Мягков А. А., Раскатова М. В. Исследование эффективности методов классификации для определения мошеннических банковских транзакций // Международный журнал информационных технологий и энергоэффективности. – 2022. – Т. 7 № 2(24) с. 15–21

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.896

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ МЕТОДОВ КЛАССИФИКАЦИИ ДЛЯ ОПРЕДЕЛЕНИЯ МОШЕННИЧЕСКИХ БАНКОВСКИХ ТРАНЗАКЦИЙ

Мягков А.А., Раскатова М.В.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Национальный исследовательский университет «МЭИ», Россия (111250, г.Москва, ул. Красноказарменная, д.14); e-mail: myagkov.andrey.ru@mail.ru

В статье рассматривается исследование работы методов классификации в определении мошеннических банковских транзакций и сравнение полученных результатов. Обработаны исходные данные. Проведен ряд экспериментов по обучению моделей на различных обучающих выборках, полученных в результате балансирования данных, с использованием языка программирования Python. Результаты предсказаний моделей оценены по ряду выбранных метрик.

Ключевые слова: классификация, транзакция, метрика, машинное обучение, информационная система.

STUDY OF THE EFFICIENCY OF CLASSIFICATION METHODS FOR IDENTIFYING FRAUDIOUS BANKING TRANSACTIONS

Myagkov A.A., Raskatova M.V.

National Research University “Moscow Power Engineering Institute”, Russia (111250, Moscow, Krasnokazarmennaya street, 14); e-mail: myagkov.andrey.ru@mail.ru

The article discusses the study of the work of classification methods in the identification of fraudulent banking transactions and a comparison of the results obtained. Processed initial data. Several experiments were carried out to train models on various training samples obtained because of data balancing using the Python programming language. The results of model predictions are evaluated by several selected metrics.

Keywords: classification, transaction, metrics, machine learning, information system.

В современном мире большинство операций с деньгами совершаются путём банковских транзакций. Благодаря развитию инфраструктуры приема и обслуживания банковских карт, все больше людей переходит к такому удобному способу оплаты как безналичная оплата. Однако такие транзакции могут быть уязвимы к несанкционированному воздействию третьих лиц, с целью получения доступа к счету и последующему хищению денежных средств. Согласно данным ЦБ, только за 2021 год мошенники похитили 13.5 млрд рублей, что больше, чем в 2020 году (9.7 млрд рублей), а смогли вернуть только 6.8% или 920.5 млн рублей, что меньше, чем в 2020 году (1.1 млрд рублей) [1]. Поэтому важно, чтобы все банковские транзакции подвергались тщательной проверке на предмет мошеннических действий.

Проверка проходящей транзакции происходит в антифрод-системе в процессинговом и авторизационном центрах банка. Там транзакция проверяется на наличие в стоп-листах, проверяется корректность реквизитов и IP-адрес оплаты, чтобы адрес не сильно отличался от обычного и не происходил из стран, где высокий уровень мошеннических действий. Помимо стандартных проверок, в антифрод-системе также используются проверки с помощью машинного обучения, в частности определение мошеннических транзакций с помощью кластеризации и классификации. В данной статье будет рассмотрено определение мошеннических транзакций с помощью бинарной классификации, а именно разделение транзакции на два класса – подлинные и мошеннические, с помощью языка **Python**. В качестве методов классификации были выбраны *дерево решений* (decision tree), *алгоритм k-ближайших соседей* (k-nearest neighbors) и *логистическая регрессия* (logistic regression).

В качестве данных для анализа был выбран датасет «Определение мошеннических транзакций», состоящий из 284807 записей транзакций, сделанных с европейских банковских карт за период сентября 2013 года. У каждой записи 31 поле среди которых:

- Time – количество секунд, прошедших между данной транзакцией и первой транзакцией в наборе данных;
- V1-V28 – параметры без названия, в связи с конфиденциальностью;
- Amount – сумма транзакции;
- Class – переменная, определяющая является ли транзакция мошеннической.

На рисунке 1 показаны первые пять записей из датасета.

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V2
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.12853
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.16717
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.32764
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.64737
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.20601

Рисунок 1 – Первые пять записей датасета

Из всех записей всего 492 транзакции помечены как мошеннические, что делает этот датасет сильно несбалансированным. Также известно, что столбцы V1-V28 были получены в результате PCA (метод главных компонент), что уже подразумевает в себе использование масштабирования, поэтому для корректного анализа данных были масштабированы оставшиеся столбцы Time и Amount.

Полученные данные были разбиты стратифицированной выборкой, то есть с сохранением распределения классов. Размер тестовой выборки составил 20% от общего числа записей. Для построения и обучения моделей с выбранными методами классификации была использована библиотека **Scikit-learn**.

Для оценки эффективности работы методов классификации используются метрики – специальные показатели, отображающие работу модели. Метрика выбирается относительно поставленной задачи, а неправильно выбранная метрика может привести к заблуждению и неоптимальному решению. Так как в данной задаче бинарная классификация, возможны 4 исхода предсказания модели:

- истинно положительный (TP) – модель предсказала 1 и истинный результат 1;
- ложно положительный (FP) – модель предсказала 1 и истинный результат 0;

- ложно отрицательный (FN) – модель предсказала 0 и истинный результат 0;
- истинно отрицательный (TN) – модель предсказала 0 и истинный результат 1.

В данной задаче были использованы метрики *precision*, *recall* и *F-мера*. Наиболее простая и поэтому распространенная метрика Accuracy, показывающая процент правильно угаданных классов, не использовалась в данной задаче, так как в случае сильно несбалансированных данных она может показывать сильно завышенные результаты, не отображающие способность модели предсказывания миноритарного класса [2].

Precision – метрика, показывающая отношение количества истинно положительных результатов (TP) к количеству всех результатов отмеченных положительными моделью (TP + FP), как показано на формуле (1). Данная метрика наиболее ценна в задачах, где ложно положительный результат необходимо минимизировать (положительный результат означает дорогостоящую или долгую проверку).

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

Recall – метрика, отображающая отношение количества данных истинно положительных результатов (TP) ко всем результатам, помеченных как положительные (TP + FN), как показано в формуле (2). Данная метрика важна в случаях, когда ценно распознать наибольшее количество всех данных положительного класса (например, определение болезни).

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Однако зачастую важны как Precision, так и Recall и необходимо найти оптимальный баланс между ними, для этого есть метрика **F-мера** и определяется она по формуле (3).

$$F = (\beta^2 + 1)\frac{Precision * Recall}{\beta^2 * Precision + Recall} \quad (3),$$

где β – переменная, позволяющая отдать большее предпочтение Precision или Recall. При $\beta>1$ приоритет отдается Recall, при $0<\beta<1$ приоритет точности отдается Precision, а при $\beta = 1$ получается частный случай F-меры – мера F_1, где равный баланс между Precision и Recall.

Приоритет должен определяться исходя из данных банка, таких как количество сообщений о мошенничестве, количество жалоб о блокировке подлинных транзакций и т. д. Так как в данной задаче такой информации нет, будем считать, что метрики одинаково важны, а основными метриками эффективности модели будут мера F1 и время выполнения.

Работа методов классификации напрямую зависит от их гиперпараметров, поэтому их подбор может улучшить эффективность модели. Так как перебор этих параметров вручную может занять большое количество времени и не обязательно приведет к наилучшему результату, параметры были подобраны с помощью инструмента **GridSearchCV**, который запускает поиск по сетке параметров, т. е. перебирает все возможные значения параметров в выбранном диапазоне и показывает набор параметров, показавший лучшие результаты в выбранной метрике. Оптимизируемыми параметрами для дерева решений выступили -

критерий разделения, максимальная глубина дерева и минимальное количество выборок, для k-ближайших соседей – количество соседей и алгоритм определения ближайших, а для логистической регрессии – метод регуляризации и обратная величина регуляризации.

В таблице 1 показаны результаты работы моделей классификации по выбранным метрикам после подбора гиперпараметров.

Таблица 1 – Таблица значений моделей классификации по выбранным метрикам

	Значение Precision среди мошеннических	**Значение Recall среди мошеннических**	**Значение F_1 среди мошеннических**	**Время выполнения**
Дерево решений	0.8652	0.7857	0.8235	4.30
Алгоритм k-ближайших соседей	0.9494	0.7653	0.8475	4:29.45
Логистическая регрессия	0.8636	0.5816	0.6951	2.64

Так как несбалансированные данные могут влечь за собой ошибки, связанные с тем, что обученные модели будут предполагать, что в основном транзакции являются не мошенническими, были рассмотрены методы для корректировки распределения классов, такие как Недостаточная выборка (Undersampling) и Передискретизация (Oversampling). Стоит понимать, что корректировка распределения происходит только для обучающей выборки, так как проверять работу модели мы хотим на изначальных данных.

В случае недостаточной выборки данные мажоритарного класса удаляются до тех пор, пока не будет достигнут нужный баланс. Одним из способов недостаточной выборки является алгоритм **NearMiss**, в котором удаляются те данные мажоритарного класса, которые наиболее близко находятся к данным миноритарного класса, тем самым делая большую разницу между классами для последующей классификации. Применения алгоритма NearMiss осуществляется с помощью одноименной функции входящей в состав пакета функций under_sampling библиотеки imblearn.

При передискретизации же балансировка происходит путем добавления данных миноритарного класса. Наиболее простой способ – дублирование существующих данных, что хоть и сбалансирует данные, но не даст никаких новых данных для обучения модели. Улучшением данного способа является технология **SMOTE** (Synthetic Minority Oversampling Technique), которая генерирует новые данные похожие на те, что уже есть в датасете. Применение алгоритма SMOTE осуществляется с помощью одноименной функции из пакета функций over_sampling.

Для моделей, обученных на сбалансированных данных, также были определены и применены оптимальные гиперпараметры с помощью GridSearchCV. Результаты работы моделей классификации, обученных на сбалансированных данных, полученных в результате работы методов NearMiss и SMOTE показаны в таблицах 2 и 3 соответственно.

Таблица 2 - Таблица значений моделей классификации по выбранным метрикам при NearMiss

	Значение Precision среди мошеннических	**Значение Recall среди мошеннических**	**Значение F_1 среди мошеннических**	**Время выполнения**
Дерево решений	0.0053	0.9286	0.0105	0.02
Алгоритм k-ближайших соседей	0.0118	0.9082	0.0233	2.15
Логистическая регрессия	0.0094	0.9184	0.0186	0.04

Таблица 3 - Таблица значений моделей классификации по выбранным метрикам при SMOTE

	Значение Precision среди мошеннических	**Значение Recall среди мошеннических**	**Значение F_1 среди мошеннических**	**Время выполнения**
Дерево решений	0.0240	0.9082	0.0467	5.50
Алгоритм k-ближайших соседей	0.5743	0.8673	0.6911	8:38.74
Логистическая регрессия	0.0608	0.9286	0.1142	5.77

Из полученных результатов по метрике F_1 видно, что модель, построенная при использовании технологии SMOTE, работает лучше, чем при NearMiss, но намного хуже, чем при изначальных несбалансированных данных. В данном случае балансировка данных не улучшила результат и использовать ее не стоит.

Следующим шагом была попытка улучшить результаты по методу логистической регрессии путем его модификации. Логистическая регрессия – разновидность множественной регрессии, использующая логистическую функцию для моделирования зависимости бинарной выходной переменной от набора входных данных. Не стоит путать с регрессионным алгоритмом, так как логистическая регрессия — это алгоритм классификации [3]. Классификация логистической регрессии работает путем подсчета вероятности принадлежности одному из классов. В случае бинарной классификации вероятность того, что значение принадлежит к одному классу (например, что транзакция мошенническая) обозначим P+, а вероятность того, что значение принадлежит ко второму классу (например, транзакция подлинная) P-. Тогда P+ = 1 - P-, а сами вероятности лежат в диапазоне [0, 1]. Определение класса происходит путем выбора наибольшей вероятности, так в случае бинарной классификации пороговым значением является 0.5.

Однако такое пороговое значение, может быть, не всегда целесообразно и в определенных случаях результаты по метрике F_1 бы улучшились изменой порога. Например, в случае результатов начальных несбалансированных данных видно, что у модели показатели метрики Precision выше, чем у метрики Recall и более высоких результатов метрики F_1 можно добиться, снизив порог.

Так как у базового класса логистической регрессии нет возможности установки своего порогового значения, бы написан класс, расширяющий возможности базового. В классе переопределен метод предсказания так, чтобы можно было устанавливать новое пороговое значение и написан метод помогающий определить оптимальное пороговое значение из

тренировочных данных, дающее максимальное значение по метрике f1. Результат работы улучшенной логистической регрессии по метрике F_1 составил 0.7568, что лучше результатов обычной логистической регрессии на 6%, получилось это путем понижения Precision и повышения Recall.

В случаях, когда модель обучается на тренировочных данных и оптимизирует под них параметры модели, есть вероятность, что модель слишком хорошо научится определять тренировочные данные, а тестовые данные будет определять намного хуже [4]. Для проверки корректности работы построенных моделей на предмет переобучения была произведена кросс валидация, суть ее заключается в следующем:

- Разделить тренировочные данные на n непересекающихся одинаковых по объему частей;
- Обучить модель на k-1 частей;
- Протестировать на оставшейся части;
- Повторить k раз, каждый раз выбирая новую часть для проверки.

Разбиение тренировочных данных происходило с использованием метода **StratifiedKFold**, который разбивает данные на k равных частей с одинаковым распределением классов в них. Результаты кросс валидации основных моделей показали похожие значения на полученные у моделей ранее, а разброс значений везде не превышал 2%, что хороший показатель и означает, что переобучение не происходит.

Из проделанного исследования можно сделать вывод, что наилучшей моделью для определения мошеннических транзакций является модель, построенная с использованием дерева решений при несбалансированных данных с использованием оптимальных гиперпараметров. Хотя метод k-ближайших на тех же данных по метрике F_1 и показал на 2% больший результат, время, затраченное на обучение и предсказание, занимает 4.5 минуты, что несоизмеримо больше, чем у дерева решений.

Список литературы

1. Годовой отчет Банка России за 2021 год – URL: https://www.cbr.ru/Collection/Collection/File/40915/ar_2021.pdf (дата обращения: 27.11.2021). – Режим доступа: открытая информация. – Текст: электронный
2. Бенджио, И. Глубокое обучение / И. Бенджио, Я. Гудфеллоу, А. Курвилль – ДМК Пресс, 2017. – 652с.
3. Шолле, Ф. Глубокое обучение на Python / Ф. Шолле – Питер, 2018. – 400с.
4. Орельен, Ж. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow. Концепции, инструменты и техники для создания интеллектуальных систем / Ж. Орельен, 2018. – 688 с.

References

1. Bank of Russia Annual Report for 2021 – URL: https://www.cbr.ru/Collection/Collection/File/40915/ar_2021.pdf (date of access: 11/27/2021). – Access mode: open information. – Text: electronic
2. Bengio, I. Deep Learning / I. Bengio, Y. Goodfellow, A. Courville - DMK Press, 2017. - 652p.
3. Chollet, F. Deep learning in Python / F. Chollet - Peter, 2018. - 400p.

4. Aurelien, J. Applied Machine Learning with Scikit-Learn and TensorFlow. Concepts, tools and techniques for creating intelligent systems / J. Aurelien, 2018. - 688 p.