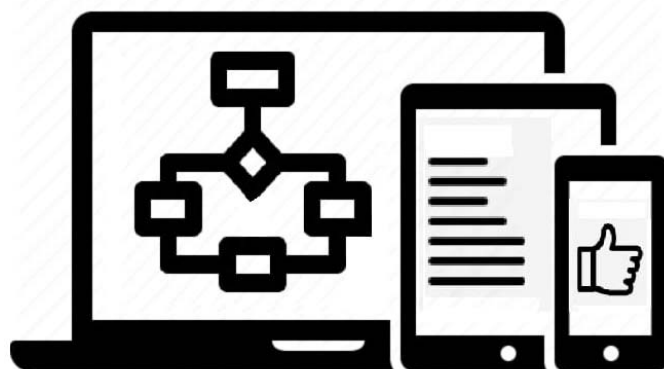


МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

ТЕХНОЛОГИИ РЕШЕНИЯ ЗАДАЧ МАШИННОГО ОБУЧЕНИЯ

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
09.03.01 «Информатика и вычислительная техника»
дневной формы обучения*



Могилев 2026

УДК 004
ББК 32.973
Т38

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «2» марта 2026 г., протокол № 8

Составитель канд. физ.-мат. наук В. А. Ливинская

Рецензент канд. техн. наук В. М. Ковальчук

Методические рекомендации к выполнению лабораторных работ предназначены для студентов направления подготовки 09.03.01 «Информатика и вычислительная техника» дневной формы обучения. Содержат цель, методические указания, задания для самостоятельного выполнения, контрольные вопросы.

Учебное издание

ТЕХНОЛОГИИ РЕШЕНИЯ ЗАДАЧ МАШИННОГО ОБУЧЕНИЯ

Ответственный за выпуск

В. В. Кутузов

Корректор

А. А. Подошевка

Компьютерная верстка

Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.- изд. л. . Тираж 21 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2026

Содержание

Введение.....	4
Лабораторная работа № 1. Основы работы в системе R.....	6
Лабораторная работа № 2. Разведывательный анализ данных.....	8
Лабораторная работа № 3. Проверка гипотез.....	11
Лабораторная работа № 4. Предобработка данных.....	13
Лабораторная работа № 5. Линейная регрессия для прогнозирования непрерывного признака.....	15
Лабораторная работа № 6. Логистическая регрессия для прогнозирования бинарного признака.....	16
Лабораторная работа № 7. Метод главных компонент (РСА) для снижения размерности.....	18
Лабораторная работа № 8. Методы классификации в задачах машинного обучения с учителем.....	19
Лабораторная работа № 9. Метрики качества решения задач классификации.....	21
Лабораторная работа № 10. Методы классификации в задачах машинного обучения без учителя.....	25
Лабораторная работа № 11. Анализ чата группы в Telegram.....	28
Список литературы.....	31
Приложение А. Варианты заданий для генерации двух выборок двумерного нормального распределения.....	32

Введение

Методические рекомендации разработаны в соответствии с программой учебной дисциплины, направленной на подготовку квалифицированных специалистов в области анализа данных (Data Science). Ключевой целью освоения дисциплины является формирование у обучающихся компетенций, необходимых для полного цикла работы с данными: от извлечения информации из больших массивов (Big Data) и их предварительной обработки до глубокого статистического анализа, визуализации результатов и их содержательной интерпретации для принятия управленческих решений.

Техническое обеспечение учебного процесса.

Выполнение лабораторных работ осуществляется в интегрированной среде разработки RStudio на базе языка программирования R (версия не ниже 4.2.0). Для реализации алгоритмов анализа данных и подготовки отчетной документации студентам необходимо использовать следующий набор программных пакетов.

Обработка и анализ данных: tidyverse, dplyr, tidyr, caret, FactoMineR, cluster, e1071, pROC.

Визуализация: ggplot2.

Работа с текстом: tm, tidytext.

Подготовка отчетов и экспорт: knitr, kableExtra, openxlsx, readxl.

Требования к структуре и содержанию отчета.

Отчет по каждой лабораторной работе должен быть выполнен в виде электронного документа (формат PDF или Word), сгенерированного с использованием технологии динамической отчетности R Markdown.

Структура отчета должна строго соответствовать следующему плану.

- 1 Титульный лист установленной формы.
- 2 Цель работы.
- 3 Краткие теоретические сведения (основные понятия и методы).
- 4 Индивидуальное задание.
- 5 Ход выполнения работы (программный код с необходимыми комментариями).
- 6 Результаты исследования (таблицы, графики, расчетные показатели).
- 7 Выводы (интерпретация полученных результатов).
- 8 Ответы на контрольные вопросы.

Требования к оформлению результатов.

Для обеспечения высокого качества полиграфического представления результатов в финальном документе рекомендуется соблюдать следующие правила.

Представление кода. Исходный код программ скрыт в итоговом документе (параметр echo=FALSE в заголовках блоков кода R Markdown), за исключением случаев, когда демонстрация алгоритма предусмотрена заданием.

Табличные данные. Результаты расчетов должны быть представлены в виде форматированных таблиц с использованием пакета kableExtra, обеспечивающего академический стиль оформления.

Графическая визуализация. Все графики должны содержать подписи осей и легенды на русском языке. Для улучшения читаемости размер шрифта элементов графика должен быть увеличен (рекомендуемая настройка: `theme(text = element_text(size = 14))`). При выборе цветовых схем следует отдавать предпочтение специализированным палитрам, адаптированным для научной графики (например, палитра Wong) и обеспечивающим корректное восприятие информации, в том числе лицами с особенностями цветовосприятия.

Описание данных. В разделе хода работы должно присутствовать описание исходных данных, раскрывающее их природу, источник происхождения и структуру.

Автоматизация и воспроизводимость.

Важным аспектом современной аналитики является воспроизводимость исследований. Студентам предписывается использовать шаблонные файлы R Markdown для генерации отчетов, что обеспечивает автоматическое обновление результатов при изменении входных данных или параметров модели. При необходимости промежуточные или итоговые данные должны быть экспортированы в формат Excel для дальнейшего использования в офисных приложениях.

Критерии оценки и защита лабораторных работ.

Защита лабораторной работы проводится в диалоговом режиме и предполагает комплексную проверку следующих аспектов.

1 **Корректность реализации.** Наличие исходного кода программы, реализующего требуемый метод анализа данных без синтаксических и логических ошибок.

2 **Верификация результатов.** Правильность работы программы на заданных наборах данных и соответствие полученных выходных данных ожидаемым теоретическим значениям.

3 **Глубина понимания.** Осознанное понимание студентом сути примененных методов исследования, способности обосновать выбор инструментов и корректно интерпретировать полученные выводы.

4 **Теоретическая подготовка.** Знание ответов на контрольные вопросы, касающиеся теоретических основ темы занятия.

5 **Качество документации.** Наличие отчета, оформленного в строгом соответствии с требованиями к учебным текстовым документам и настоящими методическими указаниями.

Несоблюдение требований к оформлению или невозможность пояснить логику работы кода может являться основанием для снижения итоговой оценки.

Лабораторная работа № 1. Основы работы в системе R

Цель: ознакомление с интерфейсом RStudio; обучение работе в режиме консоли и путем написания скриптов, а также подключением внешних пакетов; изучение основных методов обработки статистических данных; формирование базовых навыков работы в среде R; создание объектов, работа с векторами, факторами, списками, фреймами данных, импорт/экспорт данных.

Теоретические сведения

Начало работы и получение справочной информации.

Для установки среды R и RStudio на свой домашний компьютер воспользуйтесь инструкцией по ссылке https://bdemeshev.github.io/installation/r/R_installation.html.

Директория – это место (папка), где находится ваш «проект», т. е. там лежит скрипт, данные, картинки и пр.

Зачастую, у каждого проекта своя директория, поэтому приходится часто их менять. Есть несколько способов это сделать.

1 Использовать функцию `setwd("~/Desktop/R")`, где в кавычках можно прописать путь к директории.

2 На панели R нажать кнопку Session и в выпадающем списке выбрать Set Working Directory. После этого можно выбрать нужную папку.

3 В нижнем правом окошке есть вся файловая система компьютера, где также удобно установить нужную директорию. Нажав кнопку More, можно перейти в текущую директорию или установить новую.

С целью дальнейшего упрощения работы рекомендуется заранее установить некоторые основные пакеты с помощью функции `install.packages` или, войдя во вкладку Packages/Install Packages, подключить нужный пакет. Для работы чаще всего используются следующие пакеты:

- `psych` – описательная статистика;
- `ggplot2` – графика;
- `lmtest` – тестирование гипотез при построении линейных моделей;
- `MASS` – поиск подходящих распределений;
- `dplyr`, `tidyr` – вычислительные операции с данными.

После загрузки всех необходимых пакетов нужно узнать в какой директории вы находитесь. Это можно сделать с помощью функции `getwd()`.

Импорт данных – это загрузка в среду R различных данных для последующей работы с ними. Важно понимать откуда берём данные. Обычно источником импорта выступают: обычные текстовые файлы (.csv, .txt), Excel-файлы (.xlsx, .xls), базы данных (SQL), интернет, статистические пакеты (SPSS, SAS, STATA).

Импорт файлов с расширениями csv, txt осуществляется с помощью функции `read.csv()`. Формат csv самый распространенный формат хранения данных в мире для анализа данных. Его можно получить, например, сохранив Excel-файл в формате csv (разделитель – запятая).

Проверка того, что все данные импортировались нормально, осуществляется с помощью функции `str()`.

Экспорт данных – это сохранение данных с требуемым расширением на свой компьютер. Для экспорта данных применяют функции `write.csv(df, "table_car.csv")` или `write.xlsx(df, "table_car.xlsx")` или `write.html(df, "table_car.html")`, где `df` – имя фрейма данных в среде R.

Задания

- 1 Инсталлировать пакеты для ввода данных в различных форматах.
- 2 Изучить возможность получать справочную информацию по необходимым пакетам.
- 3 Создать свою директорию.
- 4 Загрузить данные для своего варианта, полученные у преподавателя, в переменную.
- 5 Получить справочную информацию по своим данным, просмотреть их содержимое.
- 6 Проверить, есть ли среди данных пропуски, удалить с помощью соответствующей команды.
- 7 Изучить основные типы данных и основные инструменты работы с ними в R:
 - объекты R: векторы, матрицы, фреймы данных, списки, факторы;
 - операции с данными: индексация, фильтрация, преобразование типов;
 - пакеты `dplyr` и `tidyr` для манипуляции данными.
- 8 Выполнить экспорт очищенных данных в MS Excel.
- 9 Составить отчет.

Контрольные вопросы

- 1 Какие типы данных используются в среде R?
- 2 Как получить справочную информацию по пакетам и встроенным в среду R датасетам?
- 3 Как импортировать в среду R файлы с различными расширениями?
- 4 Как экспортировать из среды R файлы с различными расширениями?
- 5 В чём разница между `list` и `data.frame` в R?
- 6 Как изменить порядок уровней фактора и зачем это нужно?
- 7 Какие функции используются для импорта данных из CSV и Excel?
- 8 Как проверить наличие и тип пропущенных значений?

Лабораторная работа № 2. Разведывательный анализ данных

Цель: проведение простого анализа данных, состоящего из описательной статистики и визуализации.

Теоретические сведения

Разведывательный анализ данных (EDA) предполагает проведение следующих этапов.

1 Расчет описательных статистик: среднее, медиана, квартили, стандартное отклонение.

2 Визуализация: гистограммы, boxplot, scatterplot, jitter-графики.

3 Анализ распределений, выбросов, корреляций.

Основные функции для его проведения представлены в таблице 1.

Таблица 1 – Параметры описательной статистики

Функция	Описание
mean()	Среднее значение
median()	Медиана
var()	Дисперсия
sd()	Стандартное отклонение
min()	Минимальное значение
max()	Максимальное значение
quantile()	Квантили (по умолчанию рассчитываются максимальное и минимальное значения, а также квартили)
summary()	Сводка по параметрам описательной статистики для всех переменных набора

Среди данных могут быть пропуски, которые не позволят рассчитать статистические параметры для столбцов количественных переменных с пропусками без специальных настроек.

Функция `is.na()` возвращает набор данных (вектор, таблицу и т. п.), заполненный значениями `True` (если значение отсутствует) и `False` (при его наличии). Так как `True` маркируется «1», а `False` маркируется «0», то функция `sum()`, примененная к такому преобразованному набору данных, дает число пропусков.

Команда `Glimpse(f)` (данная функция из ранее установленного пакета `dplyr`) позволяет посмотреть краткие сведения о наборе данных. Вывод на экран описательной статистики осуществляется также с помощью функции `describe()`.

Особенность R – возможность создания качественной графики с широким спектром графических возможностей R. На сайте R Graph Gallery (<https://www.r-graph-gallery.com/all-graphs>) представлены не только примеры всевозможных графиков, но и исходный R-код, написанный для их построения.

Для построения графиков используют высокоуровневые функции, доступные без подключения специализированных пакетов для построения визуализаций (таблица 2).

Таблица 2 – Функции для построения графиков

Функция	Описание
plot()	Общая функция для построения графиков. В зависимости от передаваемых данных и настройки параметров определяемая ей визуализация может быть и точечным графиком, и столбчатой диаграммой
boxplot()	График «ящик с усами»
hist()	Гистограмма
barplot()	Столбчатая диаграмма
scatterplot()	Диаграмма рассеяния (точечная)
pie()	Круговая диаграмма

Функция plot() – функция общего назначения, которая используется для построения графиков. С помощью plot() можно создавать диаграммы рассеяния, точечные графики с гладкими линиями и маркерами, точечные графики с отрезками линий и маркерами и др.

Особенность R – возможность создания качественной графики с широким спектром графических возможностей R. На сайте R Graph Gallery (<https://www.r-graph-gallery.com/all-graphs>) представлены не только примеры всевозможных графиков, но и исходный R-код, написанный для их построения.

Для построения графиков используют высокоуровневые функции, доступные без подключения специализированных пакетов для построения визуализаций (см. таблицу 2).

В технологиях машинного обучения в R наибольшей популярностью пользуется библиотека ggplot2, которая реализует концепцию Grammar of Graphics (Leland Wilkinson), согласно которой любой график строится из независимых слоёв:

График = Данные + Эстетика + Геометрия + Статистика +
+ Координаты + Фасеты + Тема.

В таблице 3 представлена концепция Grammar of Graphics (Leland Wilkinson). В таблице 4 представлены основные геометрии библиотеки ggplot2.

Ключевые преимущества использования библиотеки.

Модульность – график собирается поэтапно, как конструктор.

Согласованность синтаксиса – единый подход ко всем типам графиков.

Автоматическая легенда – не нужно вручную подписывать группы.

Интеграция с tidyverse – работает с dplyr, tidyr, readr.

Гибкое форматирование – полный контроль над шрифтами, цветами, осями.

Экспорт в высоком качестве – ggsave() поддерживает PNG, PDF, SVG, TIFF.

Доступность – поддержка цветовых схем для дальтоников (например,

палитра Wong).

Таблица 3 – Концепция Grammar of Graphics библиотеки ggplot2

Компонент	Функция в ggplot2	Пример
Данные	<code>data = ...</code>	<code>data = df</code>
Эстетика	<code>aes()</code>	<code>aes(x = вес, y = рост, color = пол)</code>
Геометрия	<code>geom_*()</code>	<code>geom_point()</code> , <code>geom_line()</code>
Статистика	<code>stat_*()</code>	<code>stat_smooth()</code> , <code>stat_summary()</code>
Координаты	<code>coord_*()</code>	<code>coord_flip()</code> , <code>coord_fixed()</code>
Фасеты	<code>facet_*()</code>	<code>facet_wrap(~группа)</code>
Тема	<code>theme_*()</code>	<code>theme_minimal()</code> , <code>theme_bw()</code>

Таблица 4 – Основные геометрии (`geom_*`)

Геометрия	Назначение	Пример использования
<code>geom_point()</code>	Точечная диаграмма	Сравнение двух непрерывных переменных
<code>geom_line()</code>	Линейный график	Временные ряды, тренды
<code>geom_bar()/geom_col()</code>	Столбчатая диаграмма	Категориальные данные, частоты
<code>geom_histogram()</code>	Гистограмма	Распределение непрерывной переменной
<code>geom_boxplot()</code>	Boxplot	Сравнение распределений по группам
<code>geom_violin()</code>	Violin plot	Плотность распределения по группам
<code>geom_density()</code>	Плотность распределения	Гладкая оценка распределения
<code>geom_errorbar()</code>	Погрешности	Доверительные интервалы
<code>geom_text()/geom_label()</code>	Подписи	Аннотации на графике
<code>geom_tile()/geom_raster()</code>	Тепловая карта	Матрицы корреляций, геоданные

Задания

- 1 Загрузить набор данных, полученных у преподавателя.
- 2 Вычислить описательные статистики по числовым переменным (`summary()`, `describe()`).
- 3 Построить:
 - гистограммы с наложением плотности (`geom_histogram(aes(y=..density..)) + geom_density()`);
 - boxplot для сравнения групп (параметр `fill` + `scale_fill_manual` с палитрой Wong);
 - scatterplot с регрессионной линией (`geom_smooth(method="lm")`).
- 4 Проанализировать корреляционную матрицу (`cor()`, `corrplot`).
- 5 Выявить выбросы с помощью правила $1,5 \times IQR$ и визуализировать их на boxplot.

6 Сгруппировать данные по категориальному признаку (например, «Область») и вычислить статистику по группам (`group_by() %>% summarise()`).

Контрольные вопросы

- 1 Зачем нужен разведывательный анализ перед построением моделей?
- 2 Как интерпретировать `boxplot` и что показывают усы?
- 3 В чём разница между средним и медианой при наличии выбросов?
- 4 Как визуально оценить нормальность распределения?
- 5 Какие существуют методы предобработки данных (удаление дубликатов, обработка пропусков, выявление аномальных наблюдений)?
- 6 Как визуализировать распределение количественных данных в R?
- 7 Как визуализировать распределение категориальных данных в R?

Лабораторная работа № 3. Проверка гипотез

Цель: изучение методов проверки гипотез для сравнения групп согласно алгоритму; научиться делать выводы по результатам статистической проверки.

Теоретические сведения

Статистическая гипотеза — это формализуемое предположение о виде неизвестного распределения или о значениях параметров распределения генеральной совокупности, истинность которого проверяется на основе выборочных данных с помощью статистического критерия с определённой вероятностью ошибки.

Структура проверки гипотез.

Всегда формулируются две взаимно исключающие гипотезы.

1 Нулевая гипотеза (H_0) — основное предположение, которое считается верным, пока не доказано обратное. Обычно утверждает отсутствие эффекта, различий или связи.

Пример: $\mu_1 = \mu_2$ (средние двух групп равны).

2 Альтернативная гипотеза (H_1 или H_a) — утверждение, которое принимается, если нулевая гипотеза отклоняется. Утверждает наличие эффекта, различий или связи.

Пример: $\mu_1 \neq \mu_2$ (средние двух групп различаются).

Проверка статистической гипотезы не доказывает истинность гипотезы на 100 %. Она позволяет лишь сделать вывод о том, насколько согласуются данные с гипотезой.

1 «Отклонить H_0 » — данные противоречат нулевой гипотезе (есть статистически значимый эффект).

2 «Не отклонить H_0 » — данных недостаточно, чтобы отвергнуть нулевую гипотезу (но это не значит, что она истинна).

Основные тесты, использующиеся в машинном обучении.

Параметрические тесты: t-критерий Стьюдента (одновыборочный, двухвыборочный, парный).

Непараметрические тесты: Манна – Уитни, Вилкоксона, Бруннера – Мюнцеля (при неравенстве дисперсий).

Тесты для категориальных данных: χ^2 , точный тест Фишера.

Проверка предпосылок: нормальность (Шапиро – Уилко), равенство дисперсий (Левена, F-тест).

Алгоритм проверки статистических гипотез

1 По выборочным данным формулируют основную H_0 и альтернативную H_1 гипотезы.

2 Задают уровень значимости α (0,05 или 0,01).

3 В зависимости от H_0 определяют статистический критерий K , имеющий известное распределение.

4 По выборке и формуле критерия K рассчитывают наблюдаемое значение критерия $K_{набл}$.

5 В зависимости от вида H_1 определяют вид критической области W и критические точки по соответствующим таблицам для распределения критерия K .

6 По результатам проверки принадлежности $K_{набл}$ к критической области делают вывод о принятии или отклонении гипотезы H_0 . Формулируют общий вывод, исходя из поставленной задачи.

В зависимости от имеющейся у исследователя информации, гипотезы классифицируются по таблицам 5 и 6.

Таблица 5 – Функции для тестирования параметрических гипотез

Функция	Описание
Одновыборочные критерии о равенстве числовому параметру	
t. test (x, μ_0)	Математического ожидания
Критерии о равенстве числовых характеристик	
var.test (x, y)	Дисперсий двух групп
t. test (x, y)	Математических ожиданий двух групп
prop.test(x, y)	Долей двух групп
aov()	Математических ожиданий в нескольких группах
bartlett.test ()	Дисперсий в нескольких группах

Таблица 6 – Функции для тестирования непараметрических гипотез

Функция	Описание
Проверка гипотезы о законе распределения	
shapiro.test(x)	О согласии с нормальным законом распределения
Критерии об однородности	
wilcox.test(x,y,paired=F)	Критерий Манна – Уитни для двух независимых выборок
wilcox.test(x,y,paired=T)	Критерий Вилкоксона для двух зависимых выборок
kruskal.test()	Критерий Краслелла – Уоллеса о равенстве распределений в нескольких группах

Задания

- 1 Получить данные у преподавателя.
- 2 Сформулировать нулевую и альтернативную гипотезы.
- 3 Проверить нормальность распределения переменной (shapiro.test()).
- 4 Сравнить средние двух независимых групп:
 - при выполнении предпосылок t.test(..., var.equal = TRUE);
 - при нарушении: t.test(..., var.equal = FALSE) или wilcox.test().
- 5 Применить критерий Бруннера – Мюнцеля (пакет BrunnerMunzel) при значимом различии дисперсий.
- 6 Проанализировать связь двух категориальных переменных с помощью chisq.test() и fisher.test().
- 7 Визуализировать результат.
- 8 Интерпретировать p-value, доверительные интервалы.

Контрольные вопросы

- 1 Когда следует использовать непараметрические тесты?
- 2 Как интерпретировать результат критерия Бруннера – Мюнцеля?
- 3 В чём разница между статистической и практической значимостью?
- 4 Как проверить предпосылки t-теста?

Лабораторная работа № 4. Предобработка данных

Цель: освоение методов очистки, трансформации и подготовки данных для машинного обучения.

Теоретические сведения

К основным методам предобработки данных в R относятся следующие.

- 1 Обработка пропусков: удаление, импутация (среднее, медиана, k -NN) (критерии использования представлены в таблице 7).
- 2 Масштабирование: стандартизация (scale()), нормализация (min-max).

3 Преобразование формата данных: `pivot_longer()`, `pivot_wider()` из `tidyr`.

Таблица 7 – Основные методы обработки пропусков

Метод	Использование	Преимущество
Импутация средним	Нормальное распределение, нет выбросов	Быстро, просто
Импутация медианой	Есть выбросы, асимметричное распределение	Устойчив к экстремальным значениям
k -NN	Сложные зависимости между переменными	Учитывает взаимосвязи признаков

Масштабирование: стандартизация (`scale()`), нормализация (`min-max`) необходимо для работы с данными, имеющие разные единицы измерения и масштабы.

Для масштабирования числовых признаков используются различные подходы.

1 **Стандартизация *Z-score normalization*** – сохраняет форму распределения, обрабатывает выбросы, приводит данные к распределению со средним равным нулю и стандартным отклонением равным 1.

2 **min/max-нормализация** – приводит данные к диапазону $[0, 1]$, используется при работе с нейронными сетями, когда важен фиксированный диапазон.

Преобразование формата данных (`tidyr`).

В зависимости от поставленной задачи используют следующие инструменты:

- `pivot_longer()` – преобразование из широкого в длинный формат для визуализации в `ggplot2` (требуется длинный формат);
- `group_by()` – для групповых операций, для анализа временных рядов.

Задания

1 Загрузить «грязный» набор данных (с пропусками, дубликатами, некорректными типами).

2 Выполнить:

- удаление дубликатов (`distinct()`);
- обработку пропусков: визуализация (`vismiss`), импутация медианой/средним;
- преобразование типов: `as.factor()`, `as.numeric()`.

3 Закодировать категориальные переменные:

- бинарные: `ifelse()`;
- множественные: `model.matrix()` или `caret::dummyVars()`.

4 Стандартизировать числовые признаки (`scale()`).

5 Преобразовать данные из широкого в длинный формат для визуализации (`pivot_longer()`).

Контрольные вопросы

- 1 Почему важно обрабатывать пропуски перед моделированием?
- 2 В чём разница между стандартизацией и нормализацией?
- 3 Когда использовать one-hot, а когда label encoding?
- 4 Как pivot_longer() помогает в визуализации?

Лабораторная работа № 5. Линейная регрессия для прогнозирования непрерывного признака

Цель: построение и интерпретация модели множественной линейной регрессии для задачи прогнозирования.

Теоретические сведения

Уравнение регрессии – функция, позволяющая по величине изменения одного коррелируемого признака определить среднюю величину другого признака.

Модель: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \varepsilon$.

Выделим основные этапы регрессионного анализа.

Первый этап. Предположение – происходит выбор формы связи между переменными (модель).

Второй этап. Параметризация – происходит оценка значений параметра в выбранной формуле статистической связи. Форма связи (функция) линейная, нелинейная.

Третий этап. Проверка надёжности полученных оценок – осуществляются следующие тесты: *F*-тест (проверка статистической значимости выбранной формы связи), *t*-тест (проверка статистической значимости найденных числовых значений параметра).

Четвертый этап. Проверка предпосылок выполнимости условий Гаусса – Маркова: линейность, гомоскедастичность, нормальность остатков, отсутствие мультиколлинеарности (VIF).

В результате анализа статистических данных, выбора и построения модели последовательно выполняются все четыре этапа.

В регрессионном анализе в машинном обучении для оценки качества предсказаний применяется несколько метрик.

1 Средняя абсолютная ошибка (Mean Absolute Error, MAE) – представляет собой среднее абсолютное значение разности между фактическими и предсказанными значениями.

2 Среднеквадратичная ошибка (Mean Squared Error, MSE) – измеряет среднее значение квадрата разности между фактическими и предсказанными значениями.

3 Корень из среднеквадратичной ошибки (Root Mean Squared Error, RMSE) –

представляет собой квадратный корень из MSE, является интерпретируемой метрикой, измеряющей среднее абсолютное отклонение от модели.

4 Коэффициент детерминации (R-squared, R^2) – измеряет долю дисперсии зависимой переменной, которая объясняется моделью. Показывает, насколько хорошо модель соответствует данным. Значения R^2 лежат в диапазоне от 0 до 1, где 1 указывает на идеальное соответствие.

Задания

- 1 Получить данные у преподавателя.
- 2 Разделить данные на обучающую и тестовую выборки `caret::createDataPartition()`.
- 3 Построить модель: `lm(Y ~ X1 + X2 + Xk, data=train)`.
- 4 Проанализировать выводы `summary()`: значимость коэффициентов, R^2 .
- 5 Проверить предпосылки МНК построением:
 - графика остатков vs предсказанные значения;
 - гистограммы остатков;
 - qq-plot.
- 6 Протестировать мультиколлинеарность VIF (`car::vif()`).
- 7 Сделать прогноз на тестовой выборке, вычислите RMSE, MAE.
- 8 Визуализировать фактические и предсказанные значения (`ggplot2`).

Контрольные вопросы

- 1 Как интерпретировать коэффициент регрессии при фиксированных остальных переменных?
- 2 Что показывает скорректированный R^2 ?
- 3 Как обнаружить мультиколлинеарность и чем она опасна?
- 4 Почему важно проверять остатки модели?

Лабораторная работа № 6. Логистическая регрессия для прогнозирования бинарного признака

Цель: построение модели логистической регрессии для классификации и интерпретация результатов.

Теоретические сведения

Задача классификации в машинном обучении – это задача отнесения объекта к одному из заранее определенных классов на основании его формализованных признаков. Каждый из объектов в этой задаче представляется в виде вектора в N -мерном пространстве, каждое измерение в котором представляет собой описание одного из признаков объекта. Для обучения классификатора необходимо иметь набор объектов, для которых заранее определены классы. Это

множество называется обучающей выборкой, её разметка производится вручную, с привлечением специалистов в исследуемой области.

Логистическая регрессия – это алгоритм классификации машинного обучения, используемый для прогнозирования вероятности категориальной зависимой переменной.

Логистическая регрессия требует больших размеров выборки.

Логистическая регрессия применяется для прогнозирования вероятности возникновения некоторого события по значениям множества признаков. Для этого вводится так называемая зависимая переменная y , принимающая лишь одно из двух значений – как правило, это числа 0 (событие не произошло) и 1 (событие произошло), и множество независимых переменных (также называемых признаками, предикторами или регрессорами) – вещественных $x_1, x_2, \dots, x_n$, на основе значений которых требуется вычислить вероятность принятия того или иного значения зависимой переменной.

Логистическая функция имеет вид

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots)}} \quad \text{или} \quad P(Y = 1) = \frac{1}{1 + e^{-Z}},$$

где Z – линейная комбинация независимых переменных.

Подбор коэффициентов β_i линейной комбинации Z осуществляется на обучающей выборке с помощью метода максимального правдоподобия.

Ключевые термины для логистической регрессии:

- 1) логит (logit) – функция, которая отображает вероятность принадлежности классу в интервале от $-\infty$ до $+\infty$ (вместо интервала от 0 до 1);
- 2) отношение шансов (odds) – отношение «успеха» (1) к «неуспеху» (0);
- 3) логарифм отношения шансов (log odds) – результат в преобразованной модели (в виде линейной комбинации), который отображается назад в вероятность (odds ratio = e^{β_i}).

Задания

- 1 Получить данные у преподавателя
- 2 Подготовить бинарную целевую переменную (например, «Исход»: 0/1).
- 3 Построить модель: `glm(Исход ~ ., family=binomial, data=train)`.
- 4 Проанализировать `summary()`: значимость коэффициентов, направление влияния.
- 5 Вычислить odds ratio и доверительные интервалы (`exp(coef())`, `confint()`).
- 6 Сделать прогноз вероятностей (`predict(type="response")`) и классов (порог 0,5).

Контрольные вопросы

- 1 Почему в логистической регрессии используется логит-преобразование?
- 2 Как интерпретировать odds ratio > 1 и < 1 ?
- 3 В чём разница между точностью и полнотой?
- 4 Как выбрать оптимальный порог классификации?

Лабораторная работа № 7. Метод главных компонент (РСА) для снижения размерности

Цель: освоение РСА для визуализации многомерных данных и снижения размерности; найти меньшее число новых признаков, сохраняющих максимальную информацию.

Теоретические сведения

Постановка задачи уменьшения размерности: с ростом числа признаков экспоненциально растёт объём данных, необходимых для обучения.

Метод главных компонент (РСА) – это статистический метод снижения размерности данных, который преобразует исходные коррелированные переменные в новые ортогональные главные компоненты, сохраняя максимум дисперсии. Первая компонента захватывает наибольшую вариацию данных, вторая – следующую и т. д., что позволяет упростить анализ без значительной потери информации.

Данные сначала центрируются (вычитается среднее), затем применяется разложение на сингулярные значения (SVD) или вычисление собственных векторов ковариационной матрицы. Компоненты ранжируются по убыванию объясняемой дисперсии; обычно выбирают те, что покрывают 80 %...95 % общей вариации.

В объекте `prcomp()` в R компонент `rotation` содержит `loadings` (нагрузки), а компонент `x` – `scores`. Векторы `loadings` показывают корреляции переменных с главными компонентами.

Как выбрать количество главных компонент?

Количество главных компонент в РСА выбирают так, чтобы сохранить ключевую информацию данных, отбросив шум.

Критерий кумулятивной дисперсии предполагает выбор минимального числа РС, объясняющих 80 %...95 % общей дисперсии (`cumulative proportion`).

Задания

- 1 Получить данные у преподавателя согласно варианту
- 2 Стандартизировать данные (РСА чувствителен к масштабу).
- 3 Выполнить РСА: `prcomp(..., scale=TRUE)` или `FactoMineR::PCA()`.
- 4 Проанализировать:
 - долю объяснённой дисперсии по компонентам;
 - каменистое осыпание (`screeplot()`);
 - нагрузки первых двух компонент (тепловая карта).
- 5 Визуализировать объекты в пространстве PC_1 , PC_2 с раскраской по группам (`ggplot2`).
- 6 При необходимости сократить размерность до двух-трех компонент и сохранить преобразованные данные.

7 Интерпретировать, какие исходные переменные вносят наибольший вклад в главные компоненты.

Контрольные вопросы

- 1 Зачем стандартизировать данные перед PCA?
- 2 Как определить оптимальное число главных компонент?
- 3 Что показывают нагрузки (loadings) и как их интерпретировать?
- 4 В каких задачах PCA наиболее эффективен?

Лабораторная работа № 8. Методы классификации в задачах машинного обучения с учителем

Цель: изучение теоретических основ алгоритмов классификации: деревьев решений и метод k -ближайших соседей; освоение практических навыков построения моделей классификации в R; обучение настраиванию гиперпараметров моделей и оценивание их качества; сравнение эффективности двух алгоритмов на реальных данных.

Теоретические сведения

Постановка задачи классификации. По обучающей выборке построить модель, предсказывающую класс нового объекта.

Алгоритм CART (Classification and Regression Trees) представляет собой метод построения двоичных деревьев решений для задач классификации и регрессии.

Цель алгоритма: разбить данные на подмножества таким образом, чтобы в каждом подмножестве объекты были как можно более однородными с точки зрения целевой переменной.

На каждом шаге алгоритм выбирает переменную и значение, чтобы разделить данные на два подмножества. Выбор происходит на основе критерия неоднородности, такого как критерий Джини для задач классификации или критерий наименьших квадратов для задач регрессии. Этот процесс повторяется рекурсивно для каждого подмножества, пока не выполнится условие останова.

Дерево строится до тех пор, пока не будет достигнуто определенное количество объектов в листовых узлах, или до тех пор, пока нет возможности провести дальнейшее разделение. В таблице 8 представлены преимущества и недостатки алгоритма.

Таблица 8 – Преимущества и недостатки алгоритма CART

Преимущество	Недостаток
Интерпретируемость (правила «если-то»)	Склонность к переобучению
Не требует масштабирования данных	Нестабильность (малые изменения данных $\rightarrow$ новое дерево)
Работает с числовыми и категориальными признаками	Предпочтение признакам с большим числом значений
Не требует предположений о распределении	Границы решений параллельны осям координат

Алгоритм построения дерева (CART).

- 1 Начать с корневого узла (вся выборка).
- 2 Для каждого признака вычислить прирост информации (IG или ΔGini).
- 3 Выбрать признак с максимальным приростом.
- 4 Разбить узел на дочерние узлы.
- 5 Рекурсивно повторять для каждого дочернего узла.
- 6 Остановиться, если:
 - все объекты в узле принадлежат одному классу;
 - достигнута максимальная глубина;
 - в узле меньше min_samples объектов.

Метод k -ближайших соседей (k -NN) – объект классифицируется по большинству классов среди k ближайших объектов обучающей выборки с использованием одной из известных метрик расстояния.

Евклидово расстояние

$$d(x, x') = \sqrt{\sum_{j=1}^p (x_j - x'_j)^2}.$$

Манхэттенское расстояние

$$d(x, x') = \sqrt{\sum_{j=1}^p |x_j - x'_j|}.$$

- 1 Для нового объекта x вычислить расстояния до всех объектов обучающей выборки.
- 2 Выбрать k -ближайших соседей.
- 3 Определить классы выбранных соседей.
- 4 Присвоить класс, встречающийся чаще всего (мажоритарное голосование).

Задания

- 1 Получив данные своего варианта у преподавателя, провести классификацию объектов методом CART, выполнив настройку параметров модели и оценить ее производительность на обучающем и тестовом наборах данных.
- 2 Визуализировать полученное дерево решений для наглядного представления классификационных правил.

3 Сформулировать выводы о производительности и применимости классификационных моделей на основе деревьев решений и решающих правил.

4 Построить модель k -NN для разного количества соседей, выбрать оптимальное k .

5 Сравнить качество моделей.

6 Оформить отчет.

Контрольные вопросы

1 В чём разница между «жадным» и «ленивым» обучением? Приведите примеры алгоритмов каждого типа.

2 Как влияет параметр k на качество модели k -NN? Что происходит при слишком малом и слишком большом k ?

3 Почему для k -NN обязательно требуется масштабирование данных, а для деревьев решений – нет?

4 Какие критерии разбиения используются в деревьях решений? В чём разница между энтропией и индексом Джини?

5 Как интерпретировать правила, извлечённые из дерева решений? Приведите пример правила из построенной модели.

6 Что такое переобучение в контексте деревьев решений? Какие методы регуляризации дерева вы знаете?

7 Как выбрать оптимальное значение k для k -NN? Какие методы кросс-валидации можно использовать?

8 В каких задачах предпочтительнее использовать деревья решений, а в каких – k -NN? Обоснуйте ответ.

9 Как обрабатывать пропущенные значения перед применением k -NN? Почему это критично для данного алгоритма?

Лабораторная работа № 9. Метрики качества решения задач классификации

Цель: изучение метрик для объективной оценки и сравнения моделей классификации.

Теоретические сведения

Когда речь заходит о сравнении методов классификации в R, важно учитывать несколько ключевых аспектов. Сравнение методов классификации в R осуществляется на основе нескольких факторов, включая производительность модели, удобство использования, способность к обобщению, требования к предварительной обработке данных и т. д. К широко используемым инструментам сравнения качества предсказания класса в R различными методами относятся: кросс-валидация для оценки производительности модели, а также

сравнение их метрик качества, точность, полнота, F-мера или ROC-кривая. Необходимо также учитывать интерпретируемость моделей, а также их способность обобщения для новых данных.

Большой выбор методов классификации в R дает возможность подобрать подходящий метод в зависимости от конкретной задачи и особенностей данных.

Задача алгоритма классификации состоит в том, чтобы относить ранее неизвестные объекты к тому или иному классу.

Результатом классификации могут быть четыре варианта:

1) истинно-положительный результат (True Positive, TP) – предсказано True, в действительности True;

2) ложноположительный результат (False Positive, FP) – предсказано True, в действительности False;

3) истинно отрицательный результат (True Negative, TN) – предсказано False, в действительности False;

4) ложноотрицательный результат (False Negative, FN) – предсказано False, в действительности True.

Для численного представления качества оценки в R используют матрицу ошибок (confusion matrix). Матрица ошибок в машинном обучении (таблица 9) – это таблица, которая показывает количество правильных и неправильных предсказаний модели, в которой по строкам указываются реальные значения (истинные классы), а по столбцам – предсказанные значения (прогнозы модели).

Таблица 9 – Матрица ошибок

Реальное значение	Предсказано: 0	Предсказано: 1
Фактически: 0	TN	FP
Фактически: 1	FN	TP

В таблице 9 принято следующее:

- TP (True Positive) – истинно положительные;
- TN (True Negative) – истинно отрицательные;
- FP (False Positive) – ложноположительные;
- FN (False Negative) – ложноотрицательные.

Три из основных метрик, которые широко используются для оценки качества работы классификационных моделей, включают точность (Precision), полноту (Recall) и F-меру (F1-score).

1 Точность измеряет сколько из объектов, которые модель отнесла к положительному классу, действительно принадлежат к положительному классу. Она определяется как отношение числа верно классифицированных положительных объектов к общему количеству объектов, которые модель отнесла к положительному классу.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}).$$

2 Полнота измеряет, сколько из общего числа реальных положительных

объектов модель успешно обнаружила. Она определяется как отношение числа верно классифицированных положительных объектов к общему количеству реальных положительных объектов.

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}).$$

3 F-мера представляет собой сбалансированную метрику, которая учитывает и точность, и полноту. Она является гармоническим средним между точностью и полнотой.

$$\text{F1-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}).$$

Точность и полнота обычно работают в противоположных направлениях: увеличение точности может привести к снижению полноты и наоборот. F-мера учитывает обе эти метрики и представляет собой компромисс между точностью и полнотой. F-мера особенно полезна в случаях, когда классы несбалансированы, т. е. когда количество объектов в одном классе существенно превышает количество объектов в другом.

Эти метрики помогают в оценке качества работы классификационных моделей и помогают понять, насколько хорошо модель способна правильно классифицировать положительные и отрицательные объекты.

ROC-анализ (Receiver Operating Characteristic) – это метод оценки качества модели классификации, основанный на анализе кривой зависимости истинно положительных результатов от ложноположительных. ROC-анализ используется для сравнения методов классификации и определения оптимального порога для принятия решений в задаче логистической регрессии.

Критерий AUC-ROC устойчив к несбалансированным классам и может быть интерпретирован как вероятность того, что случайно выбранный positive-объект будет проранжирован классификатором выше (будет иметь более высокую вероятность быть positive), чем случайно выбранный negative-объект.

Для оценки качества логистической модели строится матрица ошибок, рассчитываются коэффициенты чувствительности и специфичности, которые варьируются в зависимости от порога отсечения.

Результатом логистической регрессии является вероятность того, что событие произойдет, но чтобы сделать предсказание нам необходимо определить пороговое (threshold) значение t , такое что:

$$P(y = 1) \geq t, \text{ событие произошло};$$

$$P(y = 1) < t, \text{ событие не произошло}.$$

При высоком пороговом значении модель редко будет предсказывать положительный результат (только при высокой вероятности $P(y=1)$). И наоборот, при низком пороговом значении модель будет чаще предсказывать положительный исход и реже отрицательный. Матрица ошибок также может использоваться для настройки модели и выбора оптимального порога принятия

решений.

ROC-кривая строится на графике, где по оси абсцисс откладываются значения ложноположительных результатов, а по оси ординат – истинно положительные результаты. Каждой точке на кривой соответствует определенный порог принятия решения. Чем ближе кривая к верхнему левому углу, тем лучше модель классификации. Отсюда можно получить два значения, которые помогут нам определить какие ошибки делает модель.

Чувствительность – доля верно предсказанных позитивных исходов (True Positive rate)

$$\text{Sensitivity} = \text{TP} / (\text{TP} + \text{FN}).$$

Специфичность – доля верно предсказанных негативных исходов (True Negative rate).

$$\text{Specificity} = \text{TN} / (\text{TN} + \text{FP}).$$

Модель с более высоким пороговым значением будет иметь более высокую чувствительность и низкую специфичность. Модель с низким пороговым значением наоборот. Если предпочтений нет, можно оставить пороговое значение $t = 0,5$.

Для сравнения качества классификации нескольких методов используют площадь под ROC-кривой (AUC).

Интерпретация AUC:

AUC = 0,5 – случайное угадывание;

0,5 < AUC < 0,7 – низкое качество;

0,7 ≤ AUC < 0,9 – хорошее качество;

AUC ≥ 0,9 – отличное качество;

AUC = 1,0 – идеальная модель.

Выбор метрик в зависимости от задачи.

Для сбалансированных классов достаточно использовать Accuracy, для несбалансированных – F1.

В задачах медицинской диагностики, когда важно не пропустить болезнь рекомендуется в качестве метрики качества применять Precision-Recall и AUC.

При конструировании спам-фильтра, когда важно не пометить важное письмо как спам принято рассчитывать Precision.

Задания

1 По результатам лабораторных работ № 6 и 8 построить матрицу ошибок для прогнозов модели (table() или caret::confusionMatrix()).

2 Вычислить вручную и с помощью функций: Accuracy, Precision, Recall, F1.

3 Построить ROC-кривую и вычислить AUC (pROC::roc()).

4 Проанализировать кривую «точность–полнота» (PRROC::pr.curve()).

5 Исследовать влияние порога классификации на метрики (график метрик от порога).

6 Сравнить две модели по нескольким метрикам и обосновать выбор лучшей.

Контрольные вопросы

- 1 Когда точность (ассурасу) – ненадёжная метрика?
- 2 В каких задачах важнее полнота, а в каких – точность?
- 3 Что показывает площадь под ROC-кривой (AUC)?
- 4 Как интерпретировать F1-меру?

Лабораторная работа № 10. Методы классификации в задачах машинного обучения без учителя

Цель: выработка практических навыков проведения многомерной классификации методами кластерного анализа средствами R и последующего анализа результатов.

Теоретические сведения

Постановка задачи кластеризации.

Пусть имеется выборка U из n объектов наблюдения, каждый из которых характеризуется m числовыми признаками X_j . Известно, что каждый из этих объектов на самом деле относится к одному из k классов, причём признаки объектов из одного класса не слишком сильно различаются, а признаки объектов из разных классов различаются более существенно. Мера различия между признаками объектов может быть определена как расстояние $\rho(x, y)$ между векторами признаков в соответствующем признаковом пространстве.

Алгоритм k внутригрупповых средних.

Идея алгоритма k внутригрупповых средних (англ. *kmeans*) заключается в последовательном пересчёте внутригрупповых средних. Пусть вначале для каждого из k кластеров имеется некоторое начальное внутригрупповое среднее $\bar{x}_i$. Если внутригрупповые средние заданы, то каждый кластер U_i очевидным образом определяется, как множество объектов наблюдения, вектора признаков для которых ближе к центру, чем к центрам других кластеров. После этого внутригрупповое среднее можно пересчитать, затем снова пересчитать кластеры, пока средние не перестанут меняться.

В языке R для кластеризации методом k внутригрупповых средних используется функция *kmeans()* из пакета *stats*. Она позволяет производить кластеризацию данных различными методами.

Функция возвращает объект класса *kmeans*, который имеет методы *print* и *fitted*. Этот объект представляет собой список, содержащий следующие элементы:

cluster – вектор из n целых положительных чисел, обозначающих номер

кластера соответствующего объекта наблюдения;

centers – матрица, строки которой представляют собой центры соответствующих кластеров;

totss – сумма квадратов расстояний от объектов наблюдения до соответствующих центров кластеров;

withinss – вектор внутрикластерных сумм квадратов расстояний между объектами одного кластера для каждого кластера;

tot.withinss – общая сумма внутрикластерных сумм квадратов расстояний между объектами, т. е. сумма элементов вектора withinss;

betweenss – междукластерная сумма квадратов расстояний, т. е. разница между totss и tot.withinss;

size – количество объектов наблюдения, попавших в каждый из кластеров;

iter – количество внешних итераций, совершённых в ходе работы алгоритма;

ifault – код ошибки, возникшей в ходе работы алгоритма.

Доступ к этим значениям можно получить путём записи их названия в двойных квадратных скобках справа от переменной, содержащей модель, либо через знак доллара. Например, «a[["cluster"]]» или «a\$cluster».

Иерархическая кластеризация.

Часто задача кластеризации возникает в несколько более сложной постановке: точное число кластеров k может быть неизвестно. В этом случае использование алгоритма k внутригрупповых средних не представляется возможным. Конечно, можно просто перебирать значения k и для каждого из них использовать этот алгоритм, но сравнение среднеквадратических расстояний до центров кластеров для различных количеств кластеров лишено смысла.

Вместо этого используются алгоритмы иерархической кластеризации (англ. *Hierarchical clustering*), которые разбивают множество объектов наблюдения на кластеры, которые, в свою очередь, также разбиваются на кластеры и т. д. Получившийся граф, вершинами которого являются кластеры, а дуги направлены от кластеров более высокого уровня к соответствующим кластерам более низкого уровня, называется дендрограммой.

Построенная дендрограмма содержит информацию о близости между отдельными подмножествами объектов наблюдения, что позволяет подбирать число кластеров и производить более тщательный кластерный анализ в отдельных случаях. Иерархическая кластеризация, как и обыкновенная, производится на основе расстояний $\rho(x, y)$ между объектами наблюдения из заданной выборки. Эти расстояния также могут быть определены по-разному. На их основе строятся более сложные показатели качества кластеризации, которые допускают сравнение для различного числа кластеров. Далее с помощью различных методов, основанных на последовательном объединении и разбиении имеющихся кластеров, строится дендрограмма.

Существует ряд различных методов иерархической кластеризации:

- 1 Метод Уорда (англ. Ward's method).
- 2 Метод одиночной связи (англ. single linkage).
- 3 Метод полной связи (англ. complete linkage).
- 4 Метод средней связи (англ. pair-group method using arithmetic averages).

Задания

Входные данные: n объектов, каждый из которых характеризуется двумя числовыми признаками $\{x_i\}$ и $\{y_i\}$. Требуется исследовать работу алгоритмов кластеризации объектов наблюдения по двум признакам. Для каждого набора данных требуется выполнить следующие задания.

1 Провести кластеризацию объектов наблюдения с помощью алгоритма k внутригрупповых средних. Выбрать оптимальное количество кластеров, исходя из критерия минимизации внутригрупповых дисперсий и максимизации расстояния между центрами кластеров.

2 Графически изобразить на плоскости разбиения объектов наблюдения в соответствии с кластерами и в соответствии с классами c_i . Также отметить центры каждого кластера. Количество кластеров должно соответствовать количеству классов.

3 Для разбиения на кластеры вычислить сумму квадратов расстояний от каждого объекта наблюдения до центра соответствующего кластера.

4 Провести кластеризацию исходных данных иерархическим способом. Сравнить результаты кластеризации на тестовом множестве.

Все описанные задания требуется выполнить для двух наборов данных:

1) смоделированные независимые случайные векторы (X, Y) , n_1 из которых относятся к первому классу, а n_2 – ко второму классу. Векторы, относящиеся к первому классу, распределены по гауссовскому закону с математическим ожиданием a_1 и корреляционной матрицей R_1 , а векторы, относящиеся ко второму классу, – по гауссовскому закону с математическим ожиданием a_2 и корреляционной матрицей R_2 ;

2) реальные статистические данные из заданного набора (выдаются преподавателем).

Отчёт, кроме прочих обязательных элементов, должен включать:

– изображения данных в виде точек на плоскости, причём данные из разных классов должны изображаться отличающимися друг от друга;

– изображения результатов кластеризации данных в виде точек на плоскости, причём данные из разных кластеров должны изображаться отличающимися друг от друга, кроме того, требуется отметить центры кластеров;

– значения суммы квадратов расстояний от каждого объекта наблюдения до центра соответствующего кластера.

5 Оформить отчет.

Варианты заданий на лабораторную работу

Все описанные задания требуется выполнить для двух наборов данных.

1 В таблице (рисунок А.1) для каждого варианта задания приведены значения количеств объектов наблюдения для каждого класса (n_1 и n_2), значения векторов математических ожиданий для каждого класса (a_1 и a_2) и корреляционные матрицы для каждого класса (R_1 и R_2) для моделируемой выборки из гауссовских случайных векторов.

2 Реальные наборы данных расположены в папке DATA на странице курса в Moodle.

Контрольные вопросы

- 1 В чём состоит задача кластеризации данных?
- 2 Какие существуют различные способы определения расстояния между объектами наблюдения по их признакам?
- 3 К какому классу сложности относится задача кластеризации в классической постановке?
- 4 Как работает классическая реализация алгоритма k внутригрупповых средних?
- 5 Что можно сказать о сходимости алгоритма k внутригрупповых средних?
- 6 Какую функцию минимизирует алгоритм k внутригрупповых средних?
- 7 Какие существуют альтернативные варианты реализации алгоритма k внутригрупповых средних?
- 8 Какие существуют методы автоматического выбора начальных центров кластеров для алгоритма k внутригрупповых средних?
- 9 Что такое иерархическая кластеризация?

Лабораторная работа № 11. Анализ чата группы в Telegram

Цель: применение методов анализа текста для извлечения знаний из неструктурированных текстовых данных.

Теоретические сведения

Развитие социальной коммуникации в сети Интернет в последние годы является очень актуальным. Это предполагает формирование у студентов комплексного представления о процессе анализа неструктурированных текстовых данных на примере общения в мессенджерах. Решение задачи осуществляется посредством последовательного прохождения пяти ключевых этапов: от сбора сырых данных до интерпретации визуализированных результатов.

Этап 1. Сбор данных (Data Acquisition).

Первоочередной задачей исследователя является получение репрезентативной выборки данных. В рамках учебного проекта рассматриваются три основных подхода к экспорту информации из мессенджера Telegram.

Локальный экспорт. Использование штатного функционала клиента Telegram Desktop для выгрузки истории переписки в структурированных форматах JSON или HTML. Данный метод является наиболее доступным и не требует навыков программирования.

Программный доступ через API. Применение официальных интерфейсов взаимодействия – Bot API для работы с ботами или низкоуровневого протокола MTProto (через библиотеки типа Telethon) для автоматизированного сбора данных в реальном времени. Этот подход позволяет масштабировать сбор на большие объемы данных и множество чатов.

Веб-парсинг. Автоматизированный обход веб-версии мессенджера с помощью специализированных библиотек (например, Selenium или BeautifulSoup) для извлечения данных в случаях, когда использование официального API ограничено.

Этап 2. Предобработка текста (Text Preprocessing).

Полученные «сырые» данные содержат значительный объем информационного шума, который необходимо устранить перед анализом. Процесс очистки и нормализации включает следующие операции.

Токенизация. Декомпозиция текстового потока на минимальные смысловые единицы – токены (слова, знаки препинания).

Нормализация. Приведение текста к единому стандарту для исключения дублирования сущностей. Процедура включает приведение всех символов к нижнему регистру, удаление специфических элементов разметки (URL-ссылки, служебные символы), а также фильтрацию упоминаний пользователей (@user) и хештегов (#), которые не несут семантической нагрузки в контексте частотного анализа. При необходимости производится коррекция очевидных опечаток.

Фильтрация стоп-слов. Исключение из выборки служебных частей речи (предлогов, союзов, местоимений), которые встречаются часто, но не определяют тематику сообщения.

Лемматизация и стемминг. Приведение словоформ к их начальному виду. Студентам предлагается сравнить два подхода: стемминг (алгоритмическое отсечение окончаний, работающее быстро, но иногда нарушающее лексическую целостность корня) и лемматизацию (лингвистически точное приведение слова к лемме с учетом его части речи и контекста, требующее больших вычислительных ресурсов).

Этап 3. Векторизация данных (Feature Extraction).

Поскольку алгоритмы математического анализа оперируют числовыми величинами, следующим шагом является трансформация очищенного текста в числовой формат. На данном этапе осуществляется построение матрицы «документ-термин», где каждому уникальному слову присваивается числовой идентификатор или весовой коэффициент. В работе могут быть применены такие методы, как Bag of Words («Мешок слов»), TF-IDF (оценка важности слова в документе относительно коллекции) или современные эмбединги слов.

Этап 4. Анализ данных (Data Analysis).

На основе полученной числовой модели проводится исследование структурных и содержательных характеристик текста. Анализ может включать выявление ключевых тем кластеризацией, оценку тональности сообщений (сентимент-анализ), поиск скрытых зависимостей между активностью участников и содержанием дискуссии, а также статистическое описание распределения терминов.

Этап 5. Визуализация и интерпретация результатов.

Завершающим этапом является представление полученных выводов в наглядной форме, обеспечивающей легкость восприятия и принятия решений. Студентам предлагается реализовать следующие виды визуализации.

Облако тегов (Word Cloud). Графическое отображение частотности слов, где размер шрифта пропорционален значимости термина.

Статистические графики. Гистограммы распределения частот слов и диаграммы для сравнения активности участников.

Анализ динамики. Построение временных рядов, отражающих изменение интенсивности обсуждений или тональности переписки во времени.

Корреляционный анализ. Тепловые карты (Heatmaps) для выявления взаимосвязей между различными параметрами dataset.

Интерактивные дашборды. Создание динамических панелей управления с использованием библиотек (например, Plotly для Python или Shiny для R), позволяющих пользователю самостоятельно фильтровать данные и изменять параметры отображения в реальном времени.

Задания

- 1 Экспортировать историю чата группы (файл .json или .html).
- 2 Импортировать и распарсить данные: извлечь текст сообщений, автора, время.
- 3 Выполнить предобработку:
 - приведение к нижнему регистру, удаление URL, упоминаний;
 - токенизация (unnest_tokens()), удаление стоп-слов;
 - лемматизация (опционально, через textstem).
- 4 Проанализировать:
 - топ-20 наиболее частых слов (столбчатая диаграмма);
 - активность по часам/дням недели (временной ряд);
 - распределение длины сообщений.
- 5 Построить облако тегов (wordcloud2::wordcloud2()).

Контрольные вопросы

- 1 Какие этапы включает предобработка текстовых данных?
- 2 Зачем удалять стоп-слова и как это влияет на анализ?
- 3 Как визуализировать временные паттерны сообщений?
- 4 В чём ограничения автоматического сентимент-анализа?

Список литературы

- 1 **Кабаков, Р.** R в действии. Анализ и визуализация данных на языке R : практическое руководство / Р. Кабаков. – 2-е изд. – М. : ДМК Пресс, 2023. – 590 с.
- 2 **Лонг, Д. Р.** Книга рецептов. Проверенные рецепты для статистики, анализа и визуализации данных : практическое руководство / Д. Лонг, П. Титор. – М. : ДМК Пресс, 2020. – 510 с.
- 3 **Мастицкий, С. Э.** Статистический анализ и визуализация данных с помощью R : практическое руководство / С. Э. Мастицкий, В. К. Шитиков. – 2-е изд. – М. : ДМК Пресс, 2023. – 497 с.
- 4 **Гайдель, А. В.** Лабораторный практикум по курсу «Интеллектуальный анализ данных»: практикум / А. В. Гайдель, А. Г. Храмов. – Самара : Самар. ун-т, 2019. – 104 с.

Приложение А (обязательное)

Варианты заданий для генерации двух выборок двумерного нормального распределения

№	n_1	a_1	R_1	n_2	a_2	R_2
0	100	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	100	$\begin{pmatrix} 2 \\ 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
1	100	$\begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0,5 \\ 0,5 & 2 \end{pmatrix}$	200	$\begin{pmatrix} 2 \\ 2 \end{pmatrix}$	$\begin{pmatrix} 2 & 0,5 \\ 0,5 & 1 \end{pmatrix}$
2	1000	$\begin{pmatrix} -1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0,9 \\ 0,9 & 2 \end{pmatrix}$	2000	$\begin{pmatrix} 2 \\ 3 \end{pmatrix}$	$\begin{pmatrix} 2 & 0,9 \\ 0,9 & 1 \end{pmatrix}$
3	100	$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0,1 \\ 0,1 & 2 \end{pmatrix}$	50	$\begin{pmatrix} 2 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 2 & 0,1 \\ 0,1 & 1 \end{pmatrix}$
4	1000	$\begin{pmatrix} 0 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 1 & -0,5 \\ -0,5 & 2 \end{pmatrix}$	500	$\begin{pmatrix} 3 \\ 2 \end{pmatrix}$	$\begin{pmatrix} 2 & -0,5 \\ -0,5 & 1 \end{pmatrix}$
5	100	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 & -0,9 \\ -0,9 & 2 \end{pmatrix}$	1000	$\begin{pmatrix} 3 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 2 & -0,9 \\ -0,9 & 1 \end{pmatrix}$
6	1000	$\begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 1 & -0,1 \\ -0,1 & 2 \end{pmatrix}$	100	$\begin{pmatrix} 4 \\ 2 \end{pmatrix}$	$\begin{pmatrix} 2 & -0,1 \\ -0,1 & 1 \end{pmatrix}$
7	100	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$	200	$\begin{pmatrix} 4 \\ 3 \end{pmatrix}$	$\begin{pmatrix} 2 & 0,5 \\ 0,5 & 1 \end{pmatrix}$
8	1000	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$	2000	$\begin{pmatrix} 4 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 2 & 0,1 \\ 0,1 & 1 \end{pmatrix}$
9	100	$\begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$	50	$\begin{pmatrix} -4 \\ -4 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}$
10	1000	$\begin{pmatrix} -1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}$	500	$\begin{pmatrix} -4 \\ -3 \end{pmatrix}$	$\begin{pmatrix} 2 & -0,5 \\ -0,5 & 1 \end{pmatrix}$
11	100	$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}$	1000	$\begin{pmatrix} -4 \\ -2 \end{pmatrix}$	$\begin{pmatrix} 2 & -0,1 \\ -0,1 & 1 \end{pmatrix}$
12	1000	$\begin{pmatrix} 0 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}$	100	$\begin{pmatrix} -3 \\ -4 \end{pmatrix}$	$\begin{pmatrix} 2 & 0,1 \\ 0,1 & 1 \end{pmatrix}$
13	100	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$	200	$\begin{pmatrix} -3 \\ -2 \end{pmatrix}$	$\begin{pmatrix} 1 & 0,5 \\ 0,5 & 2 \end{pmatrix}$
14	1000	$\begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$	2000	$\begin{pmatrix} -2 \\ -4 \end{pmatrix}$	$\begin{pmatrix} 1 & 0,1 \\ 0,1 & 2 \end{pmatrix}$
15	100	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$	50	$\begin{pmatrix} -2 \\ -2 \end{pmatrix}$	$\begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix}$

Рисунок А.1 – Данные для генерации