Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»
Физтех-школа Прикладной Математики и Информатики
Кафедра корпоративных информационных систем

**Направление подготовки / специальность:** 03.03.01 Прикладные математика и физика (бакалавриат)
**Направленность (профиль) подготовки:** Математика, информатика, управление и программная инженерия

# РАЗРАБОТКА МЕХАНИЗМА ОПРЕДЕЛЕНИЯ АНОМАЛИЙ В ДАННЫХ

(бакалаврская работа)

**Студент:**
Каразеев Антон Андреевич

_____
*(подпись студента)*

**Научный руководитель:**
Дайняк Александр Борисович,
канд. физ.-мат. наук, доц.

_____
*(подпись научного руководителя)*

**Консультант** *(при наличии)***:**
_____
_____
*(подпись консультанта)*

Москва 2019

**Оглавление**

Стр.

**Введение** . . . . . 4

**Глава 1. Анализ предметной области** . . . . . 5
1.1 Постановка задачи . . . . . 5
1.2 Объект, предмет и методы исследования . . . . . 6
1.3 Актуальность выбранной темы . . . . . 7

**Глава 2. Определение аномалий** . . . . . 8
2.1 Основы алгоритмов . . . . . 8
2.1.1 Анализ . . . . . 8
2.1.2 Результат работы алгоритмов . . . . . 8
2.2 Классификация методов определения аномалий . . . . . 10
2.2.1 Метрика качества модели . . . . . 11
2.3 Подходы к решению . . . . . 12
2.3.1 Параметрический подход . . . . . 12
2.3.2 Непараметрический подход . . . . . 13
2.3.3 Восстановление смесей . . . . . 14

**Глава 3. Описание экспериментов и анализ** . . . . . 16
3.1 Рассмотренные алгоритмы . . . . . 16
3.1.1 k-Nearest Neighbors (k-NN) . . . . . 16
3.1.2 Principal Component Analysis (PCA) . . . . . 16
3.1.3 One-Class Support Vector Machines (OCSVM) . . . . . 18
3.1.4 Local Outlier Factor (LOF) . . . . . 18
3.1.5 Histogram-Based Outlier Score (HBOS) . . . . . 19
3.1.6 Isolation Forest (IFOREST) . . . . . 20
3.2 Наборы данных . . . . . 21
3.3 Визуализация данных . . . . . 22
3.4 Эксперименты . . . . . 24

**Глава 4. Сервис для анализа данных** . . . . . 27

Стр.

4.1 Описание сервиса . . . . . 27
4.2 Основные функции . . . . . 29

**Глава 5. Результаты** . . . . . 31
5.1 Выводы . . . . . 31

**Словарь терминов** . . . . . 32

**Список литературы** . . . . . 33

**Список рисунков** . . . . . 35

**Список таблиц** . . . . . 37

# Введение

В бакалаврской работе рассматриваются алгоритмы и методы определения аномалий в данных. Рассмотрена основная метрика оценки качества подобных алгоритмов – ROC-AUC. Также приведены наборы данных, на которых проверялось качество обучения алгоритмов.

Приводится возможная реализация сервиса, который объединяет в себе современные методы по определению аномалий в данных.

Работа состоит из введения и пяти глав. Полный объём работы составляет 37 страниц, включая 18 рисунков и 2 таблицы. Список литературы содержит 15 наименований.

# Глава 1. Анализ предметной области

## 1.1 Постановка задачи

В настоящее время существует множество подходов к определению аномалий в данных. Все они хорошо применимы в своей области. Именно поэтому следует в первую очередь проанализировать существующие алгоритмы и их сферы применимости. После чего предлагается объединить их в единый сервис, чтобы упростить жизнь потенциальному пользователю, которому необходимо будет проанализировать большой объём данных.

Определение аномалий это процесс поиска объектов в данных, которые отличаются от нормальных объектов. В сетевой безопасности под поиском аномалий понимают обнаружение вторжения, в области криминалистики – обнаружение мошенничества. Другие области тоже имеют свои аномалии – обнаружение мошенничества с платежами, анализ транзакций по кредитным картам, обнаружение бизнес-преступлений, анализ финансовых данных транзакций. Кроме того, обнаружение аномалий применимо в медицинской сфере путём мониторинга жизненно важных функций пациентов, а также в космической отрасли – обнаружение сбоев во время запуска ракеты.

## 1.2 Объект, предмет и методы исследования

**Объектом исследования** являются методы для поиска аномалий в данных, а также способы их реализации и области применения.

**Предметом исследования** является анализ существующих алгоритмов для обработки данных.

**Методы исследования** включают в себя анализ предметной области, анализ реализованных методов, написание программного кода и извлечение полезной информации из данных.

## 1.3 Актуальность выбранной темы

Объём данных, которые появляются каждый день, растёт с экспоненциальной скоростью. Необходимо уметь работать с большими объёмами данных, чтобы получать из этого пользу. А для этого необходимо использовать актуальные алгоритмы и подходы, которые уже имеются.

Реализация современных алгоритмов поиска аномалий в данных может помочь в разных областях, таких как выявление отклонений в здоровье пациента, мошенничества в банковской сфере и другие. Спрос на подобные сервисы в настоящее время высок и растёт с каждым днём.

# Глава 2. Определение аномалий

## 2.1 Основы алгоритмов

Задачу поиска аномалий можно отнести к классу задач обучения без учителя. Суть поиска аномалий заключается в том, чтобы найти в выборке объекты, которые не похожи на большинство объектов выборки, то есть те, которые выделяются на фоне других.

Часто бывает так, что аномальных объектов либо нет вообще, либо их очень мало и неизвестно где именно в выборке они находятся. Поэтому обычно поиск аномалий относится к классу задач обучения без учителя (так как отсутствуют размеченные данные).

### 2.1.1 Анализ

В качестве источника информации используются статьи [1—6]. Помимо статей интерес представляют наборы данных, которые тоже предстояло найти. Среди наборов данных есть хорошо известный MNIST с набором изображений рукописных цифр, а также данные касательно разных типах стёкол, о заболеваниях сердца, о свободных электронах в ионосфере Земли и другие.

В этой работе рассматриваются стационарные данные. Поиск аномалий во временных рядах и прогнозирование временных рядов находятся за рамками рассматриваемой в работе темы.

### 2.1.2 Результат работы алгоритмов

Результатом работы алгоритма для поиска аномалий могут быть как **степени аномалии** (anomaly scores), так и **бинарные метки** (binary labels).

В случае, когда алгоритм выдаёт **степень аномалии**, под степенью понимается уровень вероятности того, что объект является аномалией.

В случае **бинарных меток** алгоритм сразу указывает на нормальные (обычно обозначаемые как 0) и аномальные (обозначаемые как 1) данные. Несмотря на то, что некоторые алгоритмы детектирования аномалий возвращают бинарные метки напрямую, степени аномалий тоже могут быть переведены в бинарное представление. 0 или 1 содержат меньше информации, чем степень аномалии. Тем не менее, это конечный результат, по которому обычно принимается решение об аномальности объекта выборки.

## 2.2 Классификация методов определения аномалий

Большинство методов определения аномалий используют метки, по которым можно определить, является ли объект выборки нормальным или аномальным. Поиск или сбор размеченных данных, которые будут точными и хорошо описывать рассматриваемую проблему, чтобы хорошо обучить алгоритмы, довольно сложно и дорого.

Обычно выделяют три типа методов поиска аномалий:

1. **Supervised методы (обучение с учителем)**
   Предполагается, что имеется доступ к обучающим данным с точными и репрезентативными метками для нормальных и аномальных объектов. В таком случае обычно разрабатывают предсказательную модель для обоих классов. После обучения на тренировочных данных к каждому объекту из тестовой выборки применяется алгоритм, чтобы определить класс объекта. Проблема такого подхода заключается в том, что получить точные и репрезентативные метки, особенно для аномалий, сложно. Такая ситуация довольно распространена в таких областях, как обнаружение мошенников в банковском секторе (сложно отличить мошенника от обычного пользователя только по действиям).
2. **Semi-supervised методы (обучение с частичным привлечение учителя)**
   Предполагается, что имеются размеченные данные только для нормального класса. Так как для обучения таких алгоритмов не требуются размеченные аномальные данные, они имеют более широкое применение, чем supervised методы.
3. **Unsupervised методы (обучение без учителя)**
   Такие методы не требуют обучающих данных и поэтому наиболее широко используются. Unsupervised методы поиска аномалий могут нормальные данные из всех представленных и рассматривать отклонение от них как аномалию.

Многие semi-supervised методы могут быть использованы для unsupervised случая. Например, с их помощью можно дополнительно семплировать объекты из выборки, если данных для обучения алгоритма попросту недостаточно.

### 2.2.1 Метрика качества модели

В качестве основной метрики используется ROC-AUC, что расшифровывается как ”receiver operating characteristic – area under curve”и переводится как ”рабочая характеристика приёмника – площадь под кривой”. ROC-кривая позволяет оценить качество бинарной классификации[1]. Она показывает соотношение между долей объектов, которые были верно классифицированы как принадлежащие определённому классу (True Positive Rate, TPR), и долей объектов от общего количества объектов, которые этому классу не принадлежат, ошибочно классифицированных как принадлежащие этому классу (False Positive Rate, FPR).

Количественную интерпретацию ROC даёт показатель AUC – площадь, ограниченная ROC-кривой и осью доли ложных положительных классификаций (False Positive Rate). Чем выше показатель AUC, тем качественнее классификатор, при этом значение 0.5 демонстрирует непригодность выбранного метода классификации, что эквивалентно случайному выбору. Максимальное значение AUC составляет 1.0.

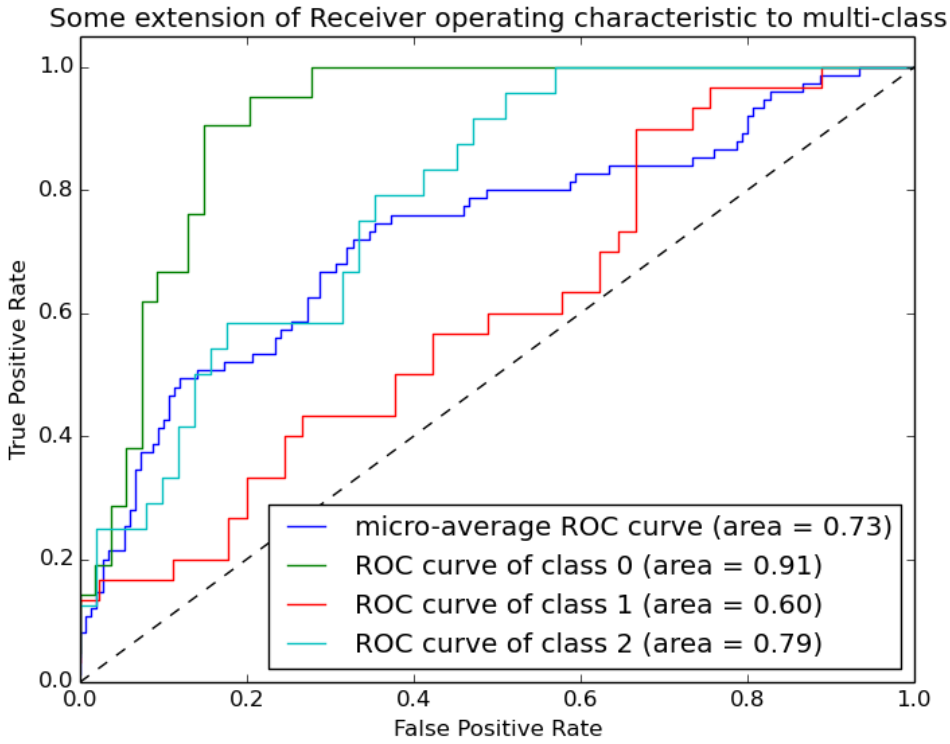


Рисунок 2.1 — Пример ROC-кривой.

График на рисунке 2.1 был построен с помощью библиотеки scikit-learn[2].

[1] бинарная означает, что есть лишь два класса объектов – например, нормальные и аномальные

[2] https://scikit-learn.org/0.15/auto_examples/plot_roc.html

## 2.3 Подходы к решению

Одним из возможных способов определения аномалий является измерение схожести между объектов. У такого способа есть два варианта:

1. Восстановление плотности
2. Классификация

В случае с восстановлением плотности необходимо построить распределение, которое хорошо описывает выборку. И это распределение позволяет посчитать вероятность для нового объекта получить его из распределения, описывающего выборку.

В терминах этого метода аномалия - объект, полученный из другого распределения, описывающего другую выборку данных.

Есть три подхода:

1. Параметрический
2. Непараметрический
3. Восстановление смесей

### 2.3.1 Параметрический подход

Распределение представляется в виде $p(x) = \varphi(x|\theta)$, где $\theta$ выступает в качестве параметра распределения. Например, в семейство параметрических распределений входит распределение Гаусса – $\theta = (\mu, \Sigma)$, где $\mu$ – вектор средних и $\Sigma$ – ковариационная матрица.

Параметры модели подбираются таким образом, чтобы вероятность объектов из обучающей выборки была максимальной. Для этого обычно пользуются Методом Максимального Правдоподобия (Maximum Likelihood Estimation, MLE):

$$\sum_i \log \varphi(x_i|\theta) \to \max_\theta .$$

В случае нормального распределения формулы для параметров распределения будут иметь следующий вид:

$$\mu = \frac{1}{N} \sum_i x_i,$$

$$\Sigma = \frac{1}{N} \sum_i (x_i - \mu)(x_i - \mu)^T.$$

Тогда алгоритм для определения аномалий будет выглядеть так:

1. Получаем новый объект $x$ из выборки
2. Если $p(x) < t$, то полагаем, что объект $x$ является аномалией
3. Порог принятия решения $t$ выбирается из априорных соображений

### 2.3.2 Непараметрический подход

Для восстановления вида распределения вероятности по данным используется формула оценки Парзена-Розенблатта:

$$p_h(x) = \frac{1}{lh} \sum_{i=1}^{l} K\left(\frac{x - x_i}{h}\right),$$

где $h$ – ширина окна, $K$ – ядровая функция (Kernel Function). Функция ядра $K$ должна удовлетворять таким требованиям, как $K(x) = K(-x) \forall x$ и $\int_{-\infty}^{+\infty} K(x)dx = 1$.

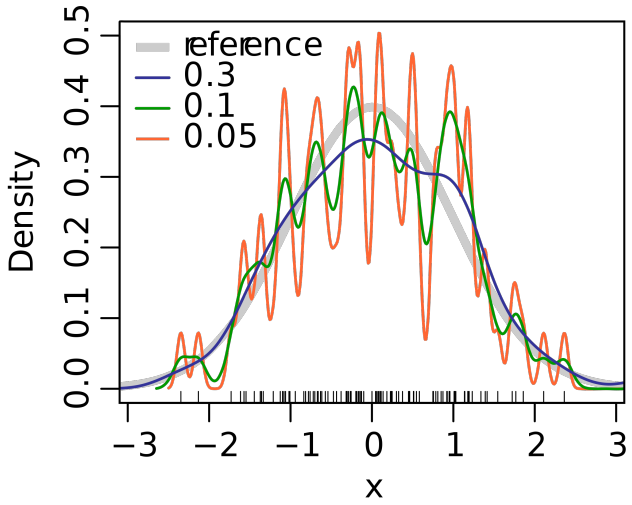


Рисунок 2.2 — Ядерная оценка плотности 100 нормально распределённых случайных чисел с использованием различных сглаживающие окон.

Примеры ядерных функций:

1. Равномерное: $K(u) = \frac{1}{2}, |u| \leqslant 1$
2. Треугольное: $K(u) = (1 - |u|), |u| \leqslant 1$
3. Гауссово: $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$

Хорошо обобщается на многомерный случай:

$$p_h(x) = \frac{1}{lV(h)} \sum_{i=1}^{l} K\left(\frac{\rho(x,x_i)}{h}\right),$$

где $\rho$ – заданная метрика (например, евклидово расстояние), $V(h) = \int K\left(\frac{\rho(x,x_i)}{h}\right) dx$. Чем выше размерность, тем больше объектов необходимо для лучшей точности алгоритма.

### 2.3.3 Восстановление смесей

В некоторых случаях параметрического подхода оказывается недостаточно. Например, распределение на рисунке 2.3 получено путём семплирования из трёх гауссовых распределений с равными стандартными отклонениями, но разными центрами. Такую выборку можно описать моделью смеси распределений. Для восстановления такой плотности отлично подходит EM-алгоритм.

Введём некоторые обозначения:

$$p(x) = \sum_{j=1}^{K} w_j p_j(x)$$

– смесь распределений (взвешенная сумма), где $p_j(x)$ – компоненты смеси (обычно являющиеся параметрическими распределениями $p_j(x) = \varphi(x|\theta_j)$),

$$g_{ji} = p(j|x_i) = \frac{w_j p_j(x_i)}{p(x_i)}$$

– апостериорная вероятность того, что объект под номером $i$ принадлежит компоненте смеси под номером $j$.

Тогда EM-алгоритм будет выглядеть следующим образом:

1. На E-шаге рассчитывается апостериорная вероятность $g_{ji}$
2. На M-шаге рассчитываются новые веса $w_j = \frac{1}{N} \sum_{i=1}^{N} g_{ji}$ и обновляется оценка на параметры $\theta$: $\theta_j = \arg\max_\theta \sum_{i=1}^{N} g_{ji} \log \varphi(\theta|x)$

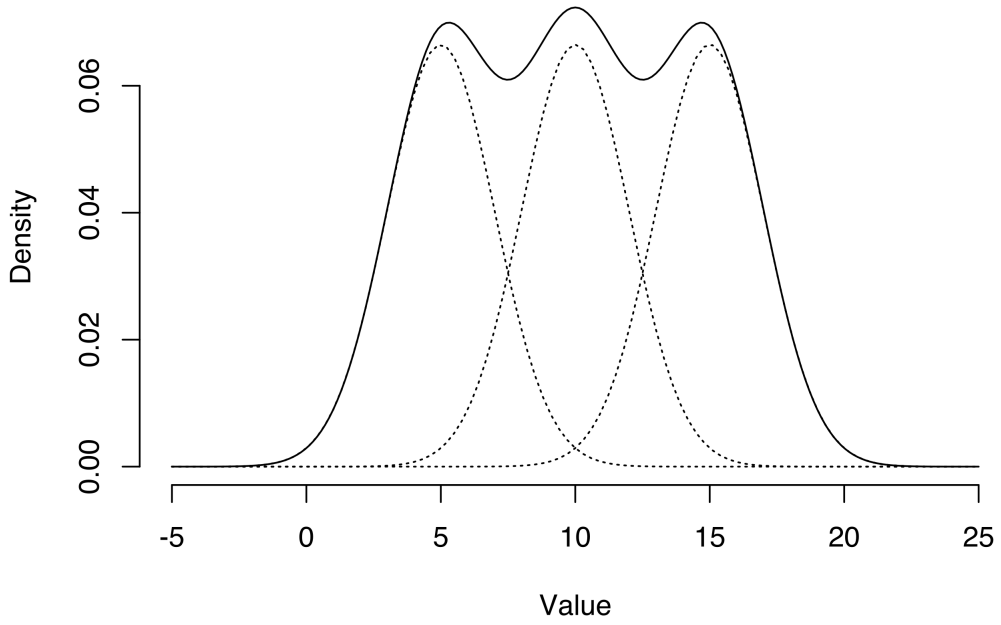


Рисунок 2.3 — Плотность распределения смеси, состоящей из трёх нормальных распределений ($\mu = [5, 10, 15]$, $\sigma = 2$) с одинаковыми весами.

Существуют ещё Автокодировщики, которые хорошо справляются с понижением размерности многомерных данных и с помощью которых можно хорошо семплировать из заданного распределения [7]. Но такой подход лежит за рамками этой бакалаврской работы.

# Глава 3. Описание экспериментов и анализ

## 3.1 Рассмотренные алгоритмы

В текущей работе были рассмотрены следующие алгоритмы:

1. k-Nearest Neighbors (k-NN) [8] – метод k-ближайших соседей.
2. Principal Component Analysis (PCA) [9] – метод главных компонент.
3. One-Class Support Vector Machines (OCSVM) [10] – одноклассовый метод опорных векторов.
4. Local Outlier Factor (LOF) [11] – метод локального уровеня выброса.
5. Histogram-Based Outlier Score (HBOS) [12] – оценка выбросов на основе гистограммы.
6. Isolation Forest (IFOREST) [13] – метод изолируещего леса.

### 3.1.1 k-Nearest Neighbors (k-NN)

Алгоритм для автоматической классификации объектов. Каждый объект выборки относится к тому классу, который является наиболее распространённым среди $k$ соседей рассматриваемого объекта, классы которых уже известны. Алгоритм может быть применим к многомерным наборам данных – в таком случае перед применением необходимо определить функцию расстояния (метрику). Классический вариантом такой функции является евклидово расстояние. На рисунке 3.1 приведён пример классификации методом k-ближайших соседей.

### 3.1.2 Principal Component Analysis (PCA)

Название алгоритма переводится как ”метод главных компонент”. Он применяется во многих областях, в том числе, биоинформатике, обработке изображений, для сжатия данных, в общественных науках. Вычисление этих главных

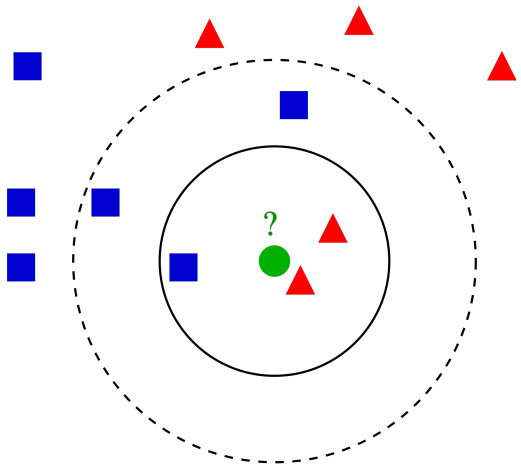


Рисунок 3.1 — Тестовый образец (зелёный круг) должен быть классифицирован как синий квадрат (класс 1) или как красный треугольник (класс 2). Если $k = 3$, то он классифицируется как класс 2, потому что внутри меньшего круга 2 треугольника и только 1 квадрат. Если $k = 5$, то он будет классифицирован как класс 1 (3 квадрата против 2 треугольников внутри большего круга).

компонент может быть сведено к вычислению сингулярного разложения матрицы данных или к вычислению собственных векторов и собственных значений ковариационной матрицы исходных данных. На рисунке 3.2 представлены собственные векторы в случае двумерной гауссианы.

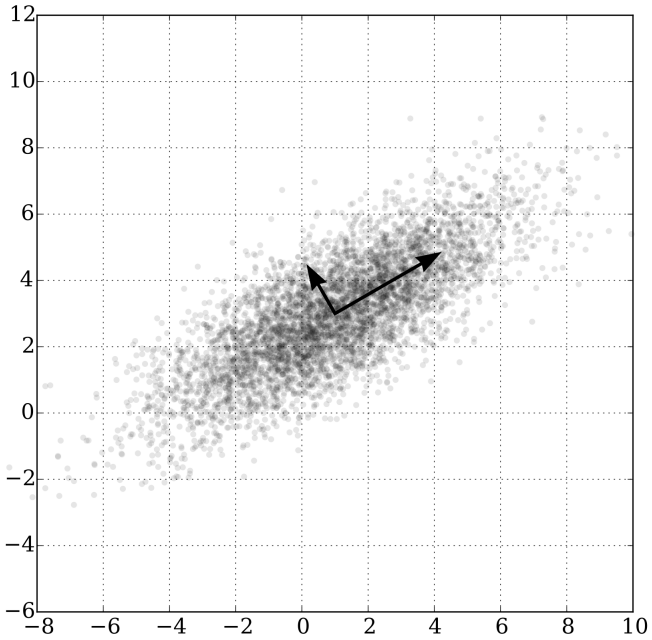


Рисунок 3.2 — PCA для многомерного гауссового распределения с центром в точке (1, 3) со стандартным отклонением 3. Векторы отражают собственные векторы ковариационной матрицы гауссианы.

### 3.1.3 One-Class Support Vector Machines (OCSVM)

В русской литературе алгоритм часто называют ”методом опорных векторов”. Он относится к семейству линейных классификаторов. Особым свойством метода опорных векторов является непрерывное уменьшение эмпирической ошибки классификации и увеличение зазора, поэтому метод также известен как метод классификатора с максимальным зазором.

Основная идея метода – перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Две параллельных гиперплоскости строятся по обеим сторонам гиперплоскости, разделяющей классы. Разделяющей гиперплоскостью будет гиперплоскость, максимизирующая расстояние до двух параллельных гиперплоскостей. На рисунке 3.3 продемонстрирована работа алгоритма.

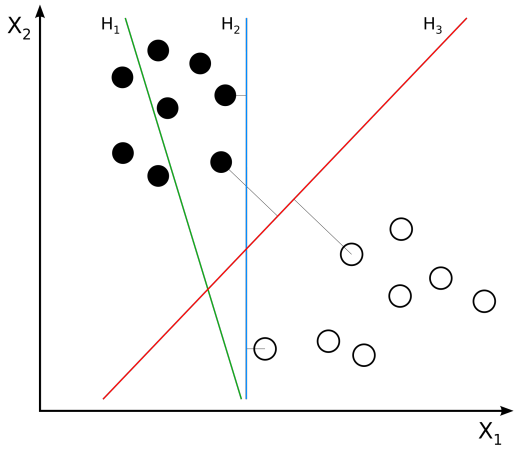


Рисунок 3.3 — Гиперплоскость $H1$ не разделяет классы. В случае гиперплоскости $H2$ разделение имеется, но зазор слишком маленький. Разделение с максимальным зазором достигается гиперплоскостью $H3$.

### 3.1.4 Local Outlier Factor (LOF)

Локальный уровень выброса основывается на концепции локальной плотности, где локальность задаётся $k$ ближайшими соседями, расстояния до которых используются для оценки плотности. Путём сравнения локальной плотности объекта с локальной плотностью его соседей, можно выделить области с аналогичной

плотностью и точки, которые имеют существенно меньшую плотность, чем её соседи. Эти точки считаются аномалиями.

Локальная плотность оценивается расстоянием, с которым точка может быть ”достигнута”от соседних точек. Определение ”расстояния достижимости используемого в алгоритме, является дополнительной мерой для получения более устойчивых результатов внутри кластеров. На рисунке 3.4 разобрана базовая идея алгоритма.

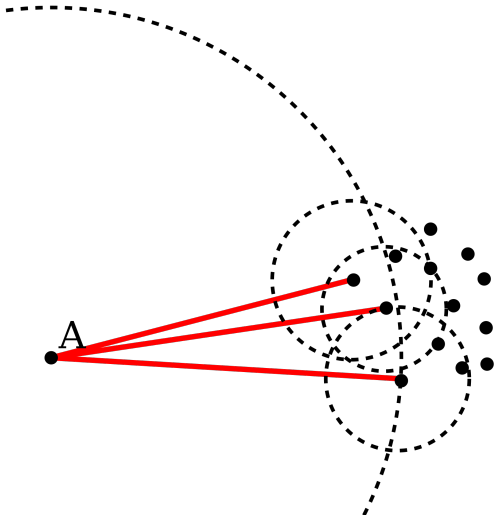


Рисунок 3.4 — Базовая идея метода – сравнение локальной плотности точки с плотностями её соседей. Точка A имеет меньшую плотность по сравнению с соседями

### 3.1.5 Histogram-Based Outlier Score (HBOS)

Алгоритм HBOS в несколько раз быстрее работает, чем алгоритмы, основанные на кластеризации и методе ближайших соседей. Для каждого измерения $d$ строится одномерная гистограмма, где высота каждой ячейки отражает оценку плотности. Затем проводится нормировка таким образом, что максимальная высота ячеек каждой гистограммы составляет $1.0$. Это обеспечивает равный вклад каждого измерения в оценку аномальности. Наконец, для каждого объекта $p$ выборки рассчитывается $HBOS$, используя высоту соответствующей ячейки, в которой объект расположен:

$$HBOS\left(p\right) = \sum_{i=0}^{d} \log\left(\frac{1}{hist_i\left(p\right)}\right).$$

Вместо произведения используется сумма логарифмов – это то же самое, что и применение логарифма к произведению $(\log\left(a \cdot b\right) = \log\left(a\right) + \log\left(b\right))$. Такой подход менее чувствительный к ошибкам, которые связаны с точностью плавающей точки в экстремально несбалансированных распределениях, что в свою очередь может приводить к очень высоким значениям оценки аномальности. Подробнее про алгоритм можно найти информацию в [12].

### 3.1.6 Isolation Forest (IFOREST)

Один из вариантов случайного леса (строится из решающих деревьев). Выбирается случайный признак и случайное разделение, по которым строится ветвление в дереве. Для каждого объекта выборки определяется мера его нормальности как среднее арифметическое глубин листьев, в которые он попал (под этим понимается ”изоляция”).

При таком способе построения деревьев аномалии будут попадать в листья на ранних этапах (на небольшой глубине дерева), то есть выбросы проще ”изолировать”. Дерево строится до тех пор, пока каждый объект не окажется в отдельном листе). На рисунке 3.5 продемонстрирована основная идея метода.

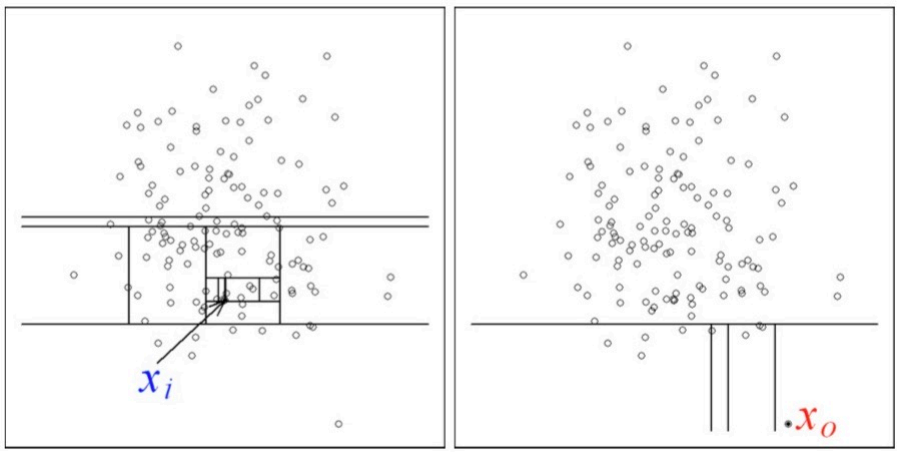


Рисунок 3.5 — Для изолирования точки $x_i$ требуется 12 случайных разбиений, а для аномальной точки $x_o$ – только 4 разбиения.

### 3.2 Наборы данных

Для проверки сервиса были рассмотрены следующие наборы данных:

1. Arrhythmia – определение наличия аритмии по данным ЭКГ [14].
2. Breast Cancer – определение типа опухоли молочной железы: доброкачественная или злокачественная.
3. Glass – идентификация типа стекла, оставленного на месте преступления.
4. Ionosphere – рассматриваются характеристики радаров, которые используется в анализе ионосферы: необходимо определить является радар ”плохим”или ”хорошим”.
5. Letter Recognition – по описанию изображения определить присутствует ли буква из английского алфавита или нет.
6. Mammography – детектирование микрокальцинатов по данным маммографии.
7. MNIST – научиться различать изображения рукописных цифр 6 и 0.
8. Satellite – определение типа почвы по спутниковым снимкам.

В качестве источника этих данных выступает библиотека ODDS [15].

Таблица 1 — Статистика по данным из рассматриваемых наборов данных.

| Датасет | Кол-во объектов | Размерность | Процент аномалий |
|---|---|---|---|
| arrhythmia | 452 | 274 | 14.60 |
| breastw | 683 | 9 | 34.99 |
| glass | 214 | 9 | 4.21 |
| ionosphere | 351 | 33 | 35.90 |
| letter | 1600 | 32 | 6.25 |
| mammography | 11183 | 6 | 2.33 |
| mnist | 7603 | 100 | 9.21 |
| satellite | 6435 | 36 | 31.64 |

В таблице 1 приведено сравнение наборов данных, которые использовались для оценки качества алгоритмов. Также был рассчитан процент аномалий для каждого из наборов.

## 3.3 Визуализация данных

Рассматриваемые наборы данных имеют высокую размерность, что означает невозможность представления данных на плоскости без применения каких-либо методов. Поэтому необходимо понизить размерность до двух, чтобы отобразить объекты каждого из наборов данных на плоскости – для этого использовался алгоритм понижения размерности t-SNE[1]. На рисунке 3.6 представлены данные после понижения размерности.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

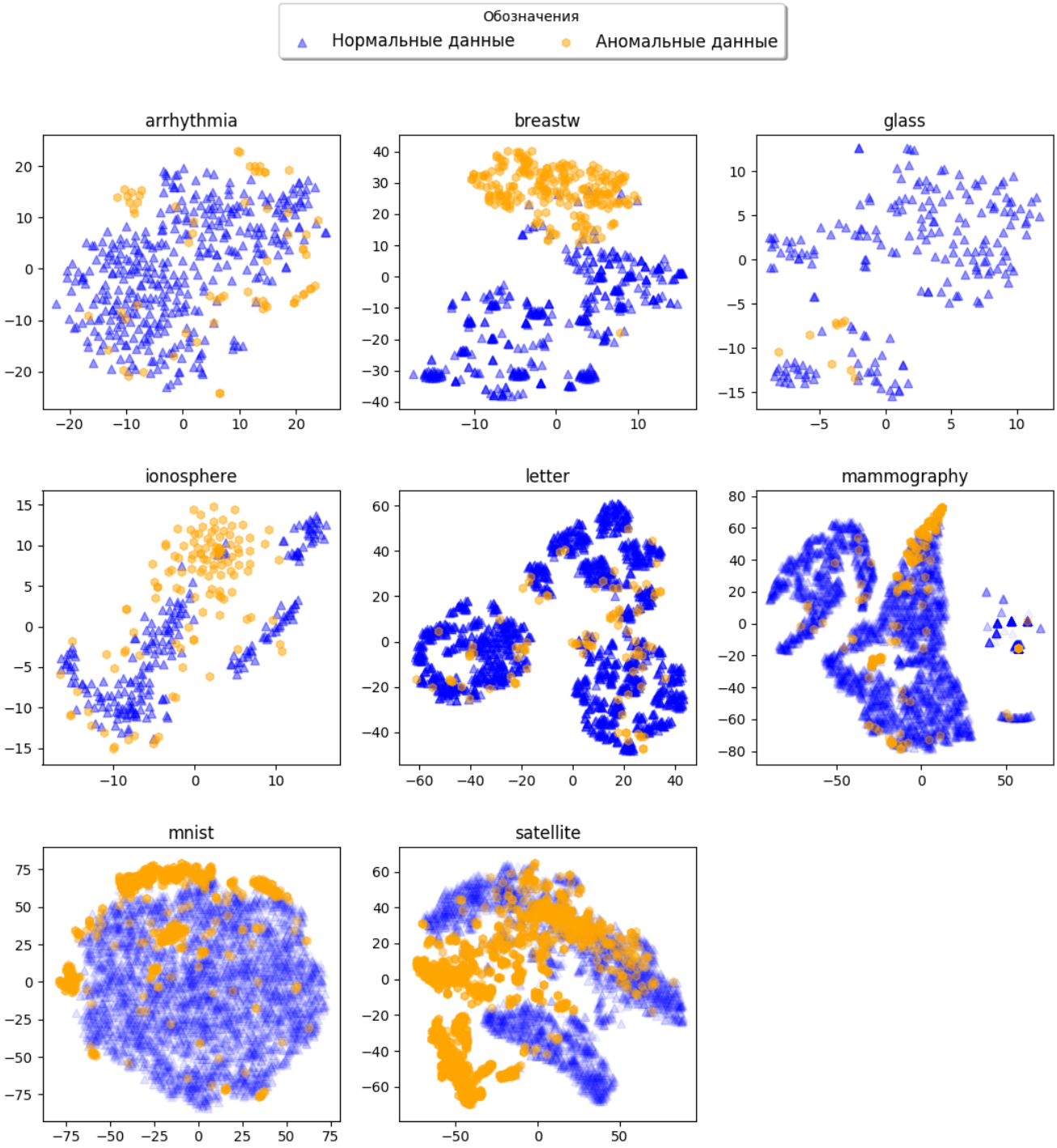


Рисунок 3.6 — Рассматриваемые наборы данных после применения алгоритма понижения размерности t-SNE.

## 3.4 Эксперименты

После подготовки данных на них были обучены и протестированы рассматриваемые алгоритмы. Набор данных разбивался на две непересекающиеся части: на первой части алгоритм обучался, а на второй – проверялось качество обученного алгоритма. Обычно, разделение происходит в соотношении 75% и 25% соответственно. Так снижается вероятность переобучения на данных, что позволяет избежать ухудшения обобщающей способности алгоритмов.

В таблице 2 представлены результаты экспериментов оценки качества рассматриваемых алгоритмов на каждом наборе данных. Жирным шрифтом выделены наилучшие алгоритмы для каждого набора данных, которые продемонстрировали самую высокую оценку по методу ROC-AUC.

Таблица 2 — Значения ROC для рассматриваемых алгоритмов на данных.

| Датасет | KNN | PCA | OCSVM | LOF | HBOS | IFOREST |
|---|---|---|---|---|---|---|
| arrhythmia | 0.7555 | 0.7794 | 0.7825 | 0.7672 | 0.7831 | **0.7849** |
| breastw | **0.9908** | 0.9608 | 0.9649 | 0.4574 | 0.9764 | 0.9872 |
| glass | **0.8558** | 0.7308 | 0.8077 | 0.6538 | 0.7500 | 0.7212 |
| ionosphere | **0.9460** | 0.8115 | 0.8684 | 0.9023 | 0.6190 | 0.8632 |
| letter | **0.8660** | 0.5119 | 0.5985 | 0.8530 | 0.5532 | 0.5770 |
| mammography | 0.8346 | **0.9039** | 0.8911 | 0.6806 | 0.8506 | 0.8680 |
| mnist | 0.8322 | **0.8493** | 0.8487 | 0.6727 | 0.5607 | 0.7942 |
| satellite | 0.6795 | 0.5601 | 0.6274 | 0.5567 | **0.7464** | 0.7008 |

На графиках, которые изображены на рисунке 3.7, показана эффективность работы рассматриваемых алгоритмов на каждом наборе данных. При построении графиков использовалась библиотека adjustText[2].

Согласно таблице 2 наилучшее качество показал алгоритм k-NN на наборе данных breastw (Breast Cancer). На рисунке 3.9) продемонстрирован результат работы обученного алгоритма на этом датасете после понижения размерности с помощью метода t-SNE.

[2] https://github.com/Phlya/adjustText/

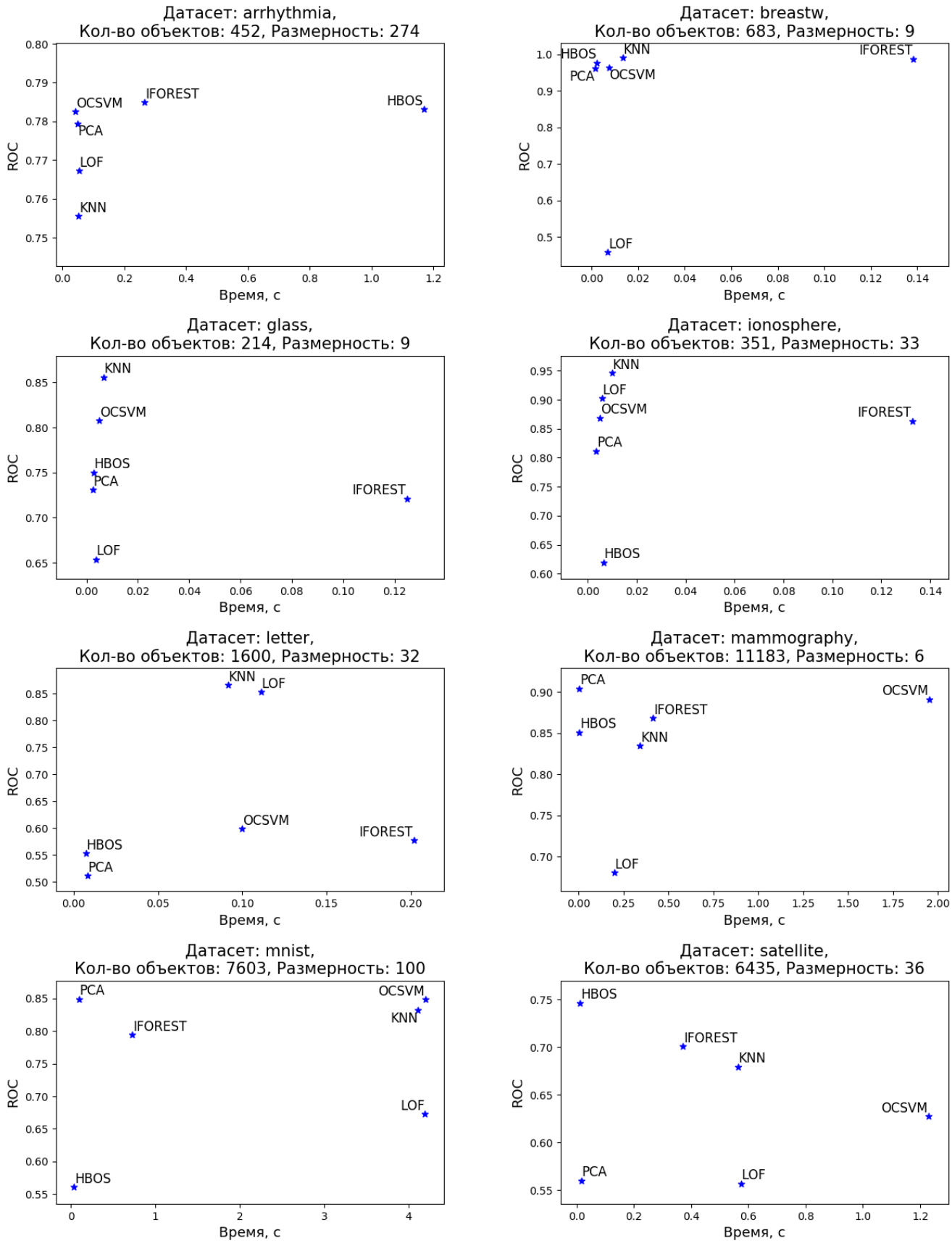


Рисунок 3.7 — Эффективность алгоритмов на разных наборах данных.

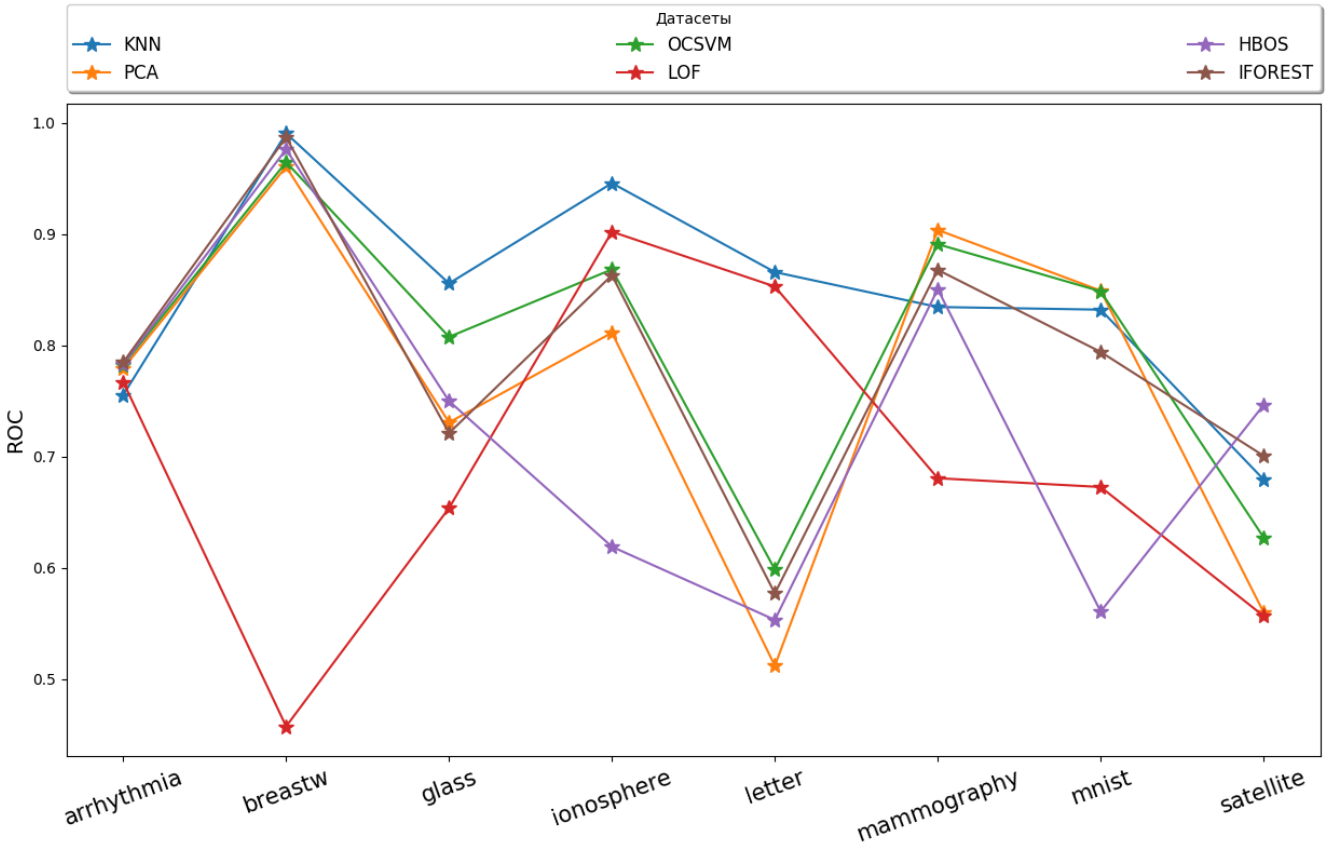


Рисунок 3.8 — Качество алгоритмов в зависимости от набора данных.

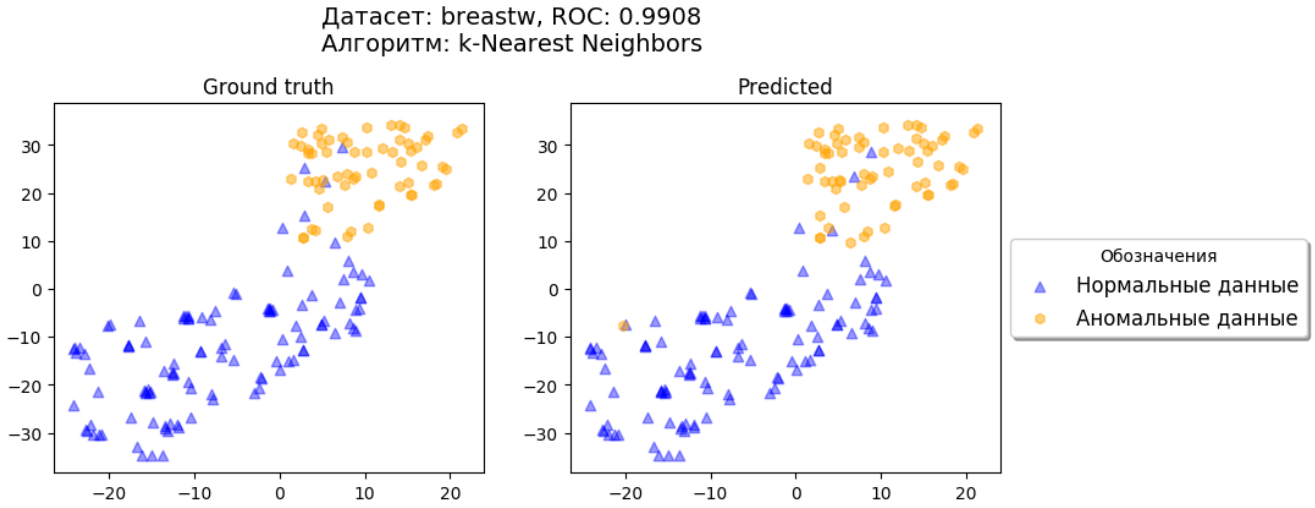


Рисунок 3.9 — Датасет Breast Cancer.

# Глава 4. Сервис для анализа данных

В этой главе будет рассмотрено устройство сервиса, которое было создано, чтобы упростить процесс анализа данных и поиска аномалий в них.

Ниже будут представлены основные сценарии работы с сервисом.

## 4.1 Описание сервиса

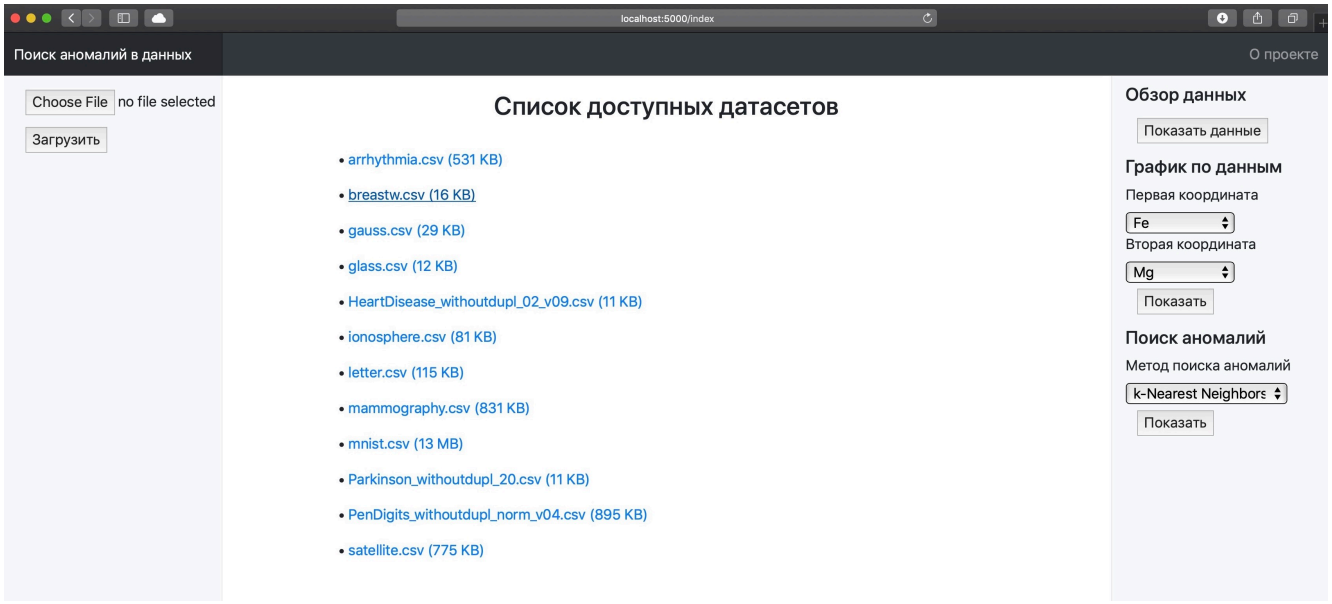


Рисунок 4.1 — Главный экран сервиса. Есть возможность загрузить данные через окно загрузки (как продемонстрировано рисунке 4.2), либо выбрать набор данных из списка загруженных ранее.

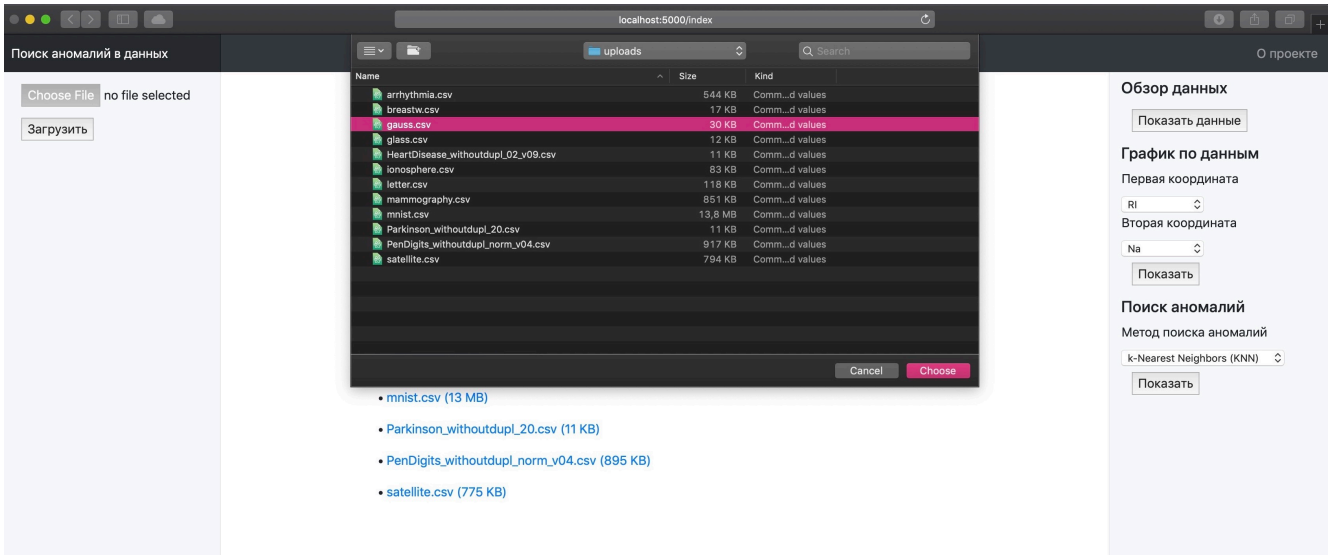


Рисунок 4.2 — Пример экрана загрузки набора данных на сервер.

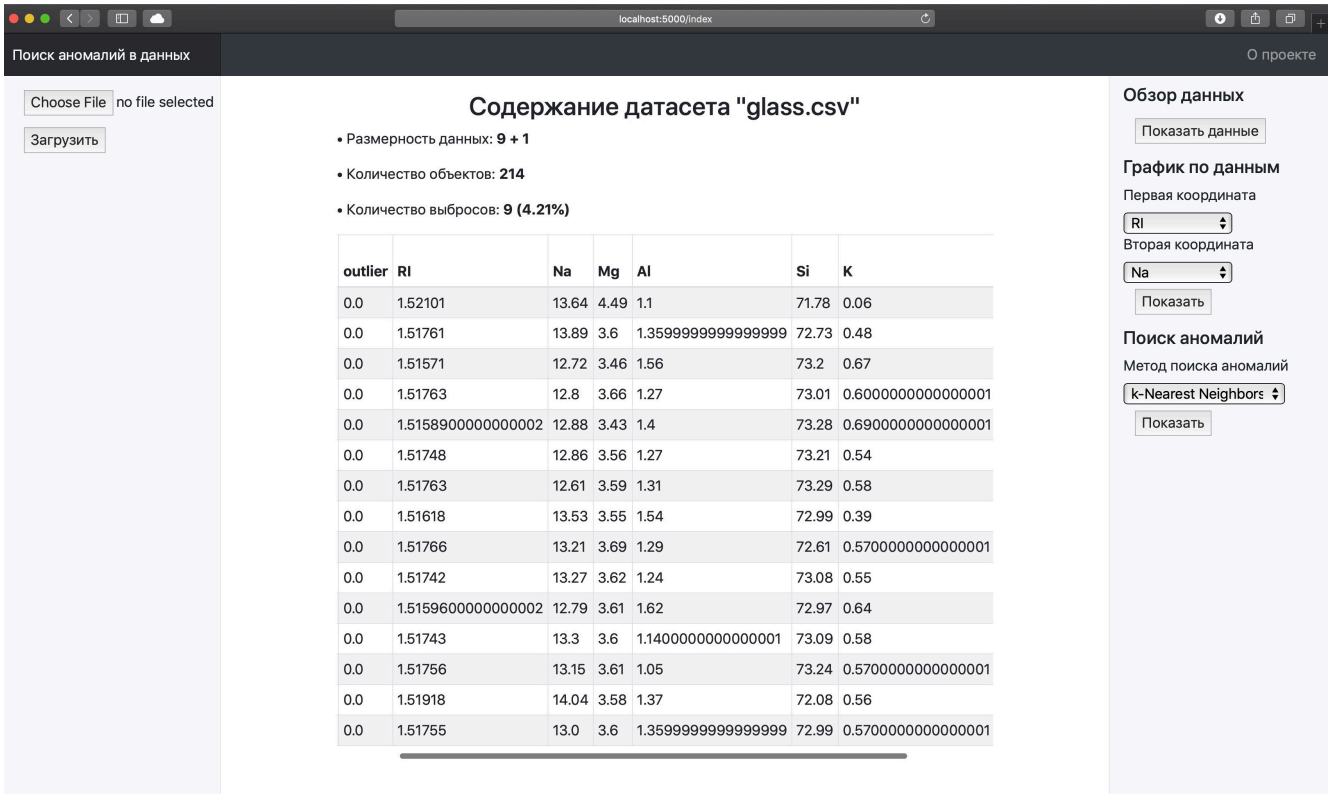


Рисунок 4.3 — После выбора файла на экране появляется таблица с некоторыми объектами из указанного набора данных. С помощью меню, которое располагается справа, можно выбрать: (**i**) построение общего графика по набору данных, на котором будут представлены данные, размерность которых была понижена с помощью метода t-SNE (продемонстрировано на рисунке 4.4), **(ii)** график зависимости одной из размерностей данных от другой (продемонстрировано на рисунке 4.5) и (**iii**) выбрать алгоритм из списка, с помощью которого будет осуществлён поиск аномалий в данных (продемонстрировано на рисунке 4.6).

## 4.2 Основные функции

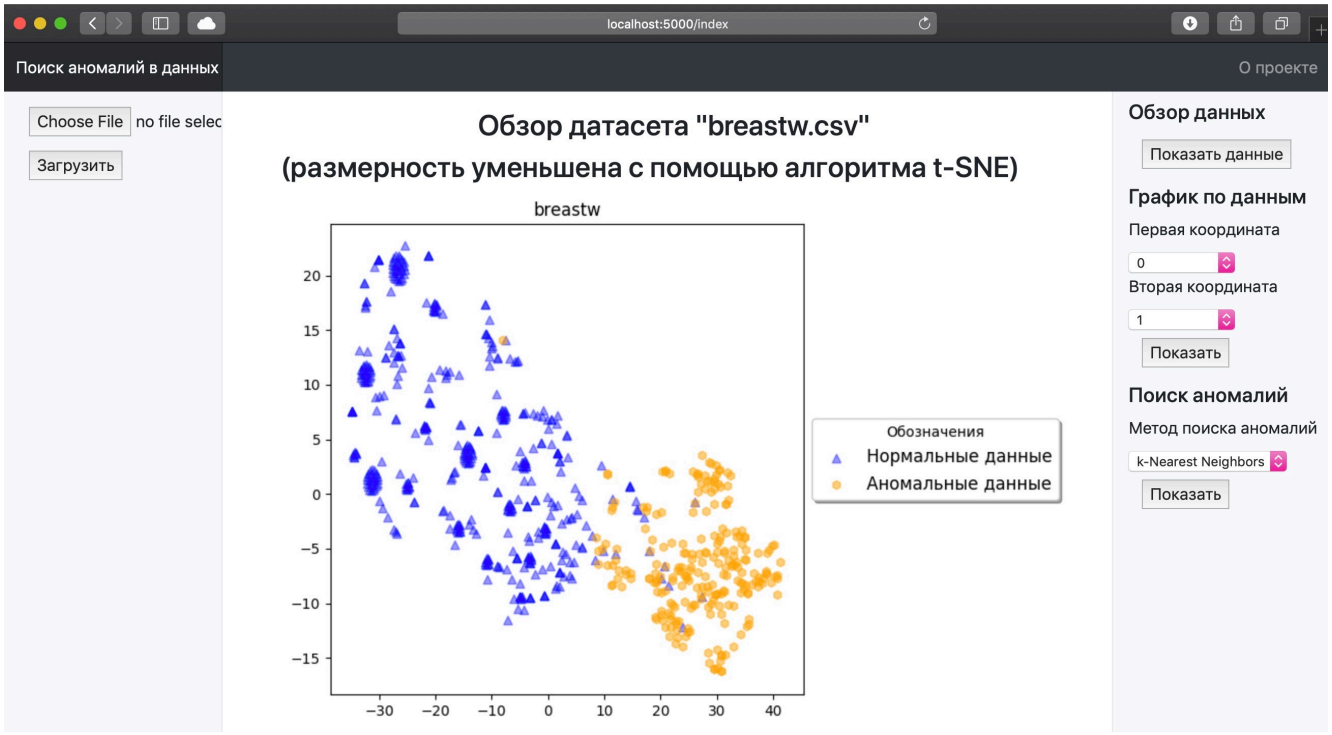


Рисунок 4.4 — Представление данных после понижения размерности до dim=2 при помощи алгоритма t-SNE.

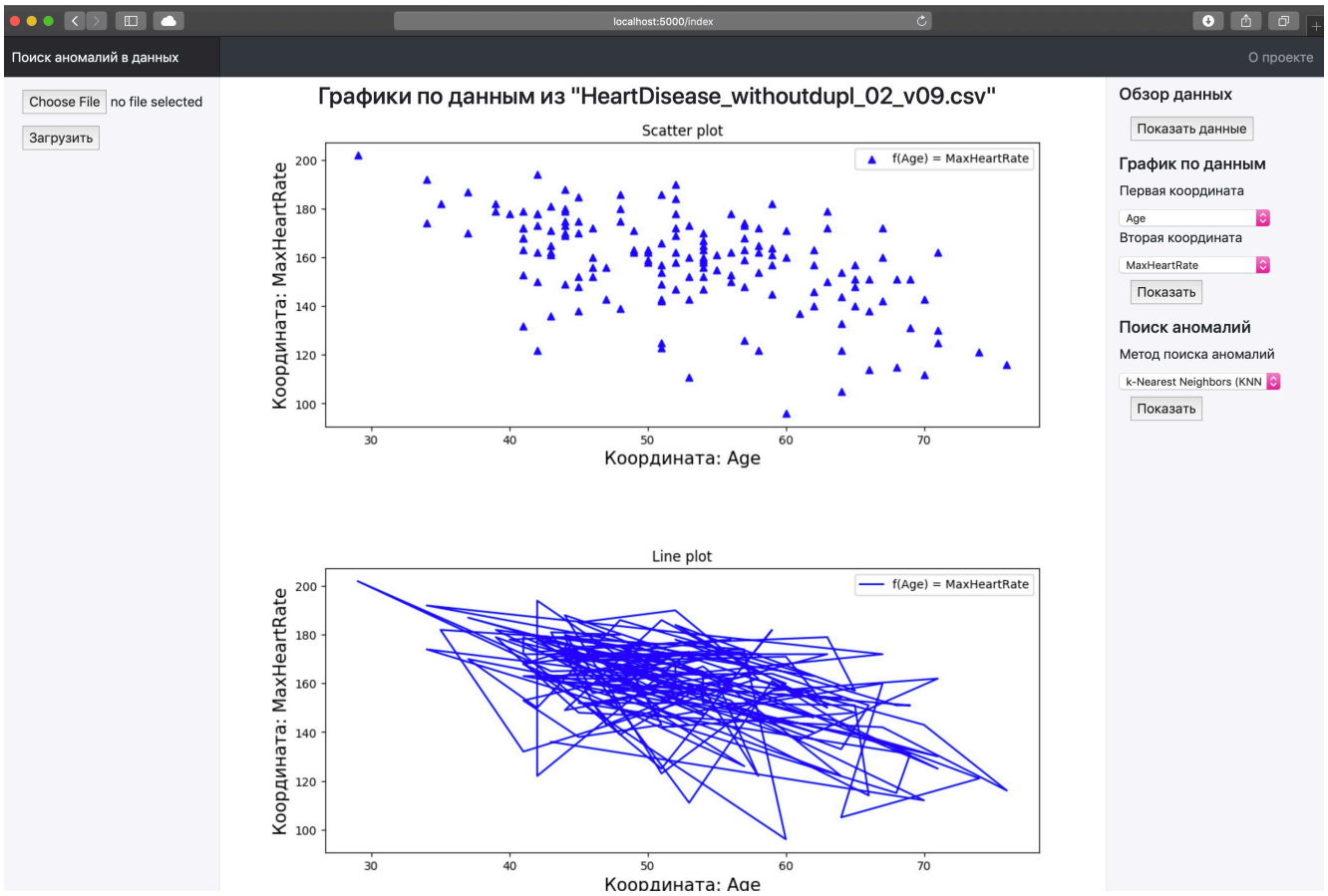


Рисунок 4.5 — Построение графика зависимости для указанных координат. На рисунке представлена зависимость максимального количества ударов в минуту от возраста пациентов (данные из датасета сердечных заболеваний).

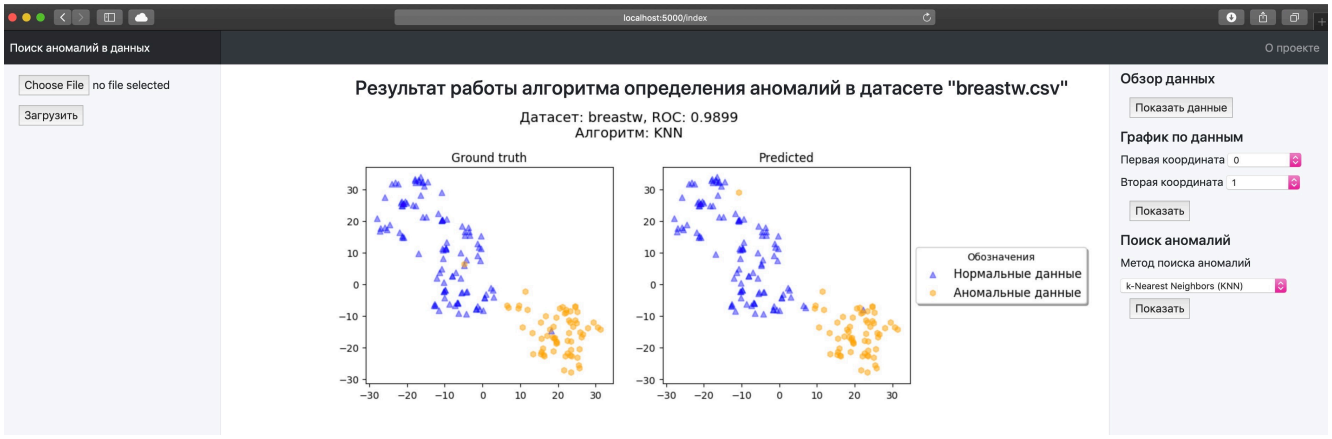


Рисунок 4.6 — Поиск аномалий в данных. Набор данных разбивается на две непересекающиеся части: на первой части выбранная модель обучается, а на второй – проверяется качество обученной модели. Обычно, разделение происходит в соотношении 75% и 25% соответственно. Так снижается вероятность переобучения на данных, что позволяет избежать ухудшения обобщающей способности алгоритма.

# Глава 5. Результаты

## 5.1 Выводы

В ходе работы были проанализированы существующие библиотеки для языка программирования Python, которые упрощают работу с алгоритмами для поиска аномалий в данных.

Была обнаружена и исправлена ошибка в библиотеке PyOD[1].

Построен сервис для анализа данных и определения аномалий. Сервис развёрнут на удалённом сервере с операционной системой Ubuntu 18.04.2 LTS (Bionic Beaver)[2].

В сервисе используется стандартный подход сервер-клиент. В качестве сервера выступает база данных PostgreSQL[3]. С помощью библиотеки Flask[4] для языка программирования Python было построено веб-приложение – на данный момент именно оно выступает в роли клиентского приложения, через который можно получить доступ к функционалу всего сервиса.

Помимо прочего, понадобилось дополнительное время, чтобы развернуть всю систему на сервере и добиться корректной работы. Сервис запущен по адресу http://bit.ly/anomd19.

[1] Pull Request #108 в репозитории PyOD – https://github.com/yzhao062/pyod/pull/108
[2] http://releases.ubuntu.com/18.04/
[3] https://www.postgresql.org/docs/11/
[4] http://flask.pocoo.org

## Словарь терминов

**Аномалия** : результат измерения в статистике, который выделяется из общей выборки.

**Выброс** : см. ”Аномалия”.

**Датасет** : набор данных.

**Дерево** : средство поддержки принятия решений, использующееся в машинном обучении, анализе данных и статистике (”Дерево принятия решений”).

**Лес** : алгоритм машинного обучения, использующий ансамбль решающих деревьев (”Случайный Лес”).

## Список литературы


1. *Dai*, *W.* Directional Outlyingness for Multivariate Functional Data / W. Dai, M. G. Genton. — 2018. — URL: https://stsda.kaust.edu.sa/Documents/2019.DG.CSDA.pdf.
2. *Hodge*, *V. J.* A Survey of Outlier Detection Methodologies / V. J. Hodge, J. Austin. — 2004. — URL: https://www-users.cs.york.ac.uk/vicky/myPapers/Hodge+Austin_OutlierDetection_AIRE381.pdf.
3. *Vakili*, *K.* Finding Multivariate Outliers With FastPCS / K. Vakili, E. Schmitt. — 2013. — URL: https://arxiv.org/pdf/1301.2053.pdf.
4. *Chandola*, *V.* Anomaly Detection: A Survey / V. Chandola, A. Banerjee, V. Kumar. — 2009. — URL: https://www.vs.inf.ethz.ch/edu/HS2011/CPS/papers/chandola09_anomaly-detection-survey.pdf.
5. *Billor*, *N.* BACON: blocked adaptive computationally efficient outlier nominators / N. Billor, A. S. Hadi, P. F. Velleman. — 2000. — URL: https://www.sciencedirect.com/science/article/pii/S0167947399001012.
6. *Wilkinson*, *L.* Visualizing Big Data Outliers through Distributed Aggregation / L. Wilkinson. — 2017. — URL: https://www.cs.uic.edu/~wilkinson/Publications/outliers.pdf.
7. *Karazeev*, *A.* Generative Adversarial Networks (GANs): Engine and Applications / A. Karazeev. — 2017. — URL: https://blog.statsbot.co/generative-adversarial-networks-gans-engine-and-applications-f96291965b47.
8. *Ramaswamy*, *S.* Efficient Algorithms for Mining Outliers from Large Data Sets / S. Ramaswamy, R. Rastogi, K. Shim. — 2000. — URL: https://dl.acm.org/citation.cfm?id=335437.
9. A Novel Anomaly Detection Scheme Based on Principal Component Classifier / M.-L. Shyu [et al.]. — 2003. — URL: https://www.researchgate.net/publication/228709094_A_Novel_Anomaly_Detection_Scheme_Based_on_Principal_Component_Classifier.
10. Estimating the support of a high-dimensional distribution / B. Schölkopf [et al.]. — 2001. — URL: https://eprints.soton.ac.uk/259007/1/TRONECLA.PS.

11. LOF: Identifying Density-Based Local Outliers / M. M. Breunig [et al.]. — 2000. — URL: http://www.dbs.ifi.lmu.de/Publikationen/Papers/LOF.pdf.

12. *Goldstein*, *M.* Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm / M. Goldstein, A. Dengel. — 2012. — URL: https://pdfs.semanticscholar.org/5cf8/81d1db19834f123fcfc79ad32097aeafe17f.pdf.

13. *Liu*, *F. T.* Isolation Forest / F. T. Liu, K. M. Ting, Z.-H. Zhou. — 2008. — URL: https://ieeexplore.ieee.org/abstract/document/4781136.

14. A Supervised Machine Learning Algorithm for Arrhythmia Analysis / H. A. Guvenir [et al.]. — 1997. — URL: http://repository.bilkent.edu.tr/bitstream/handle/11693/27699/bilkent-research-paper.pdf?sequence=1.

15. *Rayana*, *S.* ODDS Library / S. Rayana. — 2016. — URL: http://odds.cs.stonybrook.edu.

## Список рисунков


2.1 Пример ROC-кривой. . . . . 11
2.2 Ядерная оценка плотности 100 нормально распределённых случайных чисел с использованием различных сглаживающие окон. . . 13
2.3 Плотность распределения смеси, состоящей из трёх нормальных распределений ($\mu = [5, 10, 15]$, $\sigma = 2$) с одинаковыми весами. . . . . 15

3.1 Тестовый образец (зелёный круг) должен быть классифицирован как синий квадрат (класс 1) или как красный треугольник (класс 2). Если $k = 3$, то он классифицируется как класс 2, потому что внутри меньшего круга 2 треугольника и только 1 квадрат. Если $k = 5$, то он будет классифицирован как класс 1 (3 квадрата против 2 треугольников внутри большего круга). . . . . 17
3.2 PCA для многомерного гауссового распределения с центром в точке (1, 3) со стандартным отклонением 3. Векторы отражают собственные векторы ковариационной матрицы гауссианы. . . . . 17
3.3 Гиперплоскость $H1$ не разделяет классы. В случае гиперплоскости $H2$ разделение имеется, но зазор слишком маленький. Разделение с максимальным зазором достигается гиперплоскостью $H3$. . . . . 18
3.4 Базовая идея метода – сравнение локальной плотности точки с плотностями её соседей. Точка A имеет меньшую плотность по сравнению с соседями . . . . . 19
3.5 Для изолирования точки $x_i$ требуется 12 случайных разбиений, а для аномальной точки $x_o$ – только 4 разбиения. . . . . 20
3.6 Рассматриваемые наборы данных после применения алгоритма понижения размерности t-SNE. . . . . 23
3.7 Эффективность алгоритмов на разных наборах данных. . . . . 25
3.8 Качество алгоритмов в зависимости от набора данных. . . . . 26
3.9 Датасет Breast Cancer. . . . . 26

4.1 Главный экран сервиса. Есть возможность загрузить данные через окно загрузки (как продемонстрировано рисунке 4.2), либо выбрать набор данных из списка загруженных ранее. . . . . 27

4.2 Пример экрана загрузки набора данных на сервер. . . . . 28
4.3 После выбора файла на экране появляется таблица с некоторыми объектами из указанного набора данных. С помощью меню, которое располагается справа, можно выбрать: (**i**) построение общего графика по набору данных, на котором будут представлены данные, размерность которых была понижена с помощью метода t-SNE (продемонстрировано на рисунке 4.4), **(ii)** график зависимости одной из размерностей данных от другой (продемонстрировано на рисунке 4.5) и (**iii**) выбрать алгоритм из списка, с помощью которого будет осуществлён поиск аномалий в данных (продемонстрировано на рисунке 4.6). . . . . 28
4.4 Представление данных после понижения размерности до dim=2 при помощи алгоритма t-SNE. . . . . 29
4.5 Построение графика зависимости для указанных координат. На рисунке представлена зависимость максимального количества ударов в минуту от возраста пациентов (данные из датасета сердечных заболеваний). . . . . 30
4.6 Поиск аномалий в данных. Набор данных разбивается на две непересекающиеся части: на первой части выбранная модель обучается, а на второй – проверяется качество обученной модели. Обычно, разделение происходит в соотношении 75% и 25% соответственно. Так снижается вероятность переобучения на данных, что позволяет избежать ухудшения обобщающей способности алгоритма. . . . . 30

## Список таблиц

1 Статистика по данным из рассматриваемых наборов данных. . . . . 21
2 Значения ROC для рассматриваемых алгоритмов на данных. . . . . 24