ОБНАРУЖЕНИЕ АНОМАЛИЙ В НАБОРЕ ДАННЫХ С ПОМОЩЬЮ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ БЕЗ УЧИТЕЛЯ ISOLATION FOREST И LOCAL OUTLIER FACTOR

ANOMALY DETECTION IN THE DATASET USING UNSUPERVISED LEARNING ALGORITHMS ISOLATION FOREST AND LOCAL OUTLIER FACTOR

УДК 004.5

Попова И.А., студент, popovai1@student.bmstu.ru
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана, кафедра «Системы обработки информации и управления»

Popova I.A., student, popovai1@student.bmstu.ru
Russia, 105005, Moscow, MSTU N.E. Bauman, Department of Information Processing and Management Systems

Аннотация

Обнаружение выбросов, также именуемое, как обнаружение аномалий, является важной задачей в области интеллектуального анализа данных для выделения в наборе данных нетипичных объектов. Это важнейший этап перед построением моделей машинного обучения. В данной статье рассматриваются алгоритмы обучения без учителя для обнаружения аномалий в наборе данных: изоляционный лес (Isolation Forest) и локальный уровень выброса (Local outlier Factor).

Annotation

Outlier detection, also referred to as anomaly detection, is an important data mining task for highlighting atypical features in a dataset. It is a critical step before building machine learning models. This article discusses unsupervised learning algorithms for detecting anomalies in a dataset: Isolation Forest and Local outlier Factor.

Ключевые слова: аномалия, алгоритм, обнаружение, обучение без учителя, набор данных, модель.

Key words: anomaly, algorithm, detection, data set, unsupervised learning, model.

Обнаружение аномалий - это процесс идентификации точек данных, элементов выборки, наблюдений или событий, свойства которых отличаются от свойств большинства обрабатываемых образцов в наборе данных. Аномалии, присутствующие в статистической выборке могут указывать на подозрительный сетевой трафик, неисправные приборы измерения, ошибки измерений или просто определять данные, которые стоит удалить из набора данных перед последующим анализом. Обнаружение аномалий широко используется в ряде дисциплин: физике, экономике, машинном обучении, информационной безопасности и др.

Согласно определению, данному Хокинсом [1], аномалия (выброс), сильно отличается от наблюдаемых нормальных объектов и, как предполагается, возникла в результате воздействия другого механизма.

Объясним понятие аномалии на основе плотности:

- это объекты, которые находятся в областях с низкой плотностью или низкой локальной / относительной плотностью.

Объясним понятие аномалии на основе расстояния:

- это объекты с менее чем k соседями в наборе данных, где соседом является объект на расстоянии R.

Различают два типа выбросов:

- глобальные выбросы - точки данных, которые значительно отличаются от остальной части набора данных.
- локальные выбросы - точки данных, которые значительно отличаются от своих соседей в наборе данных.

Рассмотрим задачу обнаружения аномалий следующим образом. Пусть имеется набор данных X, состоящий из объектов, где каждый объект x описывается множеством признаков и представлен вектором из $\mathbb{R}^d$. Для каждого объекта из набора данных X определить значение 0, если этот объект классифицируется, как нормальный и 1, если объект считается аномальным.

В [3] рассмотрен ряд методов обнаружения аномалий в наборе данных. Так как каждый алгоритм обнаружения аномалий трактует понятие аномальности по-разному, то есть предлагает свой метод поиска выбросов, соответственно результаты их работы будут отличны.

В данной работе рассмотрим два алгоритма обучения без учителя для обнаружения аномалий: изоляционный лес (Isolation Forest) и локальный уровень выброса (Local outlier Factor).

Изоляционный лес (Isolation Forest) для обнаружения аномалий

Данный алгоритм [2] является техникой машинного обучения без учителя, основанной на принципе Монте-Карло.

Термин «изоляция» означает процесс отделения рассматриваемого экземпляра от остальных экземпляров выборки. Поскольку предполагается, что аномалий мало и они разные, и поэтому они более подвержены изоляции.

В ходе работы алгоритма пространство признаков разделяется случайным образом. Метод отсекает изолированные точки от нормальных кластеризованных данных. Результат работы алгоритма представляет собой усредненную оценку, полученную в ходе запуска стохастического алгоритма несколько раз.

Работа алгоритма заключается в построении случайного бинарного дерева, в котором корень – все пространство признаков, а узел представлен случайным признаком и порогом разбиения. Порог разбиения выбирается из равномерного

распределения на отрезке от минимального до максимального значения выбранного признака. При тождественном совпадении объектов в узле завершаем работу алгоритма. Глубина листа в дереве будет соответствовать значению anomaly_score алгоритма.

Оценка аномалии дается следующим образом:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}},$$

где $h(x)$ - длина пути до наблюдения x, $c(n)$ - средняя длина пути безуспешного поиска в двоичном дереве поиска, а n – число точек данных. Если s принимает значения, близкие к 1, тогда объект x с большой вероятностью будет аномалией, однако, если s принимает значения меньше 0,5, в таком случае исследуемый объект признается нормальным.

Isolation Forest чувствителен только к глобальным выбросам и слабо справляется с локальными выбросами.

Локальный уровень выброса (Local outlier Factor) для обнаружения аномалий

Метод LOF является техникой машинного обучения без учителя для обнаружения аномалий, которая вычисляет локальное отклонение плотности данной точки по отношению к ее соседям. То есть производится оценка плотности распределения объектов выборки. Если плотность точки намного меньше, чем плотность ее соседей (LOF >> 1), то есть точка находится далеко от плотных областей, в таком случае рассматриваемая точка является выбросом.

Данный алгоритм основан на методе k ближайших соседей (kNN) и описан в работе [4].

Определяется досягаемость точки x относительно точки y с помощью выражения:

$$R_k(x,y) = \max(\rho(x,y), D_k(y)),$$

где $D_k(y)$ – выражает расстояние от точки данных y до k ближайшего соседа.

Тогда оценку LOF определим с помощью выражения:

$$LOF_k(x) = \underset{y \in N_k(x)}{\text{mean}} \frac{AR_k(x)}{AR_k(y)},$$

где $AR_k(x)$ – определяет среднюю досягаемость точки данных x относительно k своих ближайших соседей, $AR_k(y)$ – определяет множество k ближайших соседей точки данных x. Если LOF принимает значения, близкие к 1, тогда объект x будет считаться нормальным, однако, если LOF >> 1, в таком случае исследуемый объект признается аномалией.

LOF определит выброс с учетом плотности рассматриваемой области. Алгоритм справляется, когда плотность данных не одинакова во всем наборе данных и показывает хорошие результаты при обнаружении локальных выбросов, однако имеет высокую вычислительную сложность.

Алгоритм показывает хорошую работу в ситуациях, когда в наборе данных находятся объекты разных классов, однако они не являются аномалиями.

Сравним работу описанных выше алгоритмов на наборе данных Credit Card Fraud Detection, который содержит данные об анонимных транзакциях с кредитными картами, помеченные как мошеннические или подлинные. Каждая транзакция в наборе данных помечена меткой (1 – мошенническая транзакция и 0 – подлинная). Информация о типе транзакции хранится в переменной Class. Так как данные конфиденциальны, признаки V1 - V28 были преобразованы с помощью метода PCA. Также в наборе данных нет пропущенных значений.

На рисунке 1 показана информация о типе данных каждого признака и количестве непустых значений.

```
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

Рисунок 1. Краткая сводка о наборе данных.

Построим гистограмму для признака Class (Класс), который является бинарным (содержит значение 0 или 1). Значением 1 закодированы мошеннические транзакции, а признаком 0 – подлинные. На рисунке 2 представлена гистограмма распределения для признака Class.

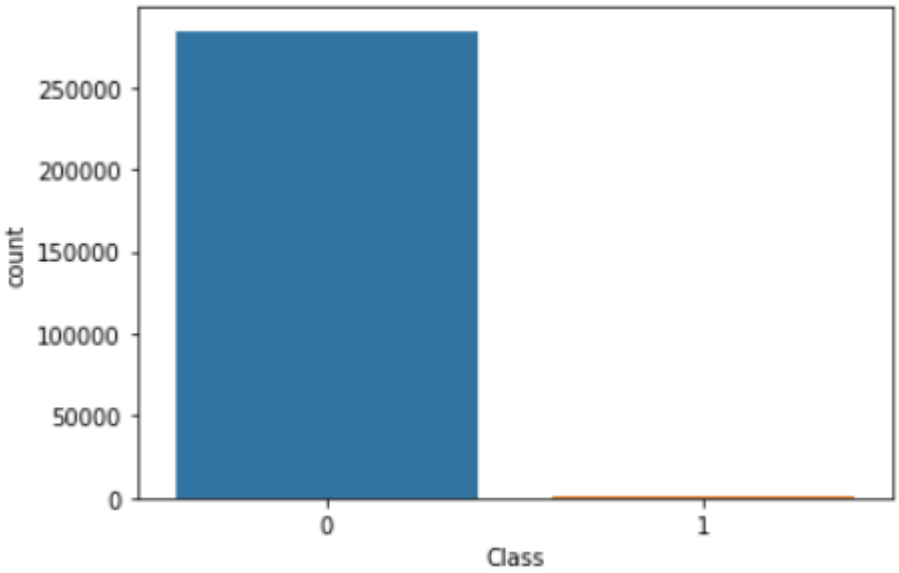

Рисунок 2. Гистограмма распределения для признака Class.

Соответственно набор данных содержит большое число легитимных транзакции и 0,17% мошеннических, которые будем считать аномалиями.

Отобразим одномерное распределение мошеннических и нормальных транзакций в зависимости от времени суток.

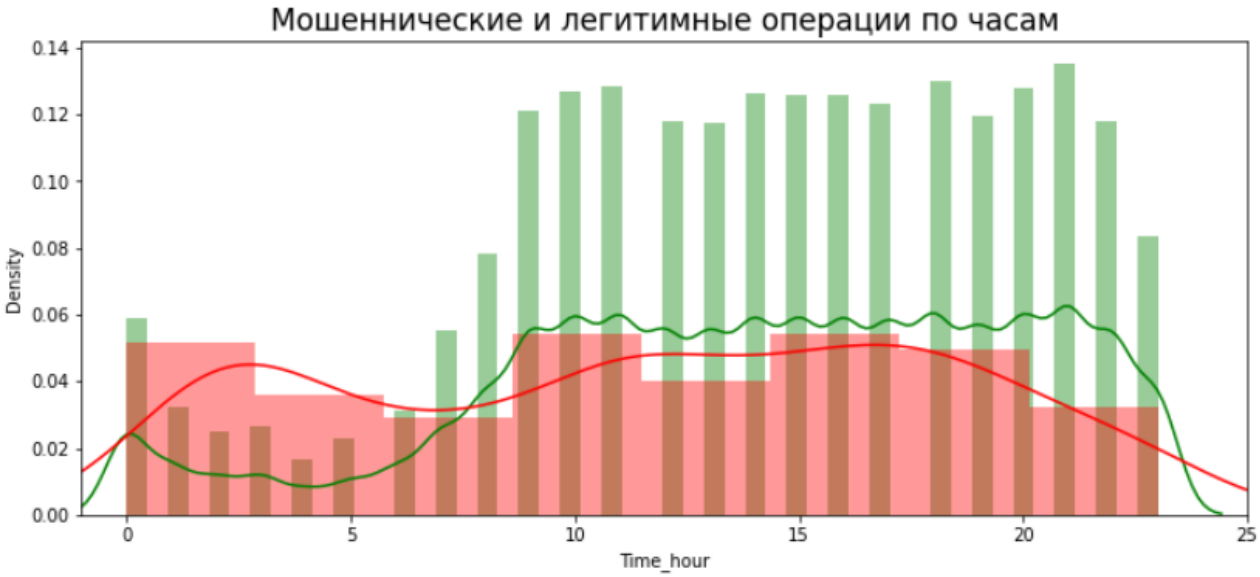

Рисунок 3. Гистограмма и плотность распределения для транзакций.

Можно сделать предположение, что время суток имеет некоторое влияние на количество случаев мошенничества.

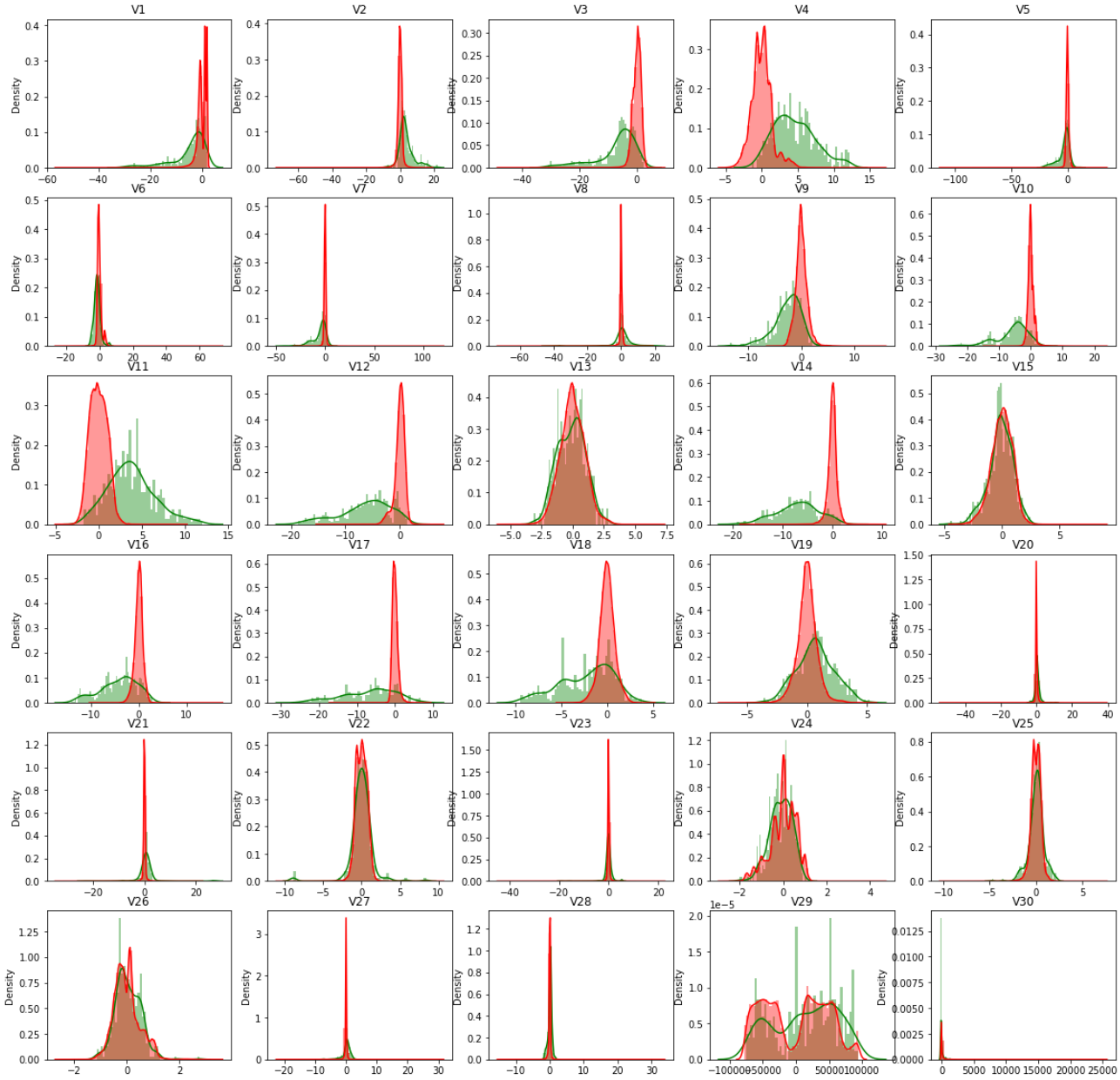

Рисунок 4. Парные диаграммы.

На полученных диаграммах распределение данных нормальных операций и случаев мошенничества для некоторых признаков, таких как V13, V20, V25 перекрывается и кажется одинаковым. Эти признаки не позволяют отличить обычные транзакции от мошеннических.

Определим оценку аномалии для каждого образца из выборки с помощью классов IsolationForest и LocalOutlierFactor из библиотеки sklearn. На основе полученных оценок посчитаем точность построенных моделей.

Результаты вычислений приведены в таблице 1.

Таблица 1. Результаты работы алгоритмов IsolationForest и LocalOutlierFactor.

	IsolationForest	LocalOutlierFactor
Точность обнаружения нормальных операций	0,958	0,988
Точность обнаружения мошеннических операций	0,86	0,236
Время работы алгоритма, сек	3,36	212,96

По результатам вычислений видно, что алгоритм Isolation Forest справляется с задачей обнаружения аномалий более качественно и быстро. Представим результат работы методов в наглядной форме.

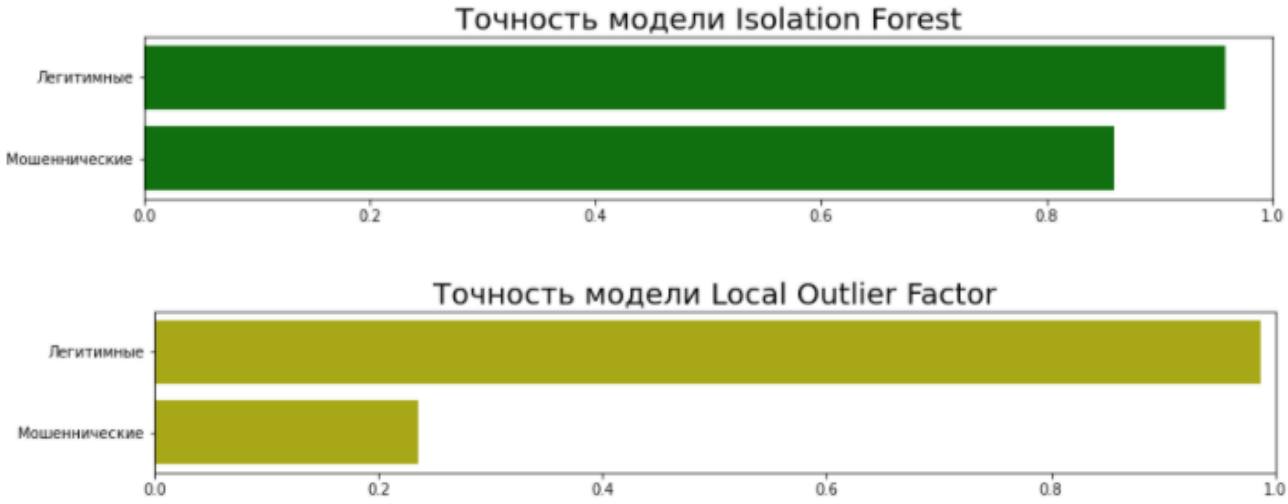

Рисунок 5. Результаты работы алгоритмов IsolationForest и LocalOutlierFactor.

Обе модели Isolation Forest и Local Outlier Factor показали похожие результаты при прогнозировании легитимных транзакций, однако модель Isolation Forest гораздо лучше справилась с задачей обнаружения мошеннических транзакций. Скорее всего выборка содержала большое число глобальных выбросов, к которым чувствителен метод Isolation Forest. Таким образом можно сделать вывод о том, что метод обнаружения аномалий Isolation Forest показал достаточно хорошую работу на исследуемом наборе данных. Возможно создание

ансамбля из этих моделей позволит более качественно и быстро справляться с задачей обнаружения аномалий в выборке.

Литература

1. Д. М. Хокинс, Выявление выбросов, том. 11, Шпрингер, Берлин, Германия, 1980.
2. Лю Ф. Т., Тинг К. М., Чжоу З. Х. Изоляционный лес // Data Mining, 2008. ICDM’08. Восьмая международная конференция IEEE по. - IEEE, 2008. - С. 413-422.
3. Чандола В., Банерджи А., Кумар В. Обнаружение аномалий: исследование // ACM Computing Surveys. 2009. Том 41, №. 3. Статья 15. DOI: 10.1145 / 1541880.1541882
4. Бреуниг М., Кригель Х.-П., Т. Нг Р., Сандер Дж. LOF: Определение локальных выбросов на основе плотности // Труды Международной конференции ACM SIGMOD по обработке данными. ACM Press. С. 93-104.
5. Jupyter Notebook [Электронный ресурс] https://jupyter.org/ (Дата обращения: 25.11.2020).
6. Google Colaboratory [Электронный ресурс] https://colab.research.google.com/ (Дата обращения: 25.11.2020).

Literature

1. D. M. Hawkins, Identification of Outliers, vol. 11, Springer, Berlin, Germany, 1980.
2. Liu F. T., Ting K. M., Zhou Z. H. Isolation Forest // Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on. – IEEE, 2008. – C.413-422.
3. Chandola V., Banerjee A., Kumar V. Anomaly detection: A Survey // ACM Computing Surveys. 2009. Vol. 41, no. 3. Article 15. DOI: 10.1145/1541880.1541882

4. Breunig M., Kriegel H.-P., T. Ng R., Sander J. LOF: Identifying Density-Based Local Outliers // Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM Press. P. 93-104.
5. Jupyter Notebook [Electronic resource] https://jupyter.org/ (Date of access: 25.11.2020).
6. Google Colaboratory [Electronic resource] https://colab.research.google.com/ (Date of access: 25.11.2020).