МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯРОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»(ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет информационных технологий и компьютерной безопасности

Кафедра компьютерных интеллектуальных технологий проектирования

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

По дисциплине: «Искусственный интеллект в программно-информационных системах»

Тема: «Методы анализа данных»

Выполнил работу студент группы змИИВТ-231:                            Енин М. М.

подпись, дата

Принял:                                                                                             Короленко В.В.

подпись, дата

Воронеж 2024

Содержание

1. Содержание 2
2. Введение 3
3. Описание работы 3
4. Заключение 5
5. Контрольные вопросы 6
6. Литература 13

# **ВВЕДЕНИЕ**

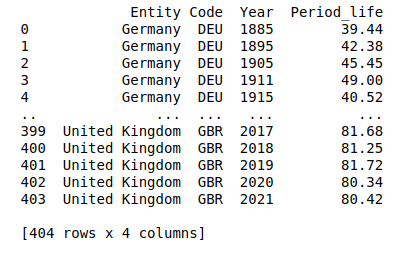
**Jupyter Notebook** – это веб-приложение, которое позволяет создавать и делиться документами, содержащими код на различных языках программирования, математические выражения, графики и текстовые комментарии. Оно широко используется в научных исследованиях, анализе данных и образовании благодаря своей интуитивной и интерактивной природе.

Jupyter Notebook состоит из ячеек двух типов: ячейки кода и ячейки markdown. Ячейки кода содержат исходный код, который можно выполнять для получения результатов, в то время как ячейки markdown позволяют добавлять форматированный текст, изображения и математические формулы.

**Описание работы.**

Для работы были взяты данные с сайта <https://www.kaggle.com/>. Набор данных представляет выборку средней продолжительности жизни в разных странах по гадам.

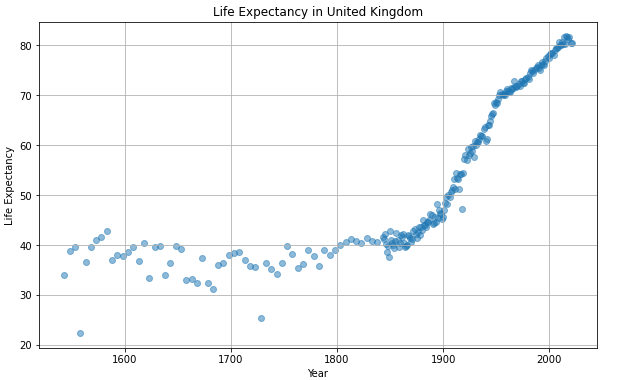
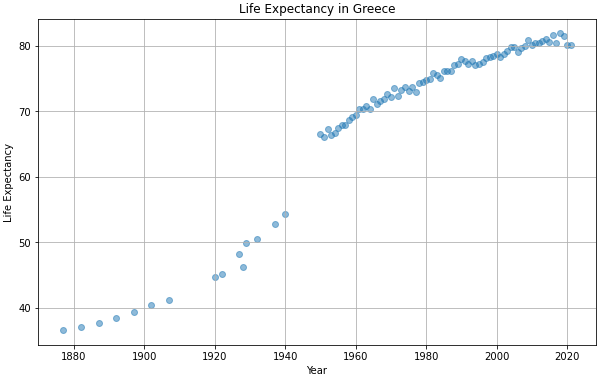
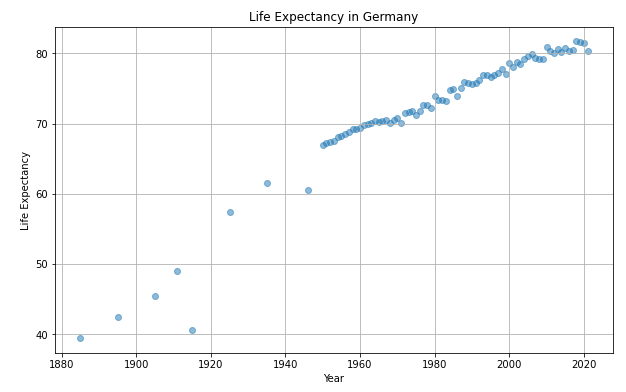
1. Страна
2. Код страны
3. Год
4. Период жизни



Для поверхностного обзора возьмем три страны и представим в рафическом распределении с помощью plt.figure

|  |
| --- |
| import pandas as pd # Импортируем библиотеку pandas для работы с данными в формате таблицы  import matplotlib.pyplot as plt # Импортируем библиотеку matplotlib.pyplot для создания графиков  # Загрузка данных из файла  data = pd.read\_csv('life-expectancy-new.txt', delimiter=',') # Читаем данные из CSV-файла и сохраняем их в переменную data  print(data) # Выводим содержимое таблицы данных  # Получение списка уникальных стран  countries = data['Entity'].unique() # Получаем список уникальных стран из столбца 'Entity' таблицы данных  # Создание графиков для каждой страны  for country in countries: # Проходим по каждой стране в списке  country\_data = data[data['Entity'] == country] # Фильтруем данные, оставляя только строки, относящиеся к текущей стране  years = country\_data['Year'] # Получаем список лет из столбца 'Year' для текущей страны  life\_expectancy = country\_data['Period\_life'] # Получаем значения ожидаемой продолжительности жизни из столбца 'Period\_life' для текущей страны  plt.figure(figsize=(10, 6)) # Создаем новый рисунок с размером 10x6 дюймов  plt.scatter(years, life\_expectancy, alpha=0.5) # Строим точечный график, где по оси x - годы, по оси y - ожидаемая продолжительность жизни  plt.title(f'Life Expectancy in {country}') # Устанавливаем заголовок графика  plt.xlabel('Year') # Устанавливаем подпись для оси x  plt.ylabel('Life Expectancy') # Устанавливаем подпись для оси y  plt.grid(True) # Включаем отображение сетки на графике  plt.show() # Отображаем график |

На основе данной выборки получим графики средней продолжительности жизни.



# **Заключение**

Jupyter Notebook и Statistica являются мощными инструментами для анализа данных и проведения исследований в различных областях. Jupyter Notebook предоставляет гибкую и интерактивную среду для работы с кодом, визуализации данных и документирования результатов. Он поддерживает множество языков программирования, что делает его универсальным инструментом для специалистов в области науки о данных, машинного обучения и анализа данных.

# **Контрольные вопросы**

Термины:

1. Данные - это набор сведений, фактов или значений, которые могут быть использованы для анализа, принятия решений или для создания моделей. Данные могут быть представлены в различных форматах, таких как числовые, текстовые, графические и т.д.

2. Предварительный анализ данных - это процесс изучения и обработки данных перед их использованием в анализе или моделировании. Он включает в себя такие действия, как очистка данных от ошибок и пропусков, преобразование данных в нужный формат, выявление и устранение аномалий и выбросов, а также определение корреляций между переменными.

3. Регрессия - это статистический метод, используемый для изучения взаимосвязи между зависимой переменной и одной или несколькими независимыми переменными. Цель регрессионного анализа - построить модель, которая может предсказывать значения зависимой переменной на основе значений независимых переменных.

4. Классификация - это задача машинного обучения, цель которой - на основе имеющихся данных и их признаков определить, к какому классу (категории) принадлежит новый объект. Классификация используется в таких задачах,как распознавание образов, анализ текста, определение спама и т.д.

5. Кластеризация - это задача машинного обучения, цель которой - разделить имеющиеся данные на группы (кластеры) таким образом, чтобы объекты внутри одного кластера были более похожи друг на друга, чем на объекты из других кластеров. Кластеризация используется в таких задачах, как сегментация клиентов, анализ социальных сетей, сжатие данных и т.д.

6. Библиотеки Python, используемые для работы в области машинного обучения: - NumPy: предоставляет мощные структуры данных для работы с многомерными массивами и матрицами, а также функции для выполнения вычислений над этими массивами. - Pandas: предназначена для анализа и манипуляции данными, предоставляет удобные структуры данных и инструменты для анализа одномерных и двумерных данных. - Scikit-learn: является одной из основных библиотек для машинного обучения, предоставляет реализацию множества алгоритмов классификации, регрессии, кластеризации и визуализации данных. - TensorFlow: открытая библиотека для разработки и обучения нейронных сетей и других алгоритмов глубинного обучения. - Keras: высокоуровневая библиотека для создания нейронных сетей, работает на базе TensorFlow, Theano или CNTK. - Matplotlib: библиотека для созданиядвумерных графиков и диаграмм, используется для визуализации данных. - Seaborn: основана на Matplotlib, предоставляет более высокоуровневые функции для визуализации данных и построения статистических графиков. - SciPy: библиотека для научных и инженерных вычислений, включает в себя инструменты для оптимизации, интегрирования, обработки сигналов и других задач.

7. Библиотеки Python получили широкое распространение в машинном обучении благодаря следующим причинам: - Легкость использовани: Python имеет простой синтаксис и интуитивно понятный язык, что упрощает процесс обучения и разработки. - Большое сообщество: Python имеет одно из самых больших сообществ разработчиков, что обеспечивает поддержку, обмен опытом и быстрое развитие библиотек. - Интерфейс с другими языками: Python может взаимодействовать с кодом на других языках, таких как C/C++ и Java, что позволяет использо его для разработки сложных систем. - Многофункциональность: Python подходит для решения широкого круга задач, от веб-разработки до научных вычислений и машинного обучения. - Открытый исходный код: многие библиотеки для машинного обучения имеют открытый исходный код, что позволяет изучать их внутреннее устройство и адаптировать под свои нужды.

8. Анализ данных – это процесс исследования и изучения данных с целью извлечения полезной информации, выявления закономерностей, тенденций и связей, а также принятия обоснованных решений на основе полученных результатов.

9. Основные задачи анализа данных: - Очистка данных: удаление или исправление некорректных, неполных или несоответствующих данных. -\*Предобработка данных: преобразование и нормализация данных для подготовки их к анализу. - Визуализация данных: представление данных в виде графиков, диаграмм и других визуальных элементов для облегчения понимания и анализа. - Статистический анализ: использование статистических методов для изучения данных и выявления закономерностей. - Прогнозирование: построение моделей для предсказания будущих значений на основе исторических данных. - Классификация и кластеризация: разделение данных на группы или классы на основе их свойств. - Оптимизация: поиск наилучших решений для задач, таких как минимизация затрат или максимизация прибыли.

10. Примеры применения методов анализа данных: - Банковское дело: анализ кредитного риска, определение кредитного рейтинга клиента. - Маркетинг: анализ поведения потребителей, выявление сегментов рынка, прогнозирование продаж. -Медицина: анализ данных медицин

11. Одним из актуальных направлений в области анализа данных является анализ текстовых данных и естественного языка (Natural Language Processing, NLP). Этот раздел анализа данных фокусируется на взаимодействии между компьютерами и человеческим языком, позволяя машинам понимать и анализировать большие объемы текстовых данных. Приложения включают в себя автоматическое реферирование, перевод текстов, распознавание речи и анализ тональности.

12. Алгоритм построения системы анализа данных может включать следующие шаги: - Определение целей анализа и бизнес-задач, которые нужно решить с помощью анализа данных. - Сбор и подготовка данных: сбор данных из различных источников, очистка данных от дубликатов, пропусков и ошибок, преобразование данных в нужный формат. - А данных: применение статистических методов и алгоритмов машинного обучения для выявления закономерностей, тенденций и взаимосвязей в данных. - Визуализация результатов: представление результатов анализа в удобном для восприятия виде, например, с помощью графиков, диаграмм и таблиц. - Интерпретация результатов: анализ полученных результатов и формулирование выводов, которые могут быть использованы для принятия решений и оптимизации бизнес-процессов. - Внедрение результатов: использование полученных результатов для улучшения деятельности компании, разработки новых продуктов или услуг, оптимизации маркетинговых стратегий и т.д.

13. Примеры непрерывных данных: - Рост человека (например, 175 см). - Вес продукта (например, 0,5 кг). - Температура воздуха (например, 23,5 градуса по Цельсию). - Время выполнения задачи (например, 3,25 минуты).

14. Примеры категориальных данных: - Пол человека (мужской, женский). - Тип транспортного средства (автомобиль, велосипед, автобус). - Национальность (русский, украинец, белорус). - Образование (среднее, высшее, неоконченное высшее).

15. Источник данных - это место, откуда берутся данные для анализа. Это может быть внутренняя база данных компании, внешние базы данных, данные, собранные с помощью опросов или наблюдений, данные из социальных сетей и т.д.

16. Способы классификации источников данных: - По происхождению: внутренние (данные, собранные внутри организации) и внешние (данные, полученные из внешних источников). - По типу данных: количественные (данные, которые могут быть измерены или подсчитаны) и качественные (данные, которые описывают свойства или характеристики объектов). - По структуре: структурированные (данные, хранящиеся в таблицах или базе данных) и неструктурированные (данные, хранящиеся в виде текстов, изображений, видео и т.д.).

17. Открытые данные — это данные, которые доступны для свободного использования и распространения. Они могут быть получены из государственных источников, научных организаций, компаний и т.д.

18. Примеры источников открытых данных: - World Bank Open Data: предоставляет доступ к глобальным данным, включая экономические, социальные и другие покаые карты. Для этого мы отбираем данные, используя запрос SQL: sql SELECT \* FROM clients WHERE has\_credit\_card = TRUE; Этот запрос выберет все записи в таблице "clients", где поле "has\_credit\_card" имеет значение TRUE, то есть только данные о клиентах с кредитными картами.

19. Операция нормализации данных: Нормализация данных — это процесс преобразования значений данных из исходного диапазона в стандартизированный диапазон или формат, обычно от 0 до 1 или от -1 до 1. Цель нормализации — устранить различия в масштабах измерения, чтобы каждый признак имел одинаковый вес при анализе или машинном обучении. Это помогает избежать проблем, связанных с тем, что некоторые признаки могут доминировать над другими только из-за их большего диапазона значений.

20. Пример нормализации данных: Предположим, у нас есть набор данных с двумя признаками: возрастом (от 18 до 80 лет) и доходом (от 10 000 до 100 000 долларов в ный формат хранения данных, оптимизированный для анализа больших наборов данных.

21. Алгоритм построения системы сбора данных на основе программного обеспечения Microsoft Excel: 1. Определение источников данных: выявление источников, из которых будут собираться данные (например, базы данных, веб-сервисы, файлы). 2. Подготовка шаблона Excel: создание рабочей книги Excel с необходимыми таблицами и формулами для обработки данных. 3. Автоматизация импорта данных: использование функций импорта данных Excel (например, «Из текста», «Из базы данных») для автоматического заполнения таблиц. 4. Обработка данных: применение формул, фильтров и сортировок для анализа и преобразования данных. 5. Визуализация данных: создание диаграмм и графиков для наглядного представления данных. 6. Сохранение и обновление данных: периодическое обновление данных и сохранение результатов в нужном формате.

22. Необходимость подготовки данных обоснована следующими факторами: - Улучшение качества данных: подготовка позволяет устранить ошибки, дубликаты и несоответствия в данных. - Соответствие требованиям анализа: данные должны быть приведены к нужному формату и структуре для корректного применения алгоритмов анализа. - Эффективность обработки: подготовленные данные позволяют ускорить процесс анализа и снизить нагрузку на систему.

23. Операция форматирования данных: Форматирование данных – это процесс приведения данных к определенному виду, который удобен для анализа и обработки. Оно включает в себя установление правил для представления данных (например, формат даты, валюты), а также изменение внешнего вида данных (шрифт, цвет, границы ячеек и т.д.).

24. Пример форматирования данных: - Преобразование текста в верхний или нижний регистр. - Применение денежного формата к числовым данным. - Установка формата даты для ячеек с датами.

25. Операция отбора данных: Отбор данных – это процесс выбора из общего набора данных только тех данных, которые соответствуют определенным критериям или требованиям. Это может включать фильтрацию данных по значениям, диапазонам, связям между данными и другим параметрам. Отбор данных помогает сократить объем обрабатываемой и сосредоточиться на наиболее релевантных данных для решения конкретной задачи.

26. Пример отбора данных: Допустим, у нас есть база данных с информацией о клиентах банка. Мы хотим проанализировать данные о клиентах, которые имеют кредитные карты. Для этого мы отбираем данные, используя запрос SQL: sql SELECT \* FROM clients WHERE has\_credit\_card = TRUE; Этот запрос выберет все записи в таблице "clients", где поле "has\_credit\_card" имеет значение TRUE, то есть только данные о клиентах с кредитными картами.

27. Операция нормализации данных: Нормализация данных — это процесс преобразования значений данных из исходного диапазона в стандартизированный диапазон или формат, обычно от 0 до 1 или от -1 до 1. Цель нормализации — устранить различия в масштабах измерения, чтобы каждый признак имел одинаковый вес при анализе или машинном обучении. Это помогает избежать проблем, связанных с тем, что некоторые признаки могут доминировать над другими только из-за их большего диапазона значений.

28. Пример нормализации данных: предположим, у нас есть набор данных с двумя признаками: возрастом (от 18 до 80 лет) и доходом (от 10 000 до 100 000 долларов в год). Мы хотим нормализовать эти данные, используя формулу для мини-максимальной нормализации: ``` normalized\_value = (original\_value - min\_value) / (max\_value - min\_value) ``` Для возраста нормализованное значение для человека в возрасте 40 лет будет: normalized\_age = (40 - 18) / (80 - 18) = 22 / 62 ≈ 0.355 Для дохода нормализованное значение для человека с доходом 50 000 долларов будет: normalized\_income = (50000 - 10000) / (100000 - 10000) = 40000 / 90000 ≈ 0.444 Таким образом, оба признака теперь находятся в диапазоне от 0 до 1. 24) Операция кодирования данных: Кодирование данных — это процесс преобразования категориальных данных в формат, который можно использовать в алгоритмах машинного обучения. Это может включать в себя такие методы, как one-hot encoding, label encoding, ordinal encoding и другие. Цель кодирования — представить категориальные переменные числовыми значениями, которые могут быть понятны и использованы алгоритмами машинного обучения.

29. Пример кодирования данных: Допустим, у нас есть набор данных с категориальным признаком "цвет глаз", который может принимать значения "карие", "голубые", "зеленые" и "серые". Мы хотим использовать one-hot encoding для кодирования этого признака. Для этого мы создаем новые бинарные признаки для каждой категории: - Если цвет глаз "карие", то: - brown\_eyes = 1 - blue\_eyes = 0 - green\_eyes = 0 - gray\_eyes = 0 - Если цвет глаз "голубые", то: - brown\_eyes = 0 - blue\_eyes = 1 - green\_eyes = 0 - gray\_eyes = 0 - И так далее для других цветов. Таким образом, каждый цвет глаз теперь представлен в виде отдельной бинарной переменной, и алгоритмы машинного обучения могут использовать эти данные.

**Список литературы:**

1. Уэс, Маккинли. Python и анализ данных Электронный ресурс / Маккинли Уэс ; пер. А. А. Слинкин. - Python и анализ данных,2022-04-19. - Саратов : Профобразование, 2017. - 482 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4488-0046-7, экземпляров неограниченно.

2. Сузи, Р.А. Язык программирования Python Электронный ресурс : учебное пособие / Р.А. Сузи. - Язык программирования Python,2020-07-28. - Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 350 c. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 5-9556-0058-2, экземпляров неограниченно.

3. Стенли, Липпман. Язык программирования С++ Электронный ресурс : Полное руководство / Липпман Стенли, Лажойе Жози ; пер. А. Слинкин. - Язык программирования С++,2022-04-19. - Саратов : Профобразование, 2017. - 1104 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4488-0136-5, экземпляров неограниченно

4. https://github.com/enikolaev/MMO – Репозиторий с примерами кода из лабораторных работ.

5. https://archive.ics.uci.edu/ml/index.html – Репозиторий наборов данных для машинного обучения (Центр машинного обучения и интеллектуальных систем).

6. https://www.kaggle.com – Портал и система проведения соревнований по проблемам анализа данных.

7. https://www.mockaroo.com – Сайт для генерации наборов данных.

8. <https://habr.com/ru/articles/349204> – Статья на хабре: «Моделирование динамических систем: введение в GNU Octave»