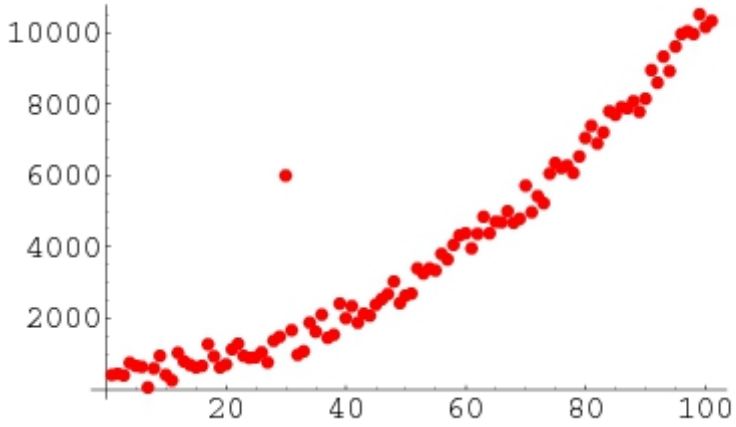
Что такое аномалия/выброс?



В статистике выбросы — это точки данных, которые не принадлежат определенной совокупности. Это ненормальное наблюдение, лежащее далеко от других значений. Выброс — это наблюдение, которое расходится с хорошо структурированными данными.

Например, вы можете ясно увидеть выброс в этом списке: [20,24,22,19,29,18,4300,30,18]

Почему мы заботимся об аномалиях?

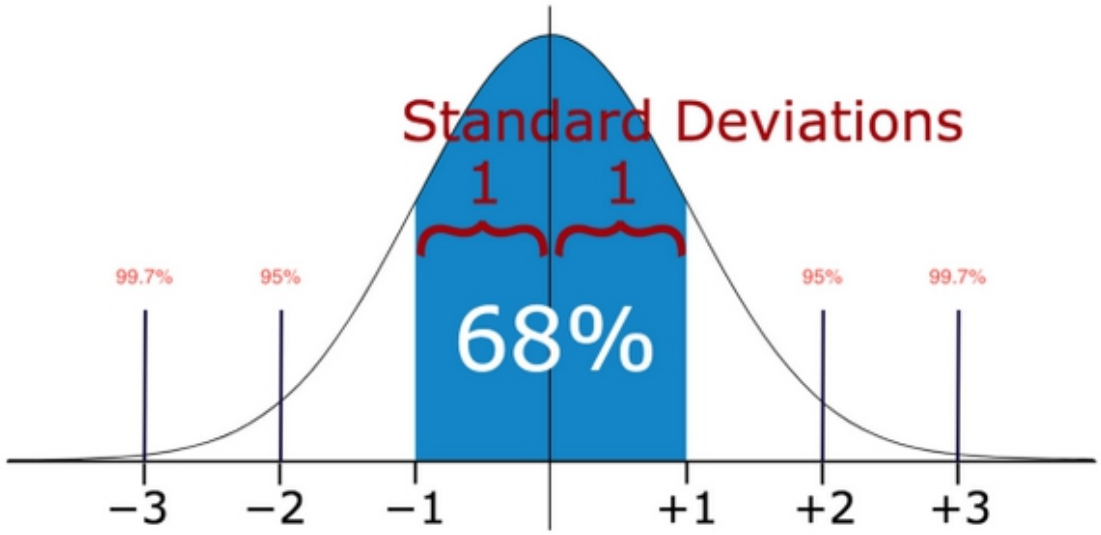
Обнаружение выбросов или аномалий — одна из основных проблем интеллектуального анализа данных. Возникающее расширение и продолжающийся рост данных, а также распространение устройств IoT заставляют нас переосмыслить подход к аномалиям и варианты использования, которые можно построить, изучая эти аномалии.

Теперь у нас есть смарт-часы и браслеты, которые могут определять сердцебиение каждые несколько минут. Обнаружение аномалий в данных сердцебиения может помочь в прогнозировании сердечных заболеваний. Аномалии в схемах движения могут помочь в прогнозировании аварий. Его также можно использовать для выявления узких мест в сетевой инфраструктуре и трафика между серверами. Следовательно, варианты использования и решения, основанные на обнаружении аномалий, безграничны.

Еще одна причина, по которой нам необходимо выявлять аномалии, заключается в том, что при подготовке наборов данных для моделей машинного обучения очень важно обнаружить все выбросы и либо избавиться от них, либо проанализировать их, чтобы понять, почему они у вас есть.

Метод 1 — Стандартное отклонение:

В статистике, если распределение данных приблизительно нормальное, то около 68% значений данных находятся в пределах одного стандартного отклонения от среднего, около 95% находятся в пределах двух стандартных отклонений и около 99,7% лежат в пределах трех стандартных отклонений.

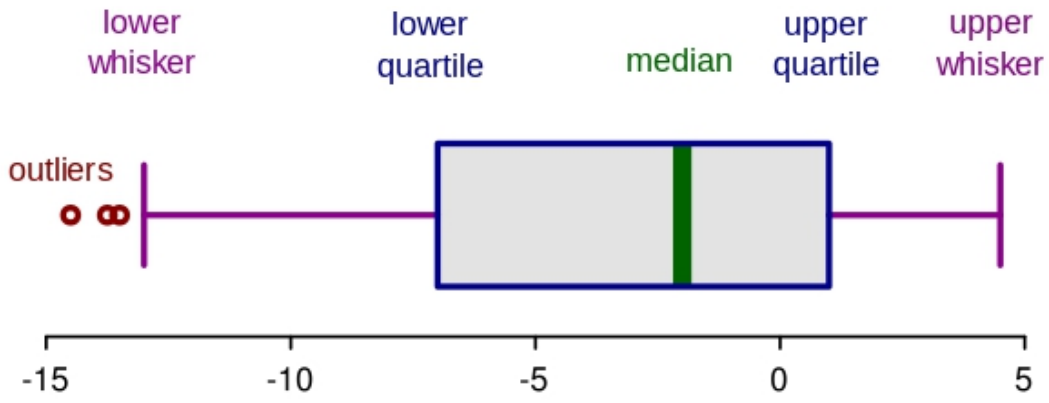


Следовательно, если у вас есть какая-либо точка данных, которая более чем в 3 раза превышает стандартное отклонение, то эти точки, скорее всего, будут аномальными или выбросами.

Давайте посмотрим код.

```python
import numpy as np
import matplotlib.pyplot as plt
seed(1)


# multiply and add by random numbers to get some real values
data = np.random.randn(50000)  * 20 + 20

# Function to Detection Outlier on one-dimentional datasets.
def find_anomalies(data):
    #define a list to accumlate anomalies
    anomalies = []

    # Set upper and lower limit to 3 standard deviation
    random_data_std = std(random_data)
    random_data_mean = mean(random_data)
    anomaly_cut_off = random_data_std * 3

    lower_limit  = random_data_mean - anomaly_cut_off
    upper_limit = random_data_mean + anomaly_cut_off
    print(lower_limit)
    # Generate outliers
    for outlier in random_data:
        if outlier > upper_limit or outlier < lower_limit:
            anomalies.append(outlier)
    return anomalies

find_anomalies(data)
```

Метод 2 — Boxplots



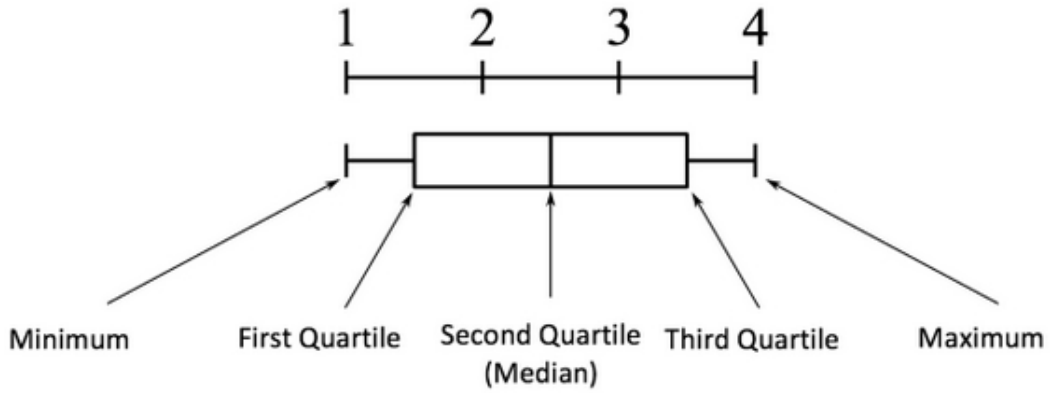
Блочные диаграммы представляют собой графическое изображение числовых данных через их квантили. Это очень простой, но эффективный способ визуализации выбросов. Думайте о нижних и верхних усах как о границах распределения данных. Любые точки данных, которые отображаются выше или ниже усов, могут считаться выбросами или аномальными. Вот код для построения блочной диаграммы:

```
1  import seaborn as sns
2  import matplotlib.pyplot as plt
3
4  sns.boxplot(data=random_data)
```

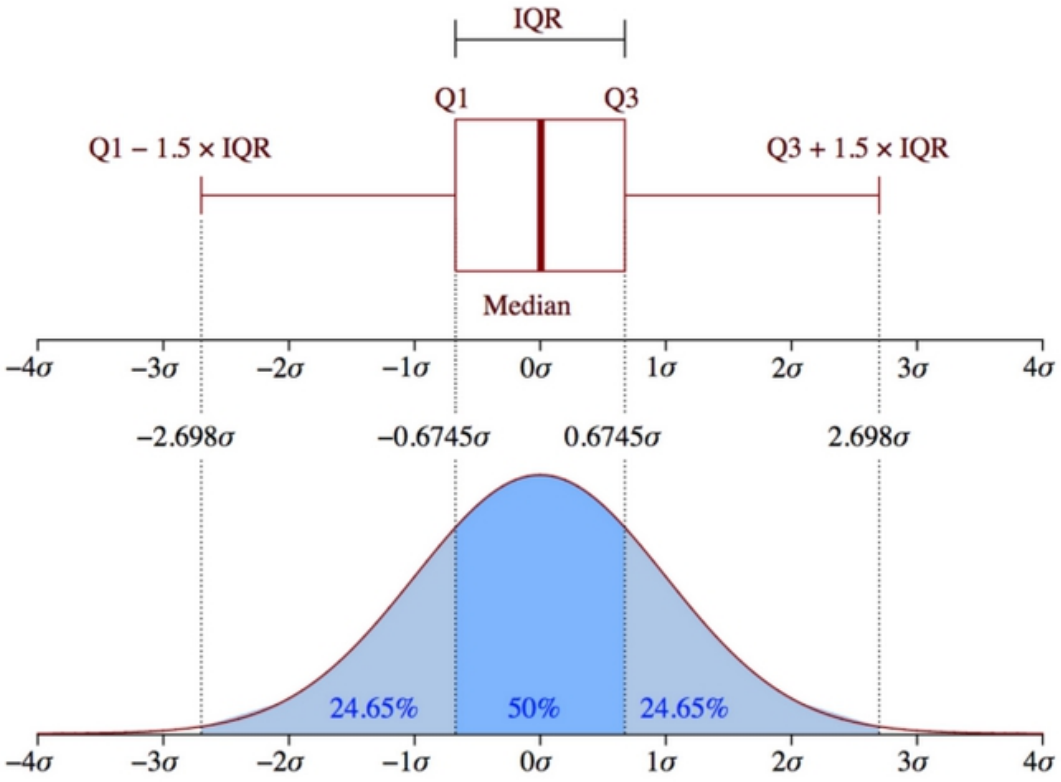
Анатомия блочной диаграммы:

Концепция межквартильного диапазона (IQR) используется для построения блочных диаграмм. IQR — это понятие в статистике, которое используется для измерения статистической дисперсии и изменчивости данных путем деления набора данных на квартили.

Проще говоря, любой набор данных или любой набор наблюдений делится на четыре определенных интервала на основе значений данных и того, как они соотносятся со всем набором данных. Квартиль — это то, что делит данные на три точки и четыре интервала.



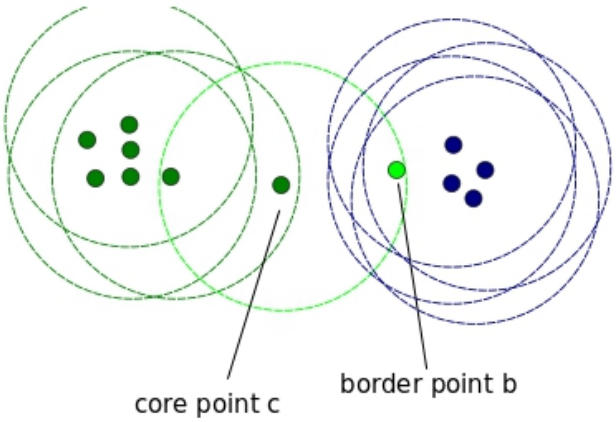
Межквартильный диапазон (IQR) важен, потому что он используется для определения выбросов. Это разница между третьим квартилем и первым квартилем (IQR = Q3 -Q1). Выбросы в этом случае определяются как наблюдения, которые находятся ниже (Q1 - 1,5x IQR) или ниже или выше ящичковой диаграммы (Q3 + 1,5x IQR) или в ящичковой диаграмме выше усов.



Метод 3 — Кластеризация DBScan:

DBScan — это алгоритм кластеризации, который использует кластерные данные в группы. Он также используется в качестве метода обнаружения аномалий на основе плотности с одномерными или многомерными данными. Другие алгоритмы кластеризации, такие как k-means и иерархическая кластеризация, также могут использоваться для обнаружения выбросов. DBScan имеет три важные концепции:

1. Основные точки: чтобы понять концепцию основных точек, нам нужно посетить некоторые гиперпараметры, используемые для определения задания DBScan. Первый гиперпараметр (HP) — min_samples. Это просто минимальное количество основных точек, необходимое для формирования кластера. второй важный HP это eps. eps — максимальное расстояние между двумя выборками, при котором они считаются принадлежащими одному кластеру.
2. Пограничные точки находятся в том же кластере, что и основные точки, но намного дальше от центра кластера.



3. Все остальное называется Noise Points, это точки данных, которые не принадлежат ни одному кластеру. Они могут быть аномальными или неаномальными и требуют дальнейшего изучения. Теперь давайте посмотрим на код.

```
from sklearn.cluster import DBSCAN
seed(1)
random_data = np.random.randn(50000,2)  * 20 + 20

outlier_detection = DBSCAN(min_samples = 2, eps = 3)
clusters = outlier_detection.fit_predict(random_data)
list(clusters).count(-1)
```

Результат приведенного выше кода — 94. Это общее количество зашумленных точек. SKLearn помечает зашумленные точки как (-1). Недостатком этого метода является то, что чем выше размерность, тем менее точным он становится. Вам также необходимо сделать несколько предположений, таких как оценка правильного значения для eps, что может быть непросто.

Метод 4 — Изолирующий лес:

Изолирующий лес — это алгоритм обучения без учителя, принадлежащий к семейству деревьев решений ансамбля. Этот подход отличается от всех предыдущих методов. Все предыдущие пытались найти нормальную область данных, а затем идентифицировали все, что находится за пределами этой определенной области, как выброс или аномальное.

Этот метод работает иначе. Он явно изолирует аномалии вместо профилирования и построения нормальных точек и областей, присваивая оценку каждой точке данных. Он использует тот факт, что аномалии являются точками данных меньшинства и что их значения атрибутов сильно отличаются от значений обычных экземпляров. Этот алгоритм отлично работает с наборами данных очень большой размерности и оказался очень эффективным способом обнаружения аномалий.

Теперь давайте изучим код:

```
from sklearn.ensemble import IsolationForest
import numpy as np
np.random.seed(1)
random_data = np.random.randn(50000,2)  * 20 + 20

clf = IsolationForest( behaviour = 'new', max_samples=100, random_state = 1,
preds = clf.fit_predict(random_data)
preds
```

Этот код будет выводить прогнозы для каждой точки данных в массиве. Если результат равен -1, это означает, что эта конкретная точка данных является выбросом. Если результат равен 1, то это означает, что точка данных не является выбросом.

Метод 5 — Надежный случайный вырезанный лес:

Алгоритм Random Cut Forest (RCF) — это неконтролируемый алгоритм Amazon для обнаружения аномалий. Он также работает, связывая оценку аномалии. Низкие значения баллов указывают на то, что точка данных считается «нормальной». Высокие значения указывают на наличие аномалии в данных. Определения «низкий» и «высокий» зависят от приложения, но обычная практика предполагает, что баллы, превышающие три стандартных отклонения от среднего балла, считаются аномальными.

Самое замечательное в этом алгоритме то, что он работает с данными очень большой размерности. Он также может работать с потоковыми данными в реальном времени (встроенными в AWS Kinesis Analytics), а также с автономными данными.