|  |
| --- |
| Московский государственный технический университет  имени Н.Э. Баумана |

Факультет «Биомедицинская техника»  
Кафедра «Медико-технические информационные технологии»

**Л.Н. Анищенко, А.В. Родина, В.В. Слизов, Е.С. Смирнова**

**Решение задач классификации методами машинного обучения**

**Методическое указание к лабораторной работе**

Москва

2021 МГТУ им. Н.Э. Баумана

**СОДЕРЖАНИЕ**

[Цель работы: 3](#_Toc65078907)

[ВВЕДЕНИЕ 3](#_Toc65078908)

[1. Теоретическая часть 8](#_Toc65078909)

[2. Практическая часть 23](#_Toc65078910)

[3. Порядок выполнения работы 32](#_Toc65078911)

[4. Содержание отчета по лабораторной работе 33](#_Toc65078912)

[5. Контрольные вопросы 34](#_Toc65078913)

[6. Список использованных источников 35](#_Toc65078914)

Цель работы: используя методы машинного обучения решить задачу классификации для самостоятельно выбранной медико-биологической базы данных.

# ВВЕДЕНИЕ

С середины прошлого века развитие технологий позволило человечеству собирать большие объемы информации об окружающем нас мире. Результаты исследований во всех областях науки представляют собой огромное количество данных, которое необходимо обработать и проанализировать для извлечения полезной информации об изучаемых процессах. Помимо этого, в современном цифровом мире происходит непрерывный сбор глобальной информации: мобильные операторы и банки собирают информацию о своих клиентах, университеты ведут учет оценок, достижений, выбранных курсов, результатов сессий студентов, медицинские исследования также представляют собой наборы данных.

Для извлечения полезной информации из абстрактных наборов цифр, а именно в таком виде в большинстве случаев содержатся собираемые данные, существует множество методов по их обработке. Можно строить аналитику, пытаться найти паттерны в поведении пользователей или использовать данные, чтобы принимать решения: какие олимпиады учитывать при наборе студентов, какие страницы показывать какой, ставить диагноз пациенту по имеющемуся набору симптомов и результатам анализов.

В большинстве случаев невозможно создать имитационную модель реального процесса, который мы хотим изучить и предсказать развитие. Из таких соображений и возникла идея создания алгоритмов и методов, которые способны строить модели на основе анализа данных, а не исходя из понимания природы самих процессов. Такой метод получил название машинное обучение.

Общий термин «Machine Learning» (ML) или «машинное обучение» (МО) обозначает множество математических, статистических и вычислительных методов для разработки алгоритмов, способных решить задачу не прямым способом, а на основе поиска закономерностей в разнообразных входных данных. Решение вычисляется не по четкой формуле, а по установленной зависимости результатов от конкретного набора признаков и их значений.

Сама идея «обучающихся» машин принадлежит Алану Тьюрингу [1], но непосредственно термин «машинное обучение» был придуман в 1959 году Артуром Сэмюэлем, американским киберспортсменом и пионером в области компьютерных игр и искусственного интеллекта [2, 3]. Одной из первых и наиболее известной книгой об исследованиях возможностей машинного обучения стала книга Нильсона об «обучающихся машинах», выпущенная в 60-ые года прошлого столетия [4]. Интерес, связанный с распознаванием закономерностей и различных паттернов, продолжался и в 1970-е годы, как описано Дудой и Хартом в 1973 году [5]. В 1981 году был представлен доклад, в котором описывалось создание прототипа современных нейронных сетей, основанного на принципах машинного обучения (описывался процесс распознавания различных символов: 26 букв, 10 цифр, и 4 специальных символов) [6].

Позднее, в 1997 году Том Митчелл дал широко цитируемое, более формальное определение алгоритмов, изучаемых в области машинного обучения. Оно формулируется следующим образом: говорят, что компьютерная программа обучается на основе опыта E по отношению к некоторому классу задач T и меры качества P, если качество решения задач из T, измеренное на основе P, улучшается с приобретением опыта E [7].

Приведем более конкретный пример, для демонстрации общих принципов и задач машинного обучения. Одним из наиболее перспективных применений методов машинного обучения является применение его в различных медицинских задачах: постановка диагноза по набору симптомов, результатов анализов и истории прецедентов, определение характера опухоли, развитие того или иного заболевания и т.д.

Рассмотрим задачу определения характера рака молочной железы: является ли опухоль доброкачественной или злокачественной. Представим, что признаком, определяющим характер опухоли, является ее размер. Множество всех возможных размеров опухолей будет называться пространством объектов и обозначаться X (рисунок 1).

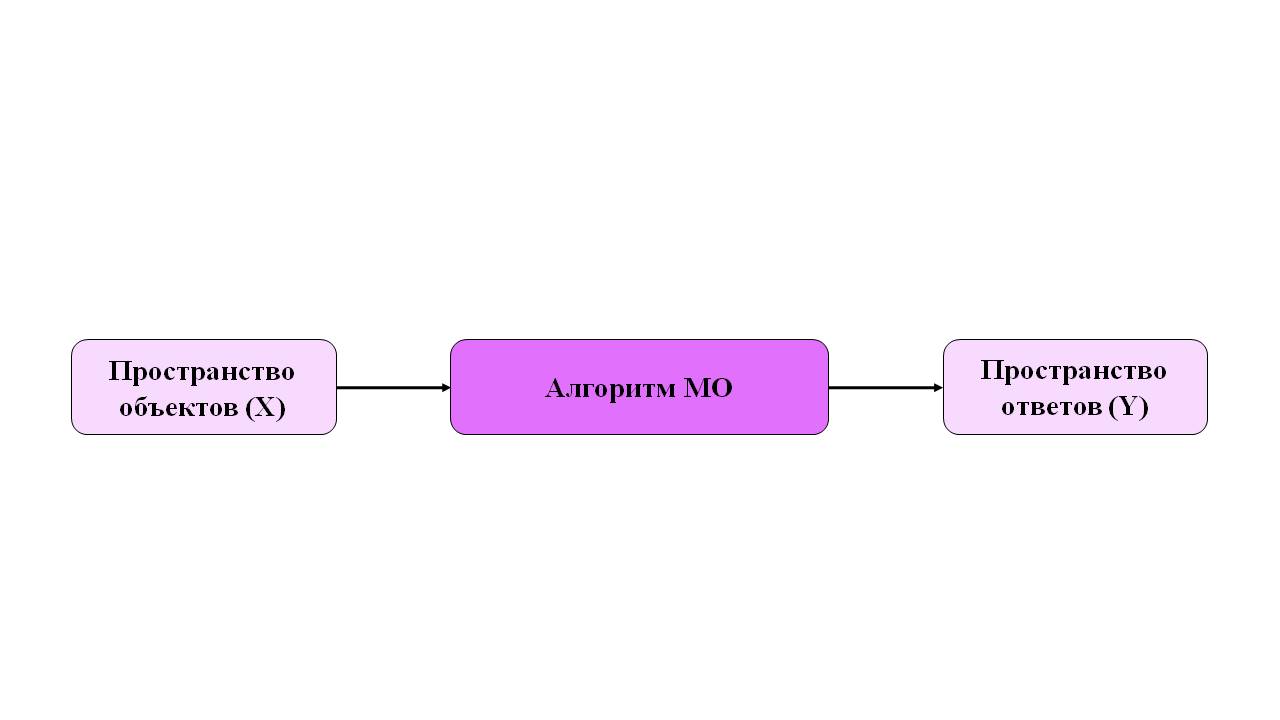
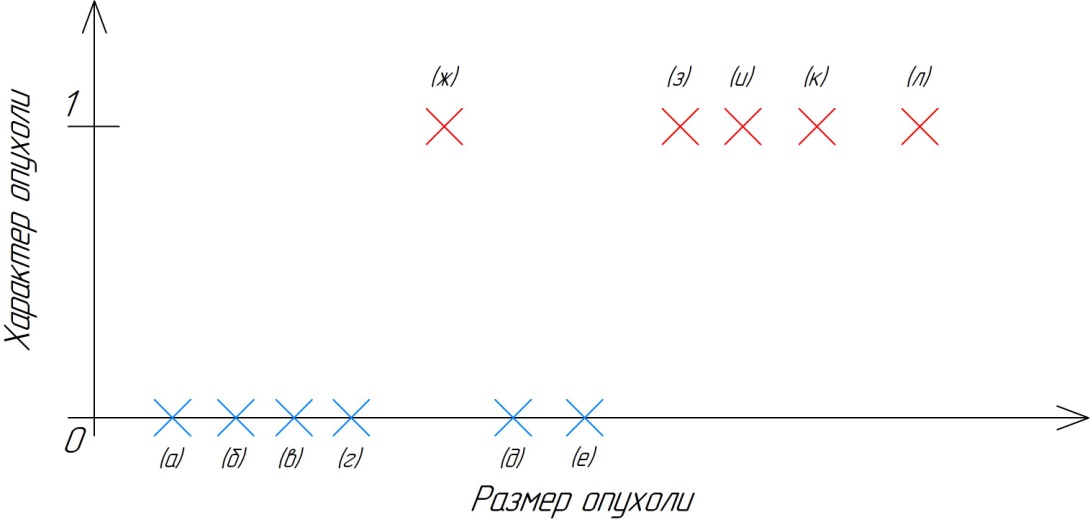


Рисунок 1 – Упрощенная схема применения методов машинного обучения

Конкретно в данном примере, забегая вперед, скажем, что он является задачей классификации, на выходе будет выдаваться дискретное значение: 0 или 1, доброкачественная или злокачественная опухоль соответственно. Ответ, который мы хотим определить, называется целевой переменной, а множество его значений – пространством ответов Y. В нашем случае пространство ответов является бинарной величиной: .

*Примечание: В некоторых задачах классификации прогнозируемая величина может принимать более двух возможных значений: пусть необходимо определить, какой из трёх видов рака груди у пациентки. Алгоритм будет выдавать одно из четырех дискретных значений – 0, 1, 2, 3, , где 0 означает доброкачественную опухоль, единица – первый тип рака, двойка – второй тип, тройка – третий тип. Это по-прежнему проблема классификации, поскольку предсказываемая величина дискретна и означает отсутствие рака или наличие одного из трёх типов рака.*

Мы не являемся специалистами в онкологии, поэтому не можем сделать прогнозы на основе своих экспертных мнений. В нашем распоряжении есть только истории болезней пациентов, в которых сказано, что при определенном размере опухоли, она оказалась доброкачественной или злокачественной соответственно. Каждый такой пример называется обучающим, а вся их совокупность – обучающей выборкой, которая обозначается как , где  – обучающие объекты, i – их количество. Особенность обучающих объектов состоит в том, что для них известны ответы . Попробуем визуализировать описанное выше: на рисунке 2 представлено графическое изображение рассматриваемого примера.

 Рисунок 2 – Визуализация рассматриваемого примера

На оси абсцисс расположен параметр размера опухоли. На оси ординат, обозначающий характер опухоли, существует всего две отметки: 0 и 1, обозначающие доброкачественную и злокачественную опухоли соответственно. Данный график показывает, что был зафиксирован случай, в котором при размере опухоли, соответствующей точке а, она являлась доброкачественной, аналогично для точек с б до е. Однако, в случаях, когда опухоль была размером, соответствующим точке ж, она оказалась злокачественной. Здесь классификация заключается в попытке прогнозирования категориальных меток классов, к которым принадлежат новые образцы, на основе прошлых наблюдений.

*Примечание: В других задачах машинного обучения мы можем иметь более одного признака (атрибута, параметра). Применительно к рассматриваемому примеру это могут быть: возраст пациента, генетический фактор, толщина опухоли, однородность размеров опухолевых клеток, однородность их формы и т.д. Вектор всех признаков объекта x называется признаковым описанием этого объекта.*

В данной ситуации алгоритм машинного обучения может попытаться разделить элементы выборки на две группы (ту, для которой опухоль оказалась доброкачественной и злокачественной, соответственно). Визуализация случая для обучающей выборки, описываемой двумя признаками, приведена на рисунке 3 (признаками в данном случае являются размер опухоли и возраст пациента). Прямая, проведенная на графике, разделяет наилучшим образом объекты выборки.

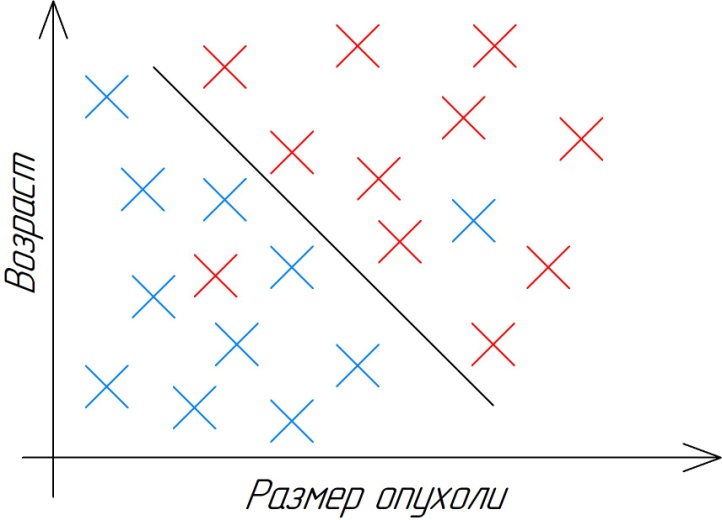


Рисунок 3 – Визуализация выборки, объекты в которой характеризуются двумя признаками

МО позволяет нам пользоваться надежными фильтрами почтового спама, удобным ПО распознавания текста и речи и испытанными поисковыми механизмами. Кроме того, прогресс в медицинских приложениях не ограничивается определением характера опухоли молочной железы. Например, исследователи продемонстрировали, что модели глубокого обучения способны обнаруживать рак кожи [8]. Также хорошие результаты были достигнуты исследователями из DeepMind, которые использовали глубокое обучение для прогнозирования третичных структур белков, впервые превзойдя физические подходы [9].

# 1. Теоретическая часть

В данном разделе будут рассмотрены три типа машинного обучения: обучение с учителем (supervised learning), обучение без учителя (unsupervised learning) и обучение с подкреплением (reinforcement learning). Будут объяснены фундаментальные отличия между указанными тремя типами обучения и приведена более подробная классификация существующих подзадач трех указанных типов МО.



Рисунок 4 – Типы обучения и предметные области, в которых они могут применяться

## 1.1 Обучение с учителем

Главная цель обучения с учителем – обучить модель на помеченных обучающих данных, что позволит вырабатывать прогнозы на не встречавшихся ранее или будущих данных. Здесь понятие «с учителем» относится к набору обучающих образцов (входных данных), где желаемые выходные сигналы (метки) уже известны [10]. На рисунке 5 представлена типичная последовательность действий при обучении с учителем, где помеченные обучающие данные передаются алгоритму МО для подгонки к прогнозирующей модели, которая может вырабатывать прогнозы на новых непомеченных входных данных.

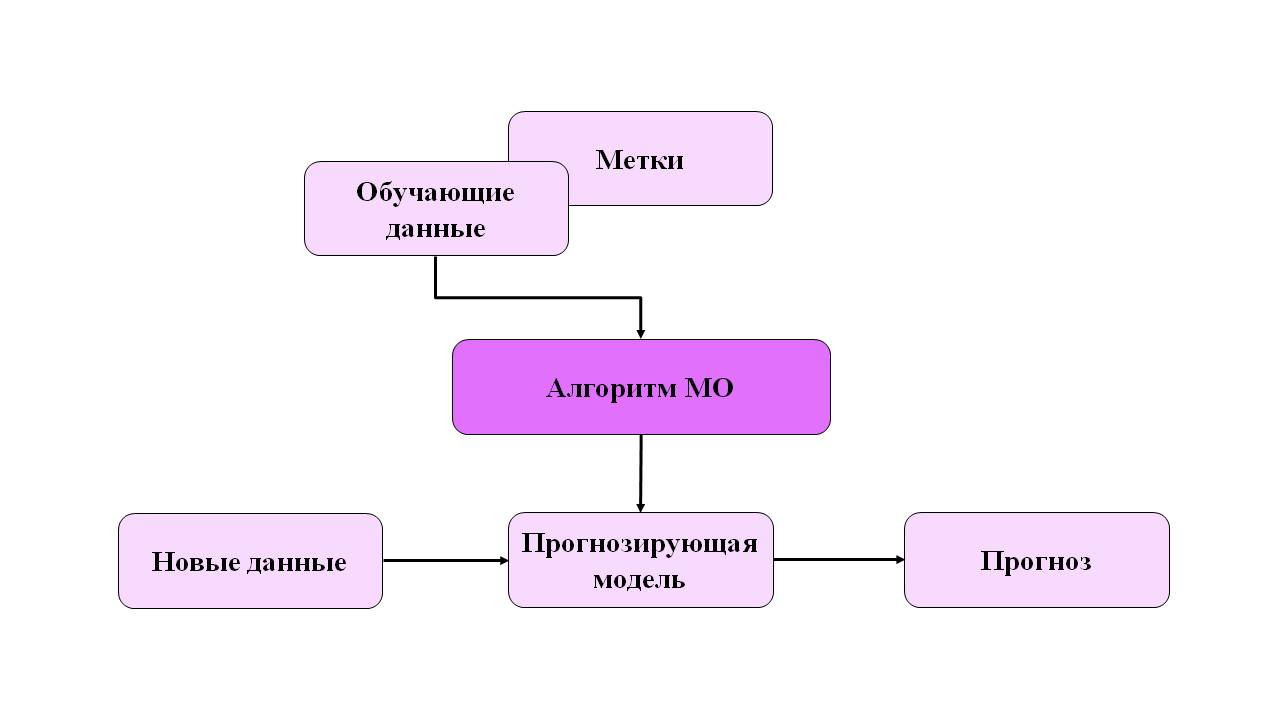


Рисунок 5 – Типичная последовательность действия при обучении с учителем

Рассматривая в качестве примера фильтрацию почтового спама, мы можем обучить модель с использованием алгоритма машинного обучения с учителем на корпусе помеченных почтовых сообщений, т.е. сообщений, которые корректно маркированы как спам или не спам, и прогнозировать, к какой из этих двух категорий принадлежит новое сообщение. Задача обучения с учителем с метками дискретных классов, такими как в нашем примере фильтрации почтового спама, также называется задачей классификации. Другой подкатегорией обучения с учителем является регрессия, в которой результирующий сигнал представляет собой непрерывную величину [10].

### 1.1.1 Классификация для прогнозирования меток классов

Классификация – это подкатегория обучения с учителем, где целью является прогнозирование категориальных меток классов, к которым принадлежат новые образцы, на основе прошлых наблюдений [10]. Такие метки классов представляют собой дискретные неупорядоченные значения, которые могут пониматься как принадлежность к группам образцов. Ранее упомянутый пример выявления почтового спама демонстрировал типичную задачу двоичной классификации, когда алгоритм МО изучал набор правил, чтобы проводить различие между двумя возможными классами: спам-сообщениями и не спам-сообщениями.

Пример классификации был рассмотрен во введении. Иллюстрация рассматриваемого примера приведена на рисунках 2 и 3. Существуют следующие типы задачи классификации [11]:

*  – бинарная классификация. Например, мы можем предсказывать, сдаст ли студент сессию, случится ли определенное заболевание с пациентом (на основе, скажем, его генома).
*  – многоклассовая классификация. Примером может служить определение предметной области для научной статьи (математика, биология, психология и т.д.).
*  – многоклассовая классификация с пересекающимися классами. Примером может служить задача автоматического проставления тегов для ресторанов (логично, что ресторан может одновременно иметь несколько тегов).

Характерным примером задачи многоклассовой классификации считается распознавание рукописных символов. Мы можем подготовить обучающий набор данных, содержащий множество рукописных примеров для каждой буквы алфавита. Буквы (А, В, С и т. д.) будут представлять различные неупорядоченные категории или метки классов, которые мы хотим прогнозировать. Если теперь пользователь с помощью устройства ввода предоставит новый рукописный символ, тогда наша прогнозирующая модель будет в состоянии с определенной точностью предсказать корректную букву алфавита для введенного рукописного символа. Тем не менее, наша система МО не будет способна правильно распознавать, например, цифры между 0 и 9, если они не входили в состав обучающего набора данных.

В рамках данной лабораторной работы предлагается изучения типа машинного обучения с учителем, в частности решение задачи классификации. Существует большое количество методов решения задач классификации, суть принципов работы некоторых из них рассмотрим ниже.

### Логистическая регрессия

Логистическая регрессия названа для функции, используемой в основе метода, логистической функции. Это S-образная кривая (сигмоида), которая может принимать любое действительное число и отображать его в значение от 0 до 1 (рисунок 6). Это весьма полезно, так как мы можем применить правило к выходу логистической функции для привязки к 0 и 1 и предсказания класса.

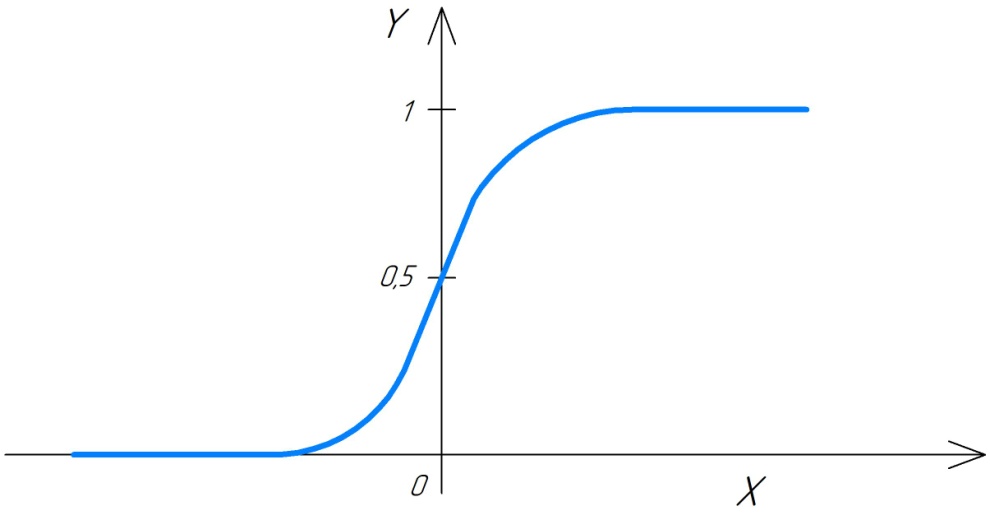


Рисунок 6 – График сигмоиды

Формула сигмоиды, представленной выше на графике:

 (1)

Благодаря тому, как обучается модель, предсказания логистической регрессии можно использовать для отображения вероятности принадлежности образца к классу 0 или 1. Подробнее о логистической регрессии можно узнать в книге [12].

### Метод k-ближайших соседей

Цель метода k-ближайших соседей – классифицировать объекты на основе их сходства. Класс, который присвоят объекту – это класс, наиболее часто встречающийся среди k ближайших соседей.

Рассмотрим задачу присвоения белого круга классу 1 (черные круги) или классу 2 (серые круги), представленной ниже на рисунке 7.

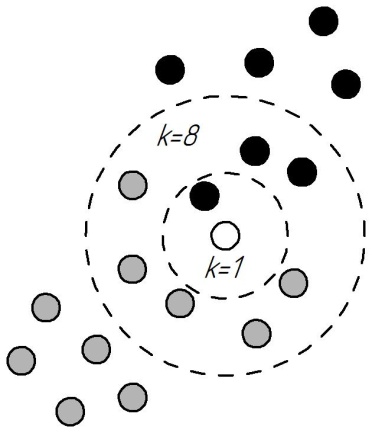


Рисунок 7 – Визуализация работы метода k-ближайших соседей

Рассмотрим частный случай метода k-ближайших соседей, когда k = 1. В этом случае метод k-ближайших соседей присвоит белому кругу класс 1, то есть раскрасит его в черный цвет. При увеличении количества ближайших соседей до k = 8, согласно рисунку внутри круга находятся три объекта класса 1 и пять объектов класса 2. Метод k-ближайших соседей классифицирует белый круг как объект класса 2, поскольку они составляют большинство, и раскрасит его в серый цвет. Подробнее о методе k-ближайших соседей можно узнать в книге [10].

### Линейный дискриминантный анализ

Система использует статистические методы, объединив их с машинным обучением, чтобы получить более точный результат. Результаты, как правило, применяют для создания линейного классификатора.

Рассмотрим случай, когда не существует прямой линии на двумерной плоскости, которая могла бы полностью разделить два класса точек данных (рисунок 8а). Метод линейного дискриминантного анализа использует обе оси (X и Y) для создания новой оси и проецирует данные на новую ось таким образом, чтобы максимизировать разделение двух категорий и, следовательно, превратить двумерный график в одномерный (рисунок 8б).

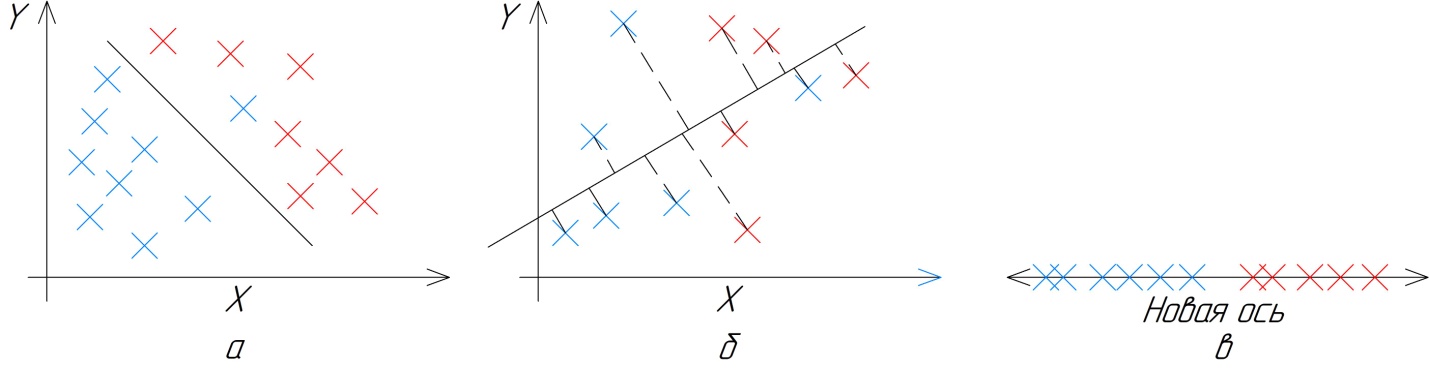


Рисунок 8 – Визуализация работы метода линейного дискриминантного анализа

После генерации этой новой оси с использованием вышеупомянутых критериев все точки данных классов нанесены на эту новую ось и показаны на рисунке 8в. Подробнее о линейной дискриминантном анализе можно узнать в книге [10].

### Наивный байесовский метод

Наивный байесовский алгоритм – это алгоритм классификации, основанный на теореме Байеса с допущением о независимости признаков. Другими словами, НБА предполагает, что наличие какого-либо признака в классе не связано с наличием какого-либо другого признака. Например, фрукт может считаться яблоком, если он красный, круглый и его диаметр составляет порядка 8 сантиметров. Даже если эти признаки зависят друг от друга или от других признаков, в любом случае они вносят независимый вклад в вероятность того, что этот фрукт является яблоком. В связи с таким допущением алгоритм называется «наивным».

Теорема Байеса позволяет рассчитать апостериорную вероятность :

 (2)

где  – апостериорная вероятность данного класса c при данном значении признака x;

 – априорная вероятность данного класса;

 – апостериорная вероятность данного значения признака x при данном классе c;

 – априорная вероятность данного значения признака.

Теорема Байеса позволяет переставить местами причину и следствие. Зная с какой вероятностью причина приводит к событию, эта теорема позволяет рассчитать вероятность того что именно эта причина привела к наблюдаемому событию.

Цель классификации состоит в том, чтобы понять к какому классу принадлежит объект, поэтому нужно найти не саму вероятность, а наиболее вероятный класс. То есть необходимо рассчитать вероятность для всех классов и выбрать тот класс, который обладает максимальной вероятностью. Подробнее о методе Байеса можно узнать в книге [10].

### Дерево решений

Дерево решений – математическая модель, которая задаёт процесс принятия решений так, что будут отображены каждое возможное решение, предшествующие и последующие этим решениям события или другие решения и последствия каждого конечного решения.

Для принятия решения, к какому классу следует отнести некоторый объект или ситуацию, требуется ответить на вопросы, стоящие в узлах этого дерева, начиная с его корня.

Рассмотрим визуализацию дерева решений, представленную ниже на рисунке 9, на примере выбора месяца для начала написания дипломной работы.

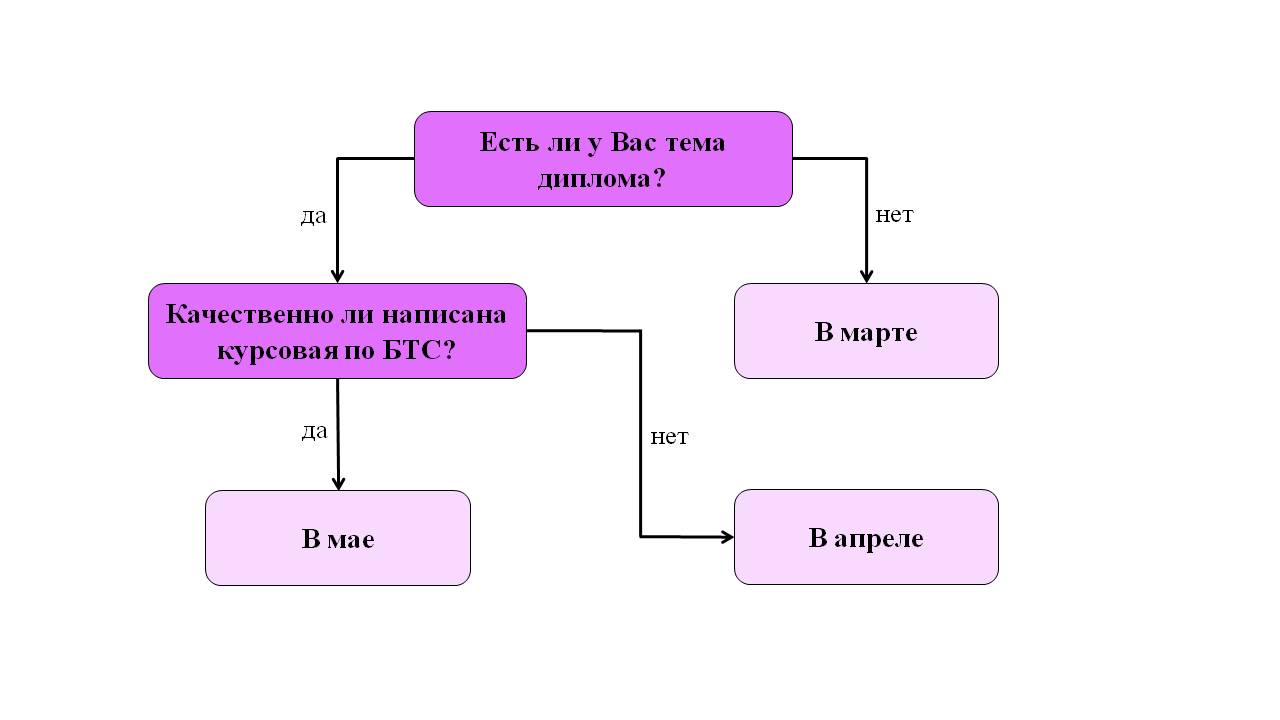


Рисунок 9 – Пример дерева решений

Вопрос о теме диплома расположен в корневом узле дерева, вопрос о курсовой работе в узле дерева, а ответы о месяце начала написания работы являются листьями дерева. Стоит отметить, что данный алгоритм склонен к переобучению в случае большого количества признаков и, следовательно, узлов дерева. Подробнее о методе дерева решений можно узнать в книге [10].

### Метод опорных векторов

Идею метода удобно проиллюстрировать на следующем простом примере: даны точки на плоскости, разбитые на два класса (рисунок 10).

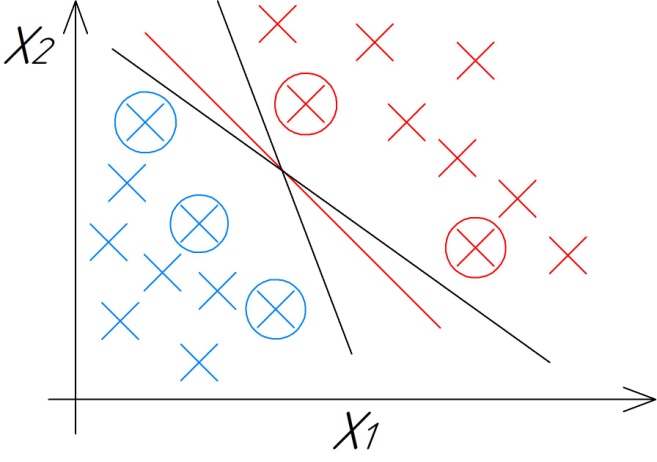


Рисунок 10 – Визуализация работы метода опорных векторов

В нашем примере существует несколько прямых, разделяющих два. С точки зрения точности классификации лучше всего выбрать прямую, расстояние от которой до каждого класса максимально. Другими словами, выберем ту прямую, которая разделяет классы наилучшим образом (красная прямая). Такая прямая, а в общем случае – гиперплоскость, называется оптимальной разделяющей гиперплоскостью. Вектора, лежащие ближе всех к разделяющей гиперплоскости, называются опорными векторами (на рисунке 10 они помечены красным). Подробнее о методе опорных векторов можно узнать в книге [10].

### 1.1.2 Регрессия для прогнозирования непрерывных результатов

Вторым типом обучения с учителем является прогнозирование непрерывных результатов, которое также называется регрессионным анализом [10]. При регрессионном анализе мы имеем несколько признаков (X) и переменную непрерывного отклика (Y) и пытаемся отыскать между указанными переменными взаимосвязь, которая позволила бы прогнозировать результат.

Скажем, пусть нас интересует прогнозирование оценок, получаемых студентами в результате прохождения теста по математике. Если существует какая-то взаимосвязь между временем, потраченным на подготовку к прохождению теста, и финальными оценками, то мы могли бы применить ее в качестве обучающих данных с целью обучения модели, которая будет использовать время подготовки для прогнозирования будущих оценок студентов, планирующих пройти тест. На рисунке 11 демонстрируется концепция линейной регрессии.

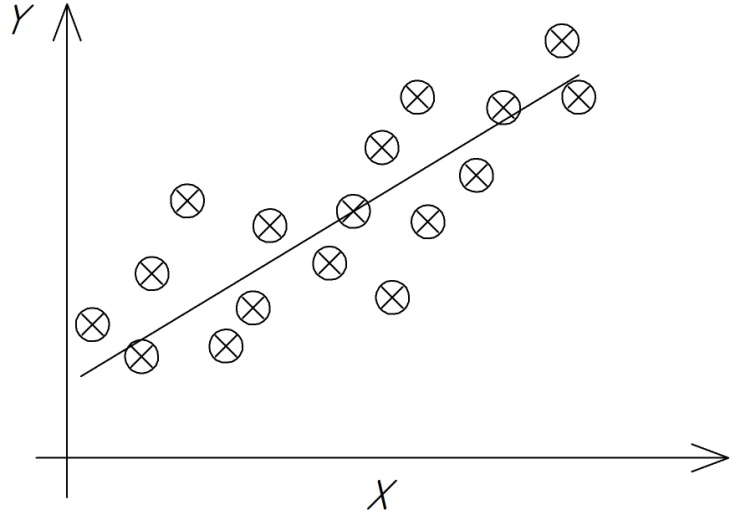


Рисунок 11 – Концепция линейной регрессии

Для переменной признака Х и целевой переменной Y мы подгоняем прямую линию к этим данным, чтобы свести к минимуму расстояние – обычно среднеквадратическое – между точками данных и подогнанной линией. Далее становится возможным вычислить угол наклона данной прямой, для прогнозирования целевой переменной новых данных.

## 1.2 Обучение без учителя

При обучении с учителем правильный ответ нам известен заранее, когда мы обучаем модель, а при обучении с подкреплением мы определяем меру награды для отдельных действий, предпринимаемых агентом. Тем не менее, при обучении без учителя мы имеем дело с непомеченными данными или данными с неизвестной структурой. Использование приемов обучения без учителя дает нам возможность исследовать структуру данных для извлечения значимой информации без управления со стороны известной целевой переменной или функции награды. Подтипами данного вида МО являются [10]:

* Кластеризация – задача разделения объектов на группы, обладающие некоторыми свойствами. Примером может служить кластеризация документов из электронной библиотеки или кластеризация абонентов мобильного оператора.
* Оценивание плотности – задача приближения распределения объектов. Примером может служить задача обнаружения аномалий, в которой на этапе обучения известны лишь примеры «правильного» поведения оборудования, а в дальнейшем требуется обнаруживать случаи некорректной работы. В таких задачах сначала оценивается распределение «правильных» объектов, а затем аномальными объявляются все объекты, которых в рамках этого распределения получают слишком низкую вероятность.
* Визуализация – задача изображения многомерных объектов в двумерном или трехмерном пространстве таким образом, чтобы сохранялось как можно больше зависимостей и отношений между ними.
* Понижение размерности – задача генерации таких новых признаков, что их меньше, чем исходных, но при этом с их помощью задача решается не хуже (или с небольшими потерями качества, или лучше – зависит от постановки). К этой же категории относится задача построения латентных моделей, где требуется описать процесс генерации данных с помощью некоторого (как правило, небольшого) набора скрытых переменных. Примерами являются задачи тематического моделирования и построения рекомендаций, которым будет посвящена часть курса.

### 1.2.1 Нахождение подгрупп с помощью кластеризации

Кластеризация – это исследовательская методика анализа данных, которая позволяет организовать нагромождение информации в виде содержательных подгрупп (кластеров), не обладая какими-то априорными знаниями о членстве в группах [10]. Каждый кластер, появляющийся во время анализа, устанавливает группу объектов, которые обладают определенной степенью подобия, но менее похожи на объекты в других кластерах. По указанной причине кластеризацию иногда называют классификацией без учителя. Кластеризация является хорошим способом структурирования информации и выведения значимых взаимосвязей из данных. Например, она предоставляет специалистам по маркетингу возможность выявления групп на основе их интересов для разработки индивидуальных маркетинговых программ.

На рисунке 12 показано, как можно применить кластеризацию с целью организации непомеченных данных в три отдельных группы, основываясь на сходстве их признаков  и .

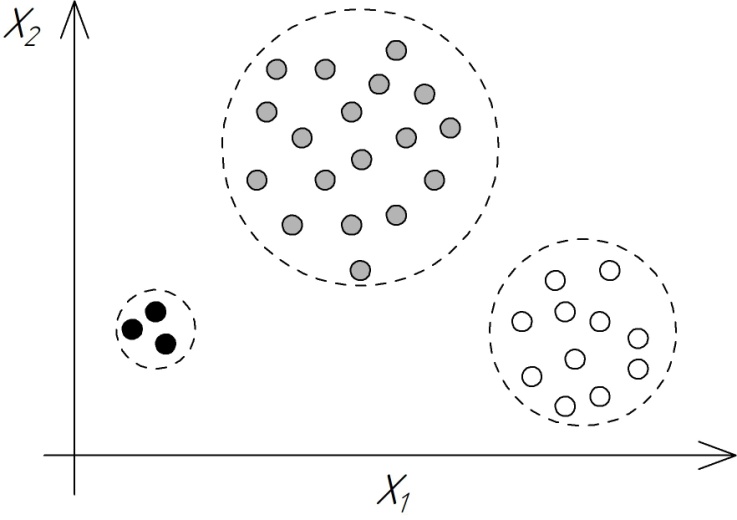


Рисунок 12 – Пример использования кластеризации

### 1.2.2 Понижение размерности для сжатия данных

Понижение размерности – еще одна подобласть обучения без учителя. Зачастую приходится работать с данными высокой размерности (каждое наблюдение сопровождается большим количеством измерений), которые могут представлять проблему для ограниченного пространства хранения и вычислительной мощности алгоритмов МО [10]. Понижение размерности без учителя является распространенным подходом к предварительной обработке признаков для устранения из данных шума, который также может приводить к ухудшению прогнозирующей эффективности определенных алгоритмов, и сжатия данных в подпространство меньшей размерности с сохранением большинства существенной информации.

## 1.3 Обучение с подкреплением

Третьим типом МО, который в рамках данной лабораторной работы нам не пригодятся, но необходимо его рассмотреть, так как он является одним из типов МО, является обучение с подкреплением (рисунок 13). Целью такого обучения является разработка системы (агента), которая улучшает свои характеристики на основе взаимодействий со средой [10].

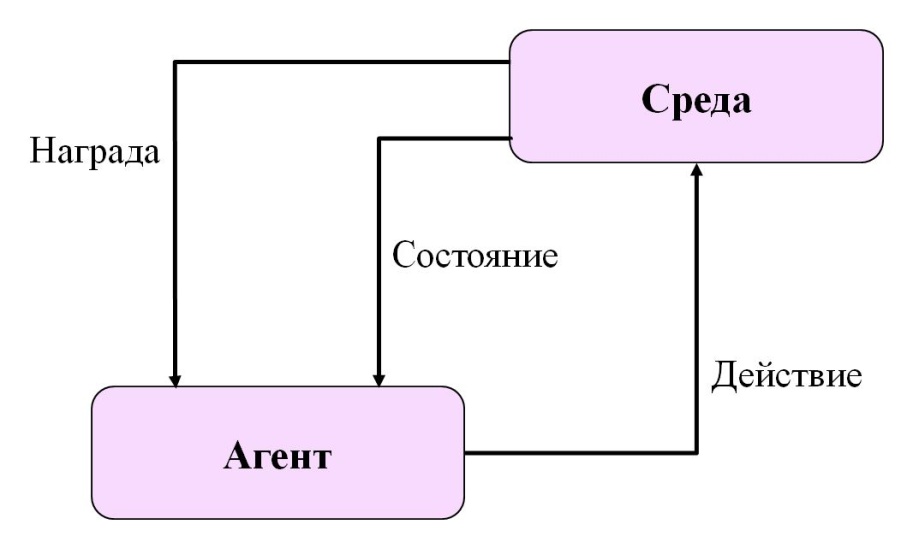


Рисунок 13 – Концепция обучения с подкреплением

Поскольку информация о текущем состоянии среды обычно включает так называемый сигнал награды, мы можем трактовать обучение с подкреплением как область, родственную обучению с учителем. Однако при обучении с подкреплением такая обратная связь не будет истинной меткой или значением, а мерой того, насколько хорошо действие было оценено функцией наград. При взаимодействии со средой агент может использовать обучение с подкреплением для выявления последовательности действий, которые доводят до максимума награду, применяя исследовательский метод проб и ошибок или совещательное планирование [10].

Популярный пример обучения с подкреплением – шахматный движок. Агент выбирает последовательность ходов в зависимости от состояния доски (среды), а награда может быть определена как «выигрыш» или «проигрыш» в конце игры.

Различают много подтипов обучения с подкреплением. Тем не менее, общепринятая схема обучения с подкреплением заключается в том, что агент пытается довести до максимума награду через последовательность взаимодействий со средой. С каждым состоянием можно ассоциировать положительную или отрицательную награду, причем награда может быть определена как достижение общей цели, подобной выигрышу или проигрышу партии в шахматы. Так, в шахматной игре результат каждого хода можно представлять себе как отличающееся состояние среды.

Обучение с подкреплением учится выбирать последовательность действий, доводящих до максимума итоговую награду, которая может быть получена либо немедленно, либо через отсроченную обратную связь.

## 1.4 Условные обозначения и основные термины

В области машинного обучения принято визуализировать выборку, как показано ниже в таблице 1.

Таблица 1 – Представление данных для машинного обучения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер объекта | Признак 1  X1 | Признак 2  X2 | Признак 3  X3 | Целевая переменная  Y |
| 1 | X1(1) | X2(1) | X3(1) | Y1 |
| 2 | X1(2) | X2(2) | X3(2) | Y2 |
| … | … | … | … | … |
| m | X1(m) | X2(m) | X3(m) | Ym |
| n – количество признаков (в данном случае 3: X1, X2, X3); m – количество объектов в выборке; x(i) – признак i-го объекта; xj(i) – значение j-го признака i-го объекта. | | | | |

Чаще всего данная таблица записывается в виде матрицы, так как при написании алгоритмов на любом языке программирования удобнее всего представлять данные в матричном виде:



В таком случае можно записать следующее:





Ниже будет приведена подборка часто применяемых терминов и их синонимов, которые часто используются в литературе по МО:

База данных для машинного обучения – это обработанная и структурированная информация в табличном виде.

Объект – строка в таблице данных.

Признак (feature) – столбец в таблице данных (значение x).

Вектор признаков (feature vector) – n-мерный вектор признаков, который представляет некоторый объект.

Обучение (training) – подгонка моделей, которая в случае параметрических моделей подобна оценке параметров.

Обучающая выборка – набор данных, который используется для настройки и оптимизации параметров алгоритма машинного обучения.

Тестовая выборка – набор данных, которые используются для оценки работы алгоритма машинного обучения.

# 2. Практическая часть

В рамках данной лабораторной работы предлагается решить задачу классификации с использованием метода логистической регрессии, описанной в теоретической части данных методических указаний.

Для поиска доступной для использования базы данных используют следующие сервисы:

* Kaggle [https://www.kaggle.com/datasets];
* Google Dataset Search [https://datasetsearch.research.google.com/];
* UCI Machine Learning Repository

[https://archive.ics.uci.edu/ml/datasets.php];

* DATA USA [https://datausa.io/].

Воспользуемся сервисом UCI Machine Learning Repository, в котором в разделе задач для классификации была выбрана база данных из 299 пациентов с сердечной недостаточностью, собранная в 2015 году и прогнозирующая выживаемость. Пациенты представлены 105 женщинами и 194 мужчинами в возрасте от 40 до 95 лет. База данных включает тринадцать клинических признаков, которые сообщают клиническую информацию, информацию о теле и образе жизни пациентов:

* Возраст: возраст пациента (лет)
* Анемия: снижение эритроцитов или гемоглобина (логическое значение)
* Высокое кровяное давление: если у пациента гипертония (булево значение)
* Креатининфосфокиназа (КФК): уровень фермента КФК в крови (мкг / л)
* Диабет: если у пациента диабет (логическое значение)
* Фракция выброса: процент крови, покидающей сердце при каждом сокращении (в процентах)
* Тромбоциты: тромбоциты в крови (килотромбоциты / мл)
* Пол: женщина или мужчина (бинарный)
* Креатинин сыворотки: уровень сывороточного креатинина в крови (мг / дл)
* Сывороточный натрий: уровень сывороточного натрия в крови (мЭкв / л)
* Курение: если пациент курит или нет (логическое)
* Время: период наблюдения (дни)
* Событие смерти: если пациент скончался в течение периода наблюдения (логическое значение).

Алгоритм бинарной логистической регрессии будем реализовывать на языке Python, для его реализации будут необходимы следующие библиотеки: Scikit-Learn и Pandas. Листинг кода для импорта перечисленных библиотек представлен ниже (листинг 1).

Листинг 1

|  |
| --- |
| # 1. Импорт необходимых библиотек  import pandas as pd  import sklearn |

Следующим шагом нам необходимо считать выбранную и скачанную базу данных, например, в структуру файлов DataFrame библиотеки pandas. Для этого необходимо создать массив заголовков колонок в базе данных, в роли которых в нашем случае выступают 13 перечисленных выше признаков. Листинг кода для перечисленных этапов представлен ниже (листинг 2).

Листинг 2

|  |
| --- |
| # 2. Загрузка базы данных  # Перечислим заголовки колонок с признаками  header = ['age', 'anaemia', 'creatinine\_phosphokinase',  'diabetes', 'ejection\_fraction', 'high\_blood\_pressure',  'platelets', 'serum\_creatinine', 'serum\_sodium', 'sex', 'smoking', 'time', 'DEATH\_EVENT']  # Сформируем DataFrame на основе выбранной базы данных  data = pd.read\_csv ('heart\_failure\_dataset.csv', names = header) |

Следующим шагом необходимо сформировать вектор признаков X и целевую переменную Y, для этого необходимо разделить базу данных на соответствующие элементы. Листинг кода для данного шага представлен ниже (листинг 3).

Листинг 3

|  |
| --- |
| # 3. Разделим данные на вектор признаков Х и целевую переменную Y  Y = data['DEATH\_EVENT'].values  X = data.drop(columns=['DEATH\_EVENT']) |

Используемые в классификации признаки имеют разные единицы измерения и разные шкалы, что может существенно искажать работу классификатора. Для решения этой проблемы перед применением метода логистической регрессии была произведена нормализация вектора признаков по методу минимакса – линейное преобразование данных в диапазоне от 0 до 1, где минимальное и максимальное масштабируемые значения соответствуют 0 и 1 соответственно. Реализация нормализации представлена ниже в листинге 4.

Листинг 4

|  |
| --- |
| # 4. Нормализуем значения вектора признаков  X = (X-X.min())/(X.max()-X.min()) |

Имеющуюся базу данных необходимо рандомизировать и разделить на две выборки:

1. Обучающая выборка – набор данных, который используется для обучения классификатора (67 % от первоначального массива данных);
2. Тестовая выборка – набор данных, которые используются для оценки работы классификатора (33 % от первоначального массива данных).

Рандомизация нужна для формирования двух выборок, в которых будет присутствовать оба класса целевой переменной, так как зачастую базы данных имеют упорядоченную структуру. Реализация перечисленных шагов представлена ниже в листинге 5.

Листинг 5

|  |
| --- |
| # 5. Рандомизируем выборку и разделим на обучающую и тестовую выборки  from sklearn.model\_selection import train\_test\_split  X\_train, X\_test, Y\_train, Y\_test = train\_test\_split (X, Y, test\_size=0.33, random\_state=1, stratify=Y) |

Выборки сформированы и можно подойти непосредственно к классификации с использованием логистической регрессии, программная реализация представлена ниже в листинге 6.

Листинг 6

|  |
| --- |
| # 6. Решим задачу классификации методом логистической регрессии  from sklearn.linear\_model import LogisticRegression  LR\_model = LogisticRegression()  # Обучим модель на обучающей выборке  LR\_model.fit(X\_train, Y\_train)  # Предскажем класс тестовой выборки  LR\_prediction = LR\_model.predict(X\_test) |

Для оценки работы классификатора необходимо оценить точности (accuracy) классификации модели на обучающей и тестовой выборках (доли правильных ответов классификатора), что возможно реализовать встроенными функциями, как приведено ниже в листинге 7, в котором также представлен результат работы алгоритма.

Листинг 7

|  |
| --- |
| # 7. Оценим точность классификации  LR\_train\_accuracy = LR\_model.score(X\_train, Y\_train)  LR\_test\_accuracy = LR\_model.score(X\_test, Y\_test)  print ('Точность модели на обучающей выборке: ', round (LR\_train\_accuracy, 2))  print ('Точность модели на тестовой выборке: ', round (LR\_test\_accuracy, 2)) |
| **Точность модели на обучающей выборке: 0.86**  **Точность модели на тестовой выборке: 0.80** |

Одним из методов оценки качества классификации является разделение результатов на следующие группы:

* Истинно-положительные результаты (ИП) – классификатор верно отнес объект к рассматриваемому классу;
* Истинно-отрицательные результаты (ИО) – классификатор верно утверждает, что объект не принадлежит к рассматриваемому классу;
* Ложноположительные результаты (ЛП) – классификатор неверно отнес объект к рассматриваемому классу;
* Ложноотрицательные результаты (ЛО) – классификатор неверно утверждает, что объект не принадлежит к рассматриваемому классу.

Для наглядного представления результатов используют матрицу несоответствий (confusion matrix) [13], которая представляет собой таблицу следующего вида для задач бинарной классификации и представлена ниже в таблице 1.

Таблица 1 – Общий вид матрицы несоответствий для двух классов

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Реальный класс** | |
| **Положительный** | **Отрицательный** |
| **Предсказанный**  **класс** | **Положительный** | ИП | ЛП |
| **Отрицательный** | ЛО | ИО |

Программная реализация построения такой матрицы и ее результат для исследуемой базы данных представлена ниже в листинге 8.

Листинг 8

|  |
| --- |
| # 8. Построим матрицу несоответствий  from sklearn.metrics import confusion\_matrix  print('Матрица несоответствий метода LR:\n', confusion\_matrix(LR\_prediction, Y\_test)) |
| **Матрица несоответствий метода *LR:***  ***[[65 18]***  ***[ 2 14]]*** |

Из выведенной на экран матрицы несоответствий следует, что число ИП результатов равно 65, ИО – 14, ЛП – 18 и ЛО – 2. Таким образом, можно отметить, что классификатор затрудняется разделять классы.

Как одно из решений данной проблемы предлагается сравнить полученные с помощью логистической регрессии результаты с результатами, полученными другими методами решений задач классификации для поиска наиболее подходящего классификатора для данной базы данных. Ниже в листинге 9 представлена реализация метода k-ближайших соседей и результат работы данного классификатора.

Листинг 9

|  |
| --- |
| # 9. Метод k-ближайших соседей  from sklearn.neighbors import KNeighborsClassifier  KNN\_model = KNeighborsClassifier(n\_neighbors = 11)  KNN\_model.fit(X\_train,Y\_train)  KNN\_prediction = KNN\_model.predict(X\_test)  print('Матрица несоответствий метода KNN:\n', confusion\_matrix(KNN\_prediction, Y\_test))  KNN\_train\_accuracy = KNN\_model.score(X\_train, Y\_train)  KNN\_test\_accuracy = KNN\_model.score(X\_test, Y\_test)  print ('Точность модели на обучающей выборке: ', round (KNN\_train\_accuracy, 2))  print ('Точность модели на тестовой выборке: ', round (KNN\_test\_accuracy, 2)) |
| **Матрица несоответствий метода *KNN:***  ***[[64 28]***  ***[ 3 4]]***  ***Точность модели на обучающей выборке: 0.76***  ***Точность модели на тестовой выборке: 0.69*** |

Согласно матрице несоответствий метод k-ближайших соседей хуже справляется с определением второго класса (события смерти), так как значительно возросло число ЛП результатов и уменьшилось число ИО. Точности модели на обучающей и тестовой выборках также оказались ниже. Далее попробуем использовать другие методы классификации.

В листинге 10 представлена реализация и результат работы метода линейного дискриминантного анализа.

Листинг 10

|  |
| --- |
| # 10. Метод линейного дискриминантного анализа  from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis  LDA\_model = LinearDiscriminantAnalysis()  LDA\_model.fit(X\_train, Y\_train)  LDA\_prediction = LDA\_model.predict(X\_test)  print('Матрица несоответствий метода LDA:\n', confusion\_matrix(LDA\_prediction, Y\_test))  LDA\_train\_accuracy = LDA\_model.score(X\_train, Y\_train)  LDA\_test\_accuracy = LDA\_model.score(X\_test, Y\_test)  print ('Точность модели на обучающей выборке: ', round (LDA\_train\_accuracy, 2))  print ('Точность модели на тестовой выборке: ', round (LDA\_test\_accuracy, 2)) |
| **Матрица несоответствий метода *LDA:***  ***[[63 16]***  ***[ 4 16]]***  ***Точность модели на обучающей выборке: 0.85***  ***Точность модели на тестовой выборке: 0.80*** |

В листинге 11 представлена реализация и результат работы наивного байесовского метода.

Листинг 11

|  |
| --- |
| # 11. Метод наивный байесовский  from sklearn.naive\_bayes import GaussianNB  GNB\_model = GaussianNB()  GNB\_model.fit(X\_train, Y\_train)  GNB\_prediction = GNB\_model.predict(X\_test)  print('Матрица несоответствий метода GNB:\n', confusion\_matrix(GNB\_prediction, Y\_test))  GNB\_train\_accuracy = GNB\_model.score(X\_train, Y\_train)  GNB\_test\_accuracy = GNB\_model.score(X\_test, Y\_test)  print ('Точность модели на обучающей выборке: ', round (GNB\_train\_accuracy, 2))  print ('Точность модели на тестовой выборке: ', round (GNB\_test\_accuracy, 2)) |
| **Матрица несоответствий метода *GNB:***  ***[[64 21]***  ***[ 3 11]]***  ***Точность модели на обучающей выборке: 0.77***  ***Точность модели на тестовой выборке: 0.76*** |

В листингах 12 и 13 представлены реализации и результаты работы метода дерева решений и метода опорных векторов соответственно.

Листинг 12

|  |
| --- |
| # 12. Метод дерева решений  from sklearn.tree import DecisionTreeClassifier  DTC\_model = DecisionTreeClassifier()  DTC\_model.fit(X\_train, Y\_train)  DTC\_prediction = DTC\_model.predict(X\_test)  print('Матрица несоответствий метода DTC:\n', confusion\_matrix(DTC\_prediction, Y\_test))  DTC\_train\_accuracy = DTC\_model.score(X\_train, Y\_train)  DTC\_test\_accuracy = DTC\_model.score(X\_test, Y\_test)  print ('Точность модели на обучающей выборке: ', round (DTC\_train\_accuracy, 2))  print ('Точность модели на тестовой выборке: ', round (DTC\_test\_accuracy, 2)) |
| **Матрица несоответствий метода *DTC:***  ***[[55 11]***  ***[12 21]]***  ***Точность модели на обучающей выборке: 1.00***  ***Точность модели на тестовой выборке: 0.77*** |

Листинг 13

|  |
| --- |
| # 13. Метод опорных векторов  from sklearn.svm import SVC  SVC\_model = SVC()  SVC\_model.fit(X\_train, Y\_train)  SVC\_prediction = SVC\_model.predict(X\_test)  print('Матрица несоответствий метода SVC:\n', confusion\_matrix(SVC\_prediction, Y\_test))  SVC\_train\_accuracy = SVC\_model.score(X\_train, Y\_train)  SVC\_test\_accuracy = SVC\_model.score(X\_test, Y\_test)  print ('Точность модели на обучающей выборке: ', round (SVC\_train\_accuracy, 2))  print ('Точность модели на тестовой выборке: ', round (SVC\_test\_accuracy, 2)) |
| **Матрица несоответствий метода *SVC:***  ***[[67 31]***  ***[ 0 1]]***  ***Точность модели на обучающей выборке: 0.72***  ***Точность модели на тестовой выборке: 0.69*** |

По результатам работы других классификаторов можно сказать, что наилучший результат показал метод логистической регрессии, результаты работы которого мы и будем использовать. Например, по результатам матрицы несоответствий, возможно оценить чувствительность и специфичность работы классификатора.

Чувствительность (sensitivity) – это доля положительных классов, которые были определены верно. Высокая чувствительность отражает способность классификатора определять истинно-положительные результаты, однако не отражает число кортежей, которые были неправильно определены как положительные. Чувствительность рассчитывается как отношение истинно-положительных результатов к сумме истинно-положительных и ложноотрицательных результатов:

 (3)

Специфичность (specificity) – это доля отрицательных результатов, которые были определены верно. Высокое значение специфичности указывает на то, что классификатор выдает меньше ложноположительных результатов. Специфичность системы рассчитывается как отношение истинно-отрицательных результатов к сумме истинно-отрицательных и ложноположительных результатов:

 (4)

Так, для результата работы метода логистической регрессии по формулам (3) и (4) получим значение чувствительности и специфичности равные 0,97 и 0,56 соответственно. Другими словами, чувствительность классификатора составляет 97 %, специфичность – 56 %.

Листинг приведенного в данном разделе кода доступен в формате ipython notebook по ссылке: <https://disk.yandex.ru/d/mFcHcKFRRVtNrA?w=1>

# 3. Порядок выполнения работы

В практической части лабораторной работы каждый из студентов реализует алгоритм решения задачи классификации для самостоятельно выбранной базы данных, сервисы для поиска которых приведены в начале раздела практической части.

Этапы выполнения практической части:

1. Выбрать и скачать базу данных, поместив ее в папке рядом с файлом будущей программы.
2. Загрузить базу данных в среду написания алгоритма.
3. Разделить базу данных на вектор признаков и целевую переменную.
4. Нормализовать вектор признаков.
5. Рандомизировать имеющуюся выборку и разделить ее на две составляющие: на обучающую и тестовую выборки.
6. Применить несколько методов для решения задач классификации.
7. Оценить точности модели на обучающей и тестовой выборках и вывести матрицу несоответствий для каждого из использованных методов.
8. Выбрать один из использованных методов на основе оценки значений, полученных на предыдущем этапе. Обосновать выбор.
9. Рассчитать чувствительность и специфичность работы выбранного классификатора.
10. Подготовить отчет по проделанной работе.

# 4. Содержание отчета по лабораторной работе

Отчет по лабораторной работе должен содержать:

1. Краткое описание проведенной работы;
2. Листинг кода;
3. Матрицу несоответствий;
4. Сведенные в таблицу данные о точностях модели на обучающей и тестовой выборках, чувствительность и специфичность работы выбранного классификатора;
5. Выводы о результатах работы выбранного классификатора.

# 5. Контрольные вопросы

1. В чем различия метода обучения с учителем от обучения без учителя?
2. С какой целью применяется понижение размерности данных?
3. В чем заключается особенность метода обучения с подкреплением?
4. Назовите основные методы решения задач классификации.
5. В чем заключается суть метода k-ближайших соседей?
6. Почему наивный байесовский метод называется «наивным»?

# 6. Список использованных источников

1. Turing A. Computing Machinery and Intelligence // Mind. – 1950. – Vol. 59. – 28 p.
2. Samuel A.L. Some Studies in Machine Learning Using the Game of Checkers. // IBM Journal of Research and Development. – 1959. – Vol. 3. – № 3. – 20 p.
3. Kohavi R. Glossary of terms. / R. Kohavi, F. Provost // Machine Learning. – 1998. – Vol. 30. – № 2-3, – 5 p.
4. Nilsson N. Learning Machines // McGraw Hill. – 1965. – 137 p.
5. Duda R. Pattern Recognition and Scene Analysis / R. Duda, P. Hart // Wiley Interscience. – 1973. – 472 p.
6. Bozinovski S. Teaching space: A representation concept for adaptive pattern classification. // Computer and Information Science Department, University of Massachusetts at Amherst, MA. – 1981. – № 81-28. – 56 p.
7. Mitchell T. Machine Learning // McGraw Hill. – 1997. – 2 p.
8. Esteva A. Dermatologist-level classification of skin cancer with deep neural networks. / A. Esteva, B. Kuprel, R. Novoa, et al. // Nature. – 2017. – № 542. – 4 p.
9. DeepMind [электронный ресурс] – URL: <https://deepmind.com/>
10. Рашка С. Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2. / С. Рашка, В. Мирджалили. // СПб.: ООО «Диалектика». – 2020. – 848 с.: ил.
11. Соколов Е.А. Курс лекций по машинному обучению. ФКН ВШЭ / 04.09.2020.
12. Menard S. Logistic Regression : From lntroductory to Advanced Concepts and Applications // ), Sage Publications. – 2009.
13. Дудченко П.В. Метрики оценки классификаторов в задачах медицинской диагностики // Сборник трудов XVI Международной научно-практической конференции студентов, аспирантов и молодых ученых. Томский политехнический университет. – 2019. – 1 c.