации: SVM, Random Forest и k-NN.

SVM (Support Vector Machine) [2] — это алгоритм классификации, который строит гиперплоскость в многомерном пространстве, которая разделяет классы данных на две части. SVM пытается найти гиперплоскость с наибольшим зазором между классами, что делает его особенно хорошо подходящим для решения задач с линейно разделимыми классами.

Random Forest — это ансамбль решающих деревьев, которые строятся на основе подвыборок обучающих данных. Каждое дерево в случайном лесу строится независимо, а затем результаты их работы объединяются, чтобы определить окончательный результат. Этот метод обладает хорошей способностью к обобщению, способностью обрабатывать большие объемы данных и простотой в использовании.

k-NN (k-ближайших соседей) [3] — это алгоритм классификации, основанный на определении класса объекта на основе его ближайших соседей в пространстве признаков. k-NN выбирает k ближайших объектов из обучающей выборки и относит исследуемый объект к тому классу, которому принадлежит большинство из этих k ближайших объектов. Этот метод является простым и хорошо работает с небольшими наборами данных, но имеет слабые стороны, когда речь идет о больших объемах данных и сложных пространствах признаков.

Методы решения проблемы дисбаланса классов

В последние несколько лет были предложены различные методы для решения проблемы дисбаланса классов. Два основных подхода, которые используются для обучения несбалансированных данных, — это методы уровня алгоритма и методы уровня данных.

Методы уровня данных являются одним из подходов к решению проблемы дисбаланса классов в классификации данных. Эти методы направлены на изменение баланса классов в обучающей выборке путем изменения количества примеров в каждом классе.

Существует несколько подходов к решению проблемы дисбаланса классов на уровне данных [4].

1. Увеличение выборки — метод балансировки классов путем увеличения количества образцов в меньшем классе. Это может быть достигнуто путем случайного дублирования образцов из меньшего класса или генерации новых образцов на основе имеющихся.

Одним из наиболее распространенных методов увеличения выборки является метод Synthetic Minority Oversampling Technique (SMOTE) [5], который генерирует новые образцы, основываясь на близости между существующими образцами в меньшем классе. SMOTE создает новые образцы путем комбинации ближайших соседей, а затем добавляет эти образцы в выборку.

Хотя генерация новых данных может быть эффективным способом балансировки

дисбаланса классов, это может привести к переобучению и увеличению шума в данных, особенно если используется случайный способ генерации новых образцов. Поэтому важно оценить качество модели и проводить кросс-валидацию при использовании этого метода.

2. Уменьшение выборки — метод балансировки классов путем уменьшения количества образцов в преобладающем классе до уровня менее представленного класса. Это позволяет уравнять количество образцов между классами и снизить дисбаланс.

Существует несколько способов реализации этого метода, включая случайный выбор набора образцов из преобладающего класса и удаление наиболее похожих образцов до достижения баланса. Однако это может привести к потере важной информации, особенно если преобладающий класс содержит уникальные или важные образцы. Поэтому перед применением этого метода необходимо проанализировать данные и оценить, какой уровень уменьшения количества образцов в преобладающем классе может быть оптимальным для сохранения репрезентативности данных и улучшения баланса между классами.

Примерами методов уменьшения выборки являются Random Under Sampling и Tomek Links [6].

3. Метод взвешивания классов — метод, который заключается в присвоении большего веса меньшему классу, чтобы сбалансировать значимость классов при обучении модели. Этот метод может использоваться в различных алгоритмах машинного обучения, включая логистическую регрессию, метод опорных векторов (SVM) и случайный лес.

Веса классов можно настроить вручную или автоматически. Вручную заданные веса могут быть основаны на экспертном знании области применения, однако это может потребовать значительных усилий и времени. Автоматическое настройка весов может быть выполнена с использованием различных методов, включая методы, основанные на расстоянии между классами, методы, основанные на анализе частоты классов, и методы, основанные на оптимизации функции стоимости.

Использование взвешивания классов может улучшить производительность модели при классификации данных с дисбалансом классов, так как он позволяет обучать модель с учетом значимости каждого класса. Однако этот метод также может привести к переобучению модели, если веса классов заданы неправильно. Поэтому настройка весов классов является важной частью применения этого метода.

Методы алгоритмического уровня [4] решения проблемы дисбаланса классов направлены на изменение алгоритмов машинного обучения для более эффективного учета дисбаланса классов. Рассмотрим несколько таких методов.

1. Модификация функции потерь. Функция потерь отвечает за оценку ошибки алгоритма и определяет, как модель должна обновлять свои параметры. При дисбалансе классов модификация функции потерь может повысить важность класса меньшинства. Например, можно добавить штраф за ошибки в классификации класса меньшинства, чтобы минимизировать ошибки в этом классе.

2. Использование алгоритмов с учетом весов классов. Некоторые алгоритмы машинного обучения позволяют задавать веса классам в соответствии с их долей в выборке. Например, в методе опорных векторов (SVM) можно задать разные штрафы за ошибки в каждом классе.

3. Использование ансамблевых методов. Ансамблевые методы могут эффективно

учитывать дисбаланс классов, объединяя несколько моделей в одну. Например, метод случайного леса (Random Forest) может обучаться на сбалансированных подвыборках каждого класса и комбинировать результаты.

4. Использование пороговой вероятности. Многие алгоритмы машинного обучения возвращают вероятности принадлежности к классам, а не просто метки классов. Можно задать пороговую вероятность для определения, какой класс выбрать. При дисбалансе классов порог можно изменить так, чтобы повысить точность в классе меньшинства.

Эксперименты и результаты

В основу исследования был взят набор данных [7], который показывает, была ли одобрена заявка на получение кредита перечисленными людьми. Данные содержат различные экономические и демографические признаки.

Данные являются несбалансированными и состоят из 4521 объектов, из которых только 521 относятся к положительному классу.

На рисунке 1 с помощью T-SNE [8] показано распределение между заявками, которые были одобрены (класс 1) и теми, на которые получен отказ (класс 0).

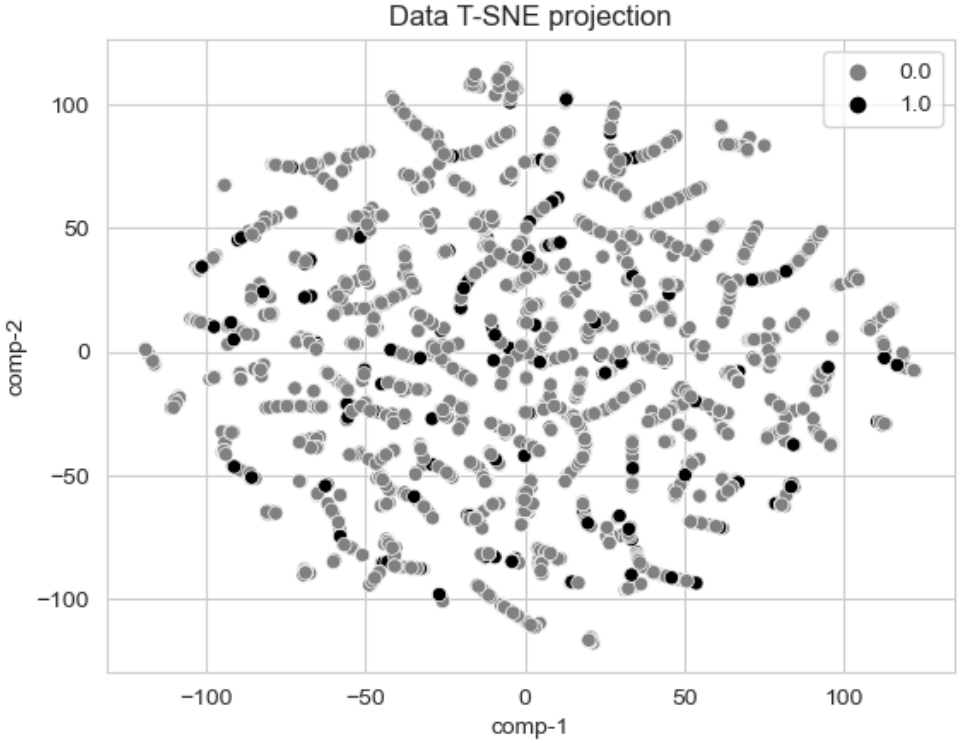

Рис. 1 - Распределение данных на плоскости

В качестве методов решения проблемы дисбаланса были использованы методы уровня данных для дальнейшего сравнения. Алгоритмом классификации выступил метод опорных векторов (SVM).

Выборка была разбита на обучающую и тестовую в соотношении 70/30, перемешана с seed = 128 для инициализации генератора случайных чисел.

При оценке качества классификатора использовались метрики [9], приведенные ниже.

Accuracy (точность) — это метрика, которая показывает, насколько часто классификатор правильно классифицирует объекты. Она вычисляется как отношение числа правильно классифицированных объектов к общему числу объектов в выборке.

Precision (точность по классу) — это метрика, которая показывает, насколько точно классификатор классифицирует объекты определенного класса. Она вычисляется как отношение числа правильно классифицированных объектов определенного класса к общему числу объектов, которые классификатор отнес к этому классу.

Recall (полнота) — это метрика, которая показывает, насколько полно классификатор находит объекты определенного класса. Она вычисляется как отношение числа правильно классифицированных объектов определенного класса к общему числу объектов этого класса в выборке.

F1-мера — это гармоническое среднее между Precision и Recall. Она вычисляется как отношение произведения precision и recall к их сумме, умноженной на два.

В таблице 1 приведены значения метрик для несбалансированных данных.

Таблица 1. Значения метрик для несбалансированных данных

Accuracy	Recall	Precision	F_1-score
0.88	0.88	0.78	0.83

Матрица ошибок (confusion matrix) представляет собой таблицу, которая показывает, сколько объектов каждого класса было правильно или неправильно классифицировано алгоритмом. Матрица может быть полезна для понимания, какие классы чаще всего определяются с неправильной меткой. Это может помочь улучшить алгоритм, например, путем добавления большего количества образцов мало представленных классов в обучающую выборку.

Несмотря на высокие показатели метрик качества, матрица ошибок показывает, что классификатор имеет проблемы в определении класса меньшинства, так модель правильно не определила ни одного образца класса с меткой 1.

Таблица 2. Матрица ошибок для несбалансированных данных

Истинный\Предсказанный	0	1
0	1200	0
1	156	0

В качестве метода увеличения выборки был использован SMOTE. Он применяется следующим образом: для каждого образца минорного класса выбирается k ближайших соседей из того же класса, затем для каждого из этих соседей случайным образом выбирается один из них, и новый синтетический образец создается путем интерполяции между исходным образцом и выбранным соседом. Этот процесс повторяется для каждого образца минорного класса до тех пор, пока не будет достигнуто желаемое соотношение между классами.

В таблице 3 представлены значения метрик качества при увеличении выборки.

Таблица 3. Метрики качества при увеличении выборки

Accuracy	Recall	Precision	F_1-score
0.63	0.63	0.63	0.62

Из матрицы ошибок (таблица 4) можно также заметить, что класс 0 имеет большее количество ложно-отрицательных примеров, то есть объектов, которые действительно принадлежат к классу 0, но были ошибочно отнесены к классу 1 (557 примеров), чем ложно-положительных примеров, то есть объектов, которые не принадлежат к классу 0, но были ошибочно отнесены к этому классу (341 пример).

В таблице 4 приведена матрица ошибок при увеличении выборки.

Таблица 4. Матрица ошибок при увеличении выборки

Истинный\Предсказанный	0	1
0	643	557
1	341	856

В качестве метода уменьшения выборки использовался RandomUnderSampler из пакета imblearn.under_sampling.

RandomUnderSampler — это метод уменьшения выборки, который удаляет случайным образом примеры из меньшего класса, чтобы сбалансировать классы.

Этот метод работает путем выбора случайного набора примеров из меньшего класса и удаления их из выборки. Таким образом, количество примеров в каждом классе становится примерно одинаковым, что позволяет устранить дисбаланс классов.

Таким образом, после уменьшения выборки осталось 520 объектов каждого класса.

В таблице 5 представлены метрики при уменьшении выборки.

Таблица 5. Значения метрик качества при уменьшении выборки

Accuracy		Recall	Precision	F_1-score
0.57		0.57	0.57	0.57

Из матрицы ошибок (таблица 6) видно, что класс 0 был предсказан верно в 91 случае, а в 65 случаях был ошибочно предсказан как класс 1. Класс 1 был предсказан верно в 86 случаях, но в 70 случаях был ошибочно предсказан как класс 0.

Из результатов можно сделать вывод, что метод RandomUnderSampler не дал значительного улучшения качества классификации по сравнению с исходным набором данных. Классификатор всё ещё имеет трудности в определении объектов класса 1.

В таблице 6 приведена матрица ошибок при уменьшении выборки.

Таблица 6. Матрица ошибок при уменьшении выборки

Истинный\Предсказанный	0	1
0	91	65
1	70	86

Для определения весов классов был использован метод compute_class_weight из библиотеки sklearn. Этот метод автоматически вычисляет веса классов на основе дисбаланса в обучающей выборке. Вычисленные веса классов затем передаются в параметр class_weight при обучении SVM. Параметр class_weight позволяет задавать веса классам в соответствии с дисбалансом. Таким образом, модель учитывает разницу в

количестве примеров между классами при обучении и старается минимизировать ошибки на меньшем классе.

В таблице 7 представлены значения метрик при взвешивании классов.

Таблица 7. Значения метрик качества при взвешивании классов

Accuracy	Recall	Precision	F_1-score
0.59	0.59	0.82	0.66

Для несбалансированных данных было предсказано 0 в 1200 случаях и 1 в 156 случаях, тогда как при использовании взвешивания классов было предсказано 0 в 709 случаях и 1 в 87 случаях.

Наблюдается значительное улучшение в предсказании класса 1, который был сильно недооценен в несбалансированных данных. В то же время количество правильных предсказаний класса 0 уменьшилось, что является естественным следствием снижения веса этого класса при взвешивании.

Можно сделать вывод, что взвешивание классов дало лучший результат в предсказании меньшего класса 1, но не сильно улучшило предсказание большего класса 0.

В таблице 8 приведена матрица ошибок при взвешивании классов.

Таблица 8. Матрица ошибок при взвешивании классов

Истинный\Предсказанный	0	1
0	709	491
1	69	87

ROC AUC и PR AUC — это метрики, используемые для оценки производительности классификатора. ROC AUC — это площадь под кривой ROC (Receiver Operating Characteristic), которая показывает зависимость между долей верно классифицированных положительных объектов (True Positive Rate, TPR) и долей ложно классифицированных отрицательных объектов (False Positive Rate, FPR), при варьировании порога классификации. PR AUC — это площадь под кривой Precision-Recall, которая показывает зависимость между долей верно классифицированных положительных объектов (Precision) и долей найденных положительных объектов (Recall), при варьировании порога классификации.

AP (Average Precision) — это метрика, используемая для оценки качества моделей машинного обучения в задачах бинарной классификации. Она вычисляется как среднее значение точности (Precision) на разных уровнях отсечения (Recall) при изменении порога решающего правила. AP представляет собой площадь под кривой Precision-Recall и может принимать значения от 0 до 1, где значение 1 соответствует идеальному качеству модели. Чем выше значение AP, тем лучше модель распознает положительный класс при различных уровнях отсечения.

В случаях, когда количество положительных и отрицательных объектов примерно равны, обе метрики могут быть полезны для оценки производительности классификатора. Однако, когда классы несбалансированы, PR AUC может быть более информативным [10], поскольку он фокусируется на предсказании положительных

объектов, что может быть более важно для приложений с низкой частотой положительных объектов. В таких случаях, ROC AUC может быть искаженным, поскольку даже классификатор со случайной производительностью может достичь высокой TPR при низкой FPR на несбалансированных данных.

Площадь под ROC-кривой (AUC) является мерой качества классификатора: чем выше значение AUC, тем лучше работает классификатор.

По данной метрике можно сделать следующие выводы для рассмотренных методов решения проблемы дисбаланса классов.

1. Несбалансированные данные имеют очень низкое значение AUC — всего 0.51, что говорит о плохой разделимости классов.

2. Метод увеличения выборки (SMOTE) показал значительный прирост в AUC — до 0.67, что может говорить об улучшении разделимости классов и работоспособности классификатора.

3. Метод уменьшения выборки (RandomUnderSampler) показал небольшое улучшение AUC по сравнению с несбалансированными данными — до 0.61, но это значение ниже, чем при использовании метода увеличения выборки.

4. Метод взвешивания классов также показал небольшое улучшение AUC по сравнению с несбалансированными данными — до 0.60. Однако, этот метод не дал такого значительного улучшения, как метод увеличения выборки.

Исходя из этого можно сделать вывод, что для данного набора данных метод увеличения выборки (SMOTE) показал лучшие результаты по метрике ROC AUC.

На рисунке 2 изображена ROC кривая для рассмотренных методов.

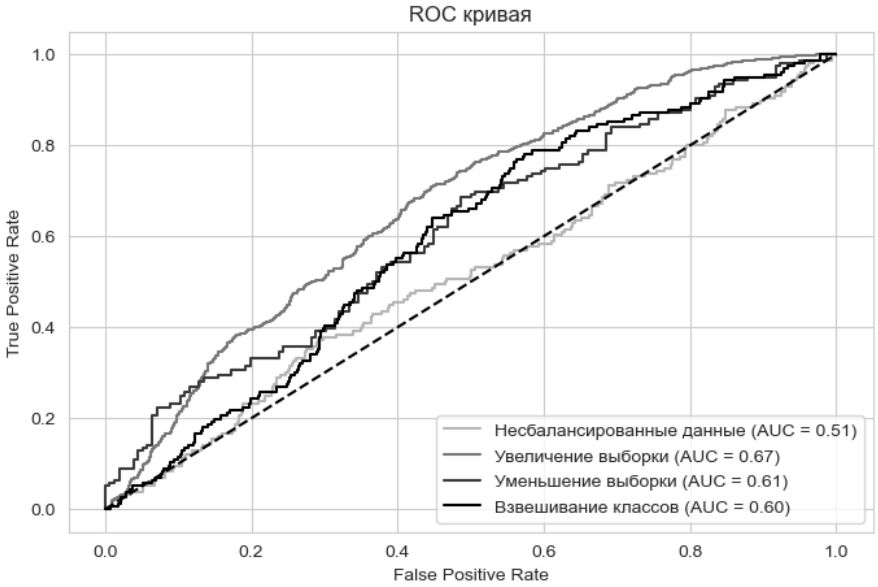

Рис. 2 - ROC кривая

Исходя из графиков для метрики PR AUC, изображенной на рисунке 3, можно сделать следующие выводы.

1. Несбалансированные данные имеют низкое значение PR AUC (AP=0.12), что означает, что классификатор плохо справляется с определением положительных объектов, при этом много объектов относится к отрицательному классу.

2. Увеличение выборки с помощью SMOTE улучшило качество классификации по сравнению с несбалансированными данными (AP=0.62), что свидетельствует о том, что увеличение числа положительных объектов в выборке сделало классификацию лучше.

3. Уменьшение выборки с помощью RandomUnderSampler также улучшило качество классификации по сравнению с несбалансированными данными (AP=0.63), что означает, что удаление части объектов отрицательного класса позволило классификатору лучше определить положительные объекты.

4. Взвешивание классов с помощью параметра sample_weight в SVM не улучшило качество классификации по сравнению с несбалансированными данными (AP=0.14), что может свидетельствовать о неэффективности данного метода в данном случае.

Таким образом, PR AUC показывает, что наилучшее качество классификации было достигнуто с помощью увеличения и уменьшения выборки, а взвешивание классов не привело к улучшению результата.

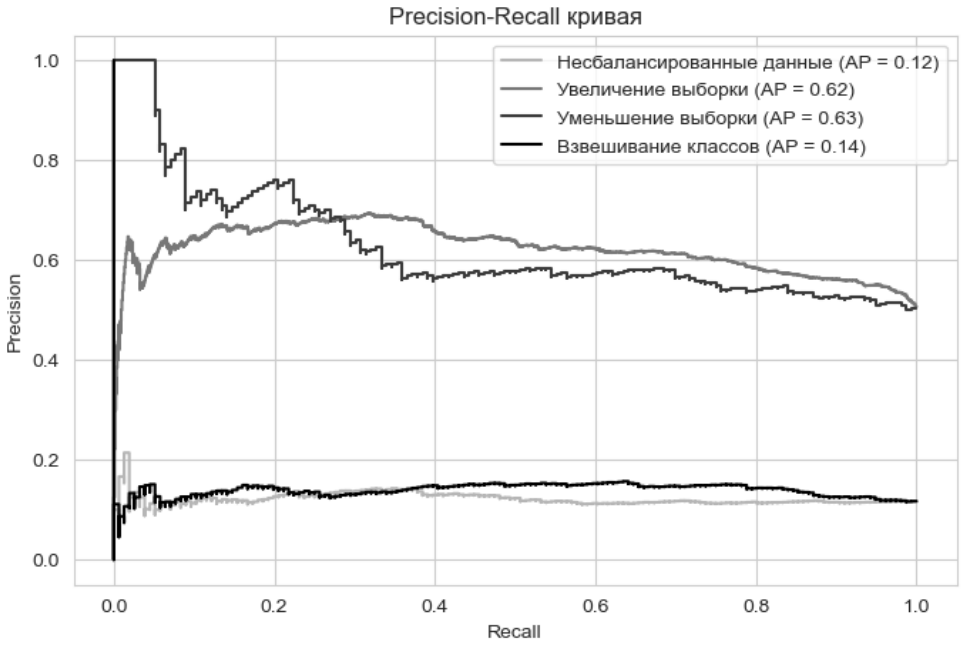

Рис. 3 - PR кривая

Заключение

В ходе анализа были проведены эксперименты на несбалансированных данных и применены различные методы борьбы с проблемой дисбаланса классов, такие как увеличение и уменьшение выборки, а также взвешивание классов.

При анализе результатов экспериментов было обнаружено, что даже при высоких значениях метрик качества, таких как Accuracy, Recall, Precision и F1-score, матрица ошибок может указывать на низкое качество классификации. Поэтому при оценке качества классификатора необходимо учитывать все метрики, включая матрицу ошибок.

Было показано, что для увеличения выборки был использован метод SMOTE, который позволил улучшить результаты классификации по метрикам ROC AUC и PR AUC. Для уменьшения выборки был использован метод RandomUnderSampler, который также позволил улучшить качество классификации. Для взвешивания классов был использован параметр sample_weight при обучении SVM и compute_class_weight из библиотеки sklearn.

Анализ метрик качества показал, что увеличение и уменьшение выборки позволяет значительно улучшить качество классификации по метрикам ROC AUC и PR AUC, в то время как метод взвешивания классов подтвердил, что на несбалансированных данных PR AUC является более информативной.

Таким образом, для решения проблемы дисбаланса классов необходимо применять различные методы в зависимости от конкретной задачи и доступных данных, а также проводить полный анализ метрик качества классификации.

Список литературы

1. Varmedja D. et al. Credit card fraud detection-machine learning methods //2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH). – IEEE, 2019. – С. 1-5.

2. Демидова Л.А., Соколова Ю.С. Использование SVM-алгоритма для уточнения решения задачи классификации объектов с применением алгоритмов кластеризации // Вестник Рязанского государственного радиотехнического университета. 2015. № 51. С. 103-113.

3. Demidova L.A. Two-Stage Hybrid Data Classifiers Based on SVM and KNN Algorithms // Symmetry. 2021. Vol. 13(4). С. 615.

4. Spelmen V. S., Porkodi R. A review on handling imbalanced data //2018 international conference on current trends towards converging technologies (ICCTCT). – IEEE, 2018. – С. 1-11.

5. Chawla N. V. et al. SMOTE: synthetic minority over-sampling technique //Journal of artificial intelligence research. – 2002. – Т. 16. – С. 321-357.

6. I. Tomek, “Two modifications of CNN,” In Systems, Man, and Cybernetics, IEEE Transactions on, vol. 6, pp 769-772, 1976.

7. Imbalanced | Credit Approval [Электронный ресурс] ¬ URL: https://www.kaggle.com/datasets/enesztrk/credit-approval

8. Van der Maaten L., Hinton G. Visualizing data using t-SNE //Journal of machine learning research. – 2008. – Т. 9. – №. 11.

9. Sokolova M., Lapalme G. A systematic analysis of performance measures for classification tasks //Information processing & management. – 2009. – Т. 45. – №. 4. – С. 427-437.

10. Saito T., Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets //PloS one. – 2015. – Т. 10. – №. 3. – С. e0118432.

References

1. Varmedja D. et al. Credit card fraud detection-machine learning methods //2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH). – IEEE, 2019. – С. 1-5.

2. Demidova L.A., Sokolova Yu.S. Using the SVM-algorithm to refine the solution of the object classification problem using clustering algorithms. Bulletin of the Ryazan State Radio Engineering University. 2015. No. 51. P. 103-113.

3. Demidova L.A. Two-Stage Hybrid Data Classifiers Based on SVM and KNN Algorithms // Symmetry. 2021. Vol. 13(4). C. 615.

4. Spelmen V. S., Porkodi R. A review on handling imbalanced data //2018 international conference on current trends towards converging technologies (ICCTCT). – IEEE, 2018. – C. 1-11.

5. Chawla N. V. et al. SMOTE: synthetic minority over-sampling technique //Journal of artificial intelligence research. – 2002. – T. 16. – C. 321-357.

6. I. Tomek, “Two modifications of CNN,” In Systems, Man, and Cybernetics, IEEE Transactions on, vol. 6, pp 769-772, 1976.

7. Imbalanced | Credit Approval [Электронный ресурс] ¬ URL: https://www.kaggle.com/datasets/enesztrk/credit-approval

8. Van der Maaten L., Hinton G. Visualizing data using t-SNE //Journal of machine learning research. – 2008. – T. 9. – №. 11.

9. Sokolova M., Lapalme G. A systematic analysis of performance measures for classification tasks //Information processing & management. – 2009. – T. 45. – №. 4. – C. 427-437.

10. Saito T., Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets //PloS one. – 2015. – T. 10. – №. 3. – C. e0118432.