Geekbrains

Исследование факторов, влияющих на ожидаемую продолжительность жизни с применением алгоритмов машинного обучения

Программа: Разработчик-аналитик
Специализация Data Engineer
Коноваленко Валерия Владимировна

Барановичи
2024

СОДЕРЖАНИЕ

Введение

До середины XIX века ожидаемая продолжительность жизни людей при рождении составляла от 20 до 50 лет. Достижения в области общественного здравоохранения и медицины в начале XX века привели к резкому росту продолжительности жизни вплоть до ста лет. Однако на темпы увеличения ожидаемой продолжительности жизни при рождении продолжали влиять географическое положение, экономическое развитие и временные факторы.

Ожидаемая продолжительность жизни, важный показатель, комплексно характеризующий уровень смертности в стране и активно используемый для анализа динамики продолжительности жизни.

Точное прогнозирование будущих тенденций в области ожидаемой продолжительности жизни имеет важное значение для социальной политики, политики в области здравоохранения и экономической политики.

Например, известно, что в настоящее время наблюдается значимое увеличение доли пожилого (т.е. нетрудоспособного) населения, что обернется в ближайшем будущем целым рядом социальных и экономических проблем. Это очень важный показатель для финансов нашего государства, он определяет размер пенсионных выплат, которые закладываются в бюджет. Ошибаться нельзя, иначе денег на выплаты пенсий может просто не хватить. И как следствие нас ожидает очередное повышение пенсионного возраста (вероятнее всего нам придется работать как минимум на 10 лет дольше).

Целью данного проекта является исследование ожидаемой продолжительности жизни населения разных стран с применением алгоритмов машинного обучения и выявление факторов, способствующих ее снижению.

Для достижения поставленной цели будут использоваться алгоритмы машинного обучения, такие как случайный лес, модель линейной регрессии, метод K-ближайших соседей (KNN). Эти алгоритмы позволят проанализировать данные о факторах, влияющих на ожидаемую продолжительность жизни, обнаружить закономерности в данных, а затем сделать выводы и принять решения на основе этих закономерностей.

При проведении исследования использовались такие инструменты как Python, Jupiter notebook и различные библиотеки для анализа данных.

В качестве исследуемого объекта была взята база данных Global Health Observatory (GHO) за 2000–2015 годы по 193 странам с сайта www.kaggle.ru.

Поскольку наблюдения, представленные в этом наборе данных, основаны на данных из разных стран, применение машинного обучения в данном проекте позволит легче определить прогнозирующий фактор, который способствует снижению значения ожидаемой продолжительности жизни. Это поможет подсказать стране, какой области следует уделить особое внимание, чтобы эффективно увеличить ожидаемую продолжительность жизни ее населения.

Глава 1. Основы машинного обучения: виды и их алгоритмы

1.1 Что такое машинное обучение

Машинное обучение - это раздел искусственного интеллекта, который изучает методы, алгоритмы и модели, позволяющие компьютеру обучаться на основе данных, не явно программируясь на определенные инструкции. В отличие от классического программирования, где набор инструкций задается заранее, в машинном обучении компьютер самостоятельно определяет закономерности и шаблоны в данных, чтобы делать прогнозы или принимать решения.

Используя технологию машинного обучения, программисты больше не обязаны тратить время на написание инструкций, рассматривающих все возможные сценарии и содержащих все решения. Вместо этого они могут встроить в компьютер или программу алгоритм, который самостоятельно находит решения, используя статистические данные для выявления закономерностей и предсказаний[3].

Технология машинного обучения на основе анализа данных впервые появилась в 1950-х годах при разработке программ для игры в шашки. За прошедшие десятилетия этот общий принцип остался неизменным, но благодаря резкому увеличению вычислительной мощности компьютеров, стало возможным создание более сложных закономерностей и прогнозов, а также решение более широкого круга задач с применением машинного обучения.

Для начала процесса машинного обучения необходимо загрузить в компьютер набор данных, на которых алгоритм будет учиться. Например, это могут быть изображения собак и кошек с уже расставленными метками, указывающими на их вид. После обучения программа сможет самостоятельно распознавать собак и кошек на новых фотографиях без меток. Чем больше данных обработала программа, тем точнее будет ее распознавание[4].

Сегодня благодаря машинному обучению компьютеры могут распознавать и фотографии, и изображения, и тексты. Например, программы уже

умеют распознавать не только лица на фотографиях, но и пейзажи, предметы, текст и цифры. Для текстов также необходимо машинное обучение: проверка грамматики в текстовых редакторах и на мобильных устройствах уже стала стандартом. Также существуют программы, способные автоматически создавать новостные статьи на различные темы без участия человека.

1.2 Основные типы и виды машинного обучения

Все задачи, решаемые с помощью ML, относятся к одной из следующих категорий:

1) *Задача регрессии* заключается в предсказании числового значения на основе данных с различными признаками. Например, прогнозирование цены акций через определенный период времени или ожидаемый объем продаж товара на следующий месяц.

2) *Задача классификации* состоит в присвоении объекту определенной категории на основе набора признаков. Например, определение наличия определенного объекта на изображении или диагностика болезни по медицинским показателям.

3) *Задача кластеризации* предполагает разделение данных на группы по их сходству без предварительного определения категорий. Например, классификация покупателей по их покупательским привычкам или разделение текстов по тематике.

4) *Задача уменьшения* размерности заключается в сокращении количества признаков для упрощения анализа данных и улучшения визуализации. Например, сокращение размерности для отображения данных на двумерном графике.

5) *Задача выявления* аномалий заключается в обнаружении нестандартных и редких случаев в данных. Например, выявление мошеннических операций с кредитными картами или выявление необычных поведенческих шаблонов в данных.

Основные виды машинного обучения

Машинное обучение подразделяется на два основных типа: обучение с учителем и обучение без учителя. В обоих случаях машинам предоставляются исходные данные для анализа и выявления закономерностей, но различие заключается в участии человека в обучении.

1) Обучение с учителем

При обучении с учителем машине предоставляются данные вместе с ответами, что позволяет ей проверять свои гипотезы. Задача заключается в том, чтобы создать модель, которая может предсказывать целевую переменную (например, цену квартиры) на основе входных данных (площадь, расположение и т. д.). Примеры задач обучения с учителем включают регрессию (предсказание непрерывных значений, таких как цена) и классификацию (разделение объектов на категории, такие как спам или не спам).

2) Обучение без учителя

В обучении без учителя машине не предоставляются ответы. Она должна самостоятельно обнаруживать шаблоны и структуры в данных. Задачи обучения без учителя включают:

- Кластеризация: Разделение данных на группы на основе сходства, например, в случае распределения людей по размерным группам для пошива рубашек.

- Уменьшение размерности: Уменьшение количества признаков в данных для облегчения их отображения и анализа. Например, отображение данных с сотнями признаков в двух- или трехмерном пространстве.

1.3 Основные алгоритмы машинного обучения

Алгоритмы моделей машинного обучения - это математические процедуры, которые используются для обучения моделей машинного обучения на данных. Они определяют, как модель будет учиться и делать прогнозы.

Существует множество различных алгоритмов машинного обучения, каждый со своими сильными и слабыми сторонами.

Вот некоторые общие категории алгоритмов машинного обучения.

1) Дерево принятия решений для бизнеса

Дерево принятия решений - это метод, который помогает принимать решения с учетом потенциальных последствий, эффективности и затрат. В контексте бизнес-процессов дерево принятия решений создается путем последовательности вопросов "да/нет", которые ведут к правильному выбору.

Этот метод структурирует и систематизирует проблему, обеспечивая логическую основу для принятия решений. Дерево принятия решений особенно полезно, когда необходимо учитывать несколько факторов и оценивать вероятность различных событий[7].

Вот как работает дерево принятия решений для бизнеса:

- Начните с определения проблемы или решения, которое нужно принять.

- Создайте корневой узел, представляющий исходную проблему.

- Добавьте ветви для каждого возможного решения или действия.

- Для каждой ветви добавьте узлы, представляющие возможные последствия, эффективность и затраты.

- Повторяйте этот процесс, создавая дочерние узлы, пока не достигнете конечных узлов, представляющих конкретные решения.

- Оцените вероятность и влияние каждого возможного пути.

- Выберите путь, который приводит к наилучшему результату с учетом всех факторов.

Дерево принятия решений помогает предприятиям принимать обоснованные решения, учитывая все релевантные факторы и возможные последствия. Оно обеспечивает прозрачный и систематический подход к принятию решений, что особенно ценно в сложных и неопределенных ситуациях.

2) Наивная байесовская классификация

Наивная байесовская классификация - это метод машинного обучения, используемый для классификации данных. Он основан на теореме Байеса, которая позволяет рассчитывать вероятность события на основе имеющейся информации.

Наивность в названии метода заключается в предположении, что признаки независимы друг от друга, даже если на практике это может быть и не так. Несмотря на это упрощение, наивные байесовские классификаторы часто показывают хорошие результаты в реальных задачах.

Вот как работает наивная байесовская классификация:

- Рассчитать вероятность того, что объект принадлежит каждому классу на основе имеющихся данных.

- Рассчитать вероятность того, что объект имеет конкретный набор признаков для каждого класса.

- Умножить вероятности из шагов 1 и 2 для каждого класса.

- Классифицировать объект в класс с наибольшим произведением вероятностей.

Наивные байесовские классификаторы широко используются в следующих областях:

- Фильтрация спама

- Классификация новостных статей

- Анализ настроений

- Распознавание лиц и образов

Они популярны благодаря своей простоте, эффективности и возможности работать с большим количеством признаков и данных.

3) Метод наименьших квадратов

Метод наименьших квадратов - это статистический метод, используемый для подгонки прямой к набору данных. Цель метода - найти прямую, которая наиболее точно соответствует точкам данных, минимизируя сумму квадратов расстояний между точками и прямой.

Как работает метод наименьших квадратов:

- Предварительная обработка данных: данные преобразуются в числовой формат, и для каждого объекта создается набор признаков.

- Определение целевой переменной: выбирается целевая переменная, которую необходимо предсказать.

- Подгонка прямой: алгоритм метода наименьших квадратов используется для подгонки прямой к данным. Алгоритм находит значения параметров прямой, которые минимизируют сумму квадратов расстояний между точками данных и прямой.

- Прогнозирование: после подгонки прямой ее можно использовать для прогнозирования целевой переменной для новых объектов.

Метод наименьших квадратов широко используется в машинном обучении для решения задач регрессии. Регрессия - это задача прогнозирования непрерывных целевых переменных (например, цены на акции или продаж).

Преимущества:

- Простой и эффективный алгоритм

- Хорошо подходит для задач регрессии с линейными зависимостями

- Может работать с большим количеством признаков и данных

Недостатки:

- Предположение о линейной зависимости может снизить точность в некоторых случаях

- Может быть чувствителен к шуму и выбросам в данных

Области применения:

- Прогнозирование спроса

- Анализ временных рядов

- Оценка рисков

- Финансовое моделирование

4) Логистическая регрессия

Логистическая регрессия - это метод анализа данных, который используется для предсказания вероятности возникновения определенных событий на основе одной или нескольких независимых переменных. Она

применяется в различных областях, таких как кредитный скоринг, анализ рекламных кампаний, прогнозирование прибыли от продаж товаров и предсказание естественных катастроф. Логистическая регрессия позволяет оценить влияние различных факторов на вероятность наступления события и принять обоснованные решения на его основе.

5) Метод опорных векторов

Метод опорных векторов (SVM) представляет собой набор алгоритмов, которые широко используются для решения задач классификации и регрессионного анализа. SVM строит гиперплоскость в N-мерном пространстве, чтобы разделить объекты на два класса. Гиперплоскость создается таким образом, чтобы максимально удалиться от ближайшей точки каждого класса.

Этот метод помогает решать разнообразные задачи машинного обучения, такие как сплайсинг ДНК, определение пола человека по фотографии, и показ рекламы на сайтах[10].

6) Метод ансамблей

Метод ансамблей в машинном обучении - это подход, при котором используется не один классификатор, а множество классификаторов, которые работают вместе для принятия решения. Этот метод снижает влияние случайных ошибок и уменьшает дисперсию результатов, так как объединение нескольких моделей дает более точные прогнозы, чем одна отдельно взятая.

Подход ансамбль учитывает различные гипотезы и позволяет расширить множество базовых гипотез для более точного прогнозирования.

7) Кластеризация

Кластеризация - это процесс разделения группы объектов на подмножества, называемые кластерами, таким образом, чтобы объекты в каждом кластере были более похожи друг на друга, чем на объекты из других кластеров.

Существует несколько алгоритмов для кластеризации объектов. Некоторые из них включают:

- Алгоритмы на основе центра тяжести, которые определяют кластеры на основе расстояния между объектами и их центром тяжести.

- Алгоритмы подключения, которые определяют кластеры на основе связей между объектами.

- Алгоритмы сокращения размерности, которые уменьшают размерность данных для упрощения кластеризации.

- Алгоритмы плотности, которые опираются на плотность объектов в пространстве для определения кластеров.

- Вероятностные алгоритмы, которые используют вероятностные модели для кластеризации.

- Методы машинного обучения, включая нейронные сети, для кластеризации данных.

Алгоритмы кластеризации применяются в различных областях, таких как биология (для анализа геномов и генетических данных), социология (для анализа социологических данных) и информационные технологии (для группировки данных и поиска закономерностей)[1].

8) Метод главных компонент

Метод главных компонент (PCA) – это статистический метод ортогонального преобразования, который используется для преобразования наблюдений над переменными, имеющими какие-то взаимосвязи, в новый набор главных компонентов – линейно некоррелированных значений.

PCA обычно применяется для визуализации данных и сжатия информации, что позволяет упростить и уменьшить размерность данных для дальнейшего анализа. Однако PCA не подходит для данных, в которых все компоненты имеют высокую дисперсию и слабо упорядочены. Поэтому его применимость зависит от того, насколько хорошо изучена предметная область и какие данные доступны для анализа[9].

9) Сингулярное разложение

Сингулярное разложение, или SVD, в линейной алгебре представляет собой разложение прямоугольной матрицы на произведение трех матриц: U, Σ и V, где U и V являются унитарными матрицами, а Σ - диагональной матрицей.

Метод главных компонент является частным случаем сингулярного разложения. В прошлом алгоритмы компьютерного зрения основывались на этом методе, представляя лица или другие объекты как сумму базисных компонент, уменьшая их размерность и сравнивая с образцами из выборки. Современные алгоритмы сингулярного разложения в машинном обучении намного сложнее своих предшественников, но их основной принцип остается неизменным.

10) Анализ независимых компонент

Анализ независимых компонент, или ICA - это метод, который позволяет выявить скрытые факторы, влияющие на случайные величины или сигналы. Он строит модель, объясняющую данные с использованием независимых компонентов, которые считаются негауссовскими сигналами. В отличие от анализа главных компонент, ICA эффективнее в ситуациях, когда стандартные методы неэффективны. Этот метод широко используется в различных областях, таких как астрономия, медицина, распознавание речи, финансовый анализ и другие. Он помогает обнаружить скрытые причины явлений и является мощным инструментом для исследований и анализа данных[2].

1.4 Инструменты в машинном обучении

Инструменты машинного обучения - это программные платформы или библиотеки, которые предоставляют набор функций и алгоритмов для разработки, обучения и развертывания моделей машинного обучения. Эти инструменты позволяют разработчикам и специалистам по данным автоматизировать процессы машинного обучения и создавать более точные и эффективные модели.

Вот некоторые из наиболее популярных инструментов машинного обучения:

- TensorFlow: открытая платформа с открытым исходным кодом, разработанная Google. TensorFlow предоставляет широкий спектр инструментов

и библиотек для создания, обучения и развертывания моделей глубокого обучения и других типов моделей машинного обучения[5].

- scikit-learn: библиотека Python с открытым исходным кодом, предоставляющая широкий спектр алгоритмов машинного обучения для классификации, регрессии, кластеризации и других задач.

- PyTorch: библиотека Python с открытым исходным кодом, разработанная Facebook. PyTorch используется в основном для глубокого обучения и предоставляет гибкую и эффективную среду для создания и обучения моделей.

- Keras: высокоуровневый API для TensorFlow, разработанный для упрощения создания и обучения моделей глубокого обучения. Keras предоставляет простой и удобный интерфейс для создания и запуска моделей с использованием TensorFlow.

- Jupyter Notebook: веб-приложение с открытым исходным кодом, которое позволяет пользователям создавать и делиться документами, содержащими живой код, уравнения, визуализации и текстовые пояснения. Jupyter Notebook широко используется для исследований в области науки о данных и разработки моделей машинного обучения[6].

Другие популярные инструменты машинного обучения:

- XGBoost: библиотека с открытым исходным кодом для градиентного бустинга, используемая для классификации и регрессии.

- LightGBM: быстрая и эффективная библиотека градиентного бустинга, используемая для классификации и регрессии.

- CatBoost: библиотека с открытым исходным кодом для градиентного бустинга, оптимизированная для категориальных данных.

- H2O.ai: коммерческая платформа машинного обучения, предоставляющая широкий спектр алгоритмов и инструментов для создания и развертывания моделей машинного обучения.

- Azure Machine Learning: облачная платформа машинного обучения от Microsoft, предоставляющая управляемую среду для разработки, обучения и развертывания моделей машинного обучения.

Эти инструменты машинного обучения предоставляют широкий спектр функциональных возможностей, включая:

- Предварительная обработка и очистка данных
- Выбор признаков и уменьшение размерности
- Обучение и оценка моделей машинного обучения
- Оптимизация гиперпараметров
- Развертывание и мониторинг моделей
- Визуализация и отчетность

Инструменты машинного обучения значительно упрощают и ускоряют процесс разработки и развертывания моделей машинного обучения, позволяя специалистам по данным и разработчикам создавать более точные и эффективные модели для широкого спектра приложений[8].

Глава 2. Проведение анализа базы данных Global Health Observatory (GHO) в Jupiter notebook

2.1 Краткая характеристика Global Health Observatory (GHO)

Global Health Observatory (GHO) — это обсерватория общественного здравоохранения, созданная Всемирной организацией здравоохранения (ВОЗ) для обмена данными о глобальном здравоохранении, включая статистику по странам и информацию о конкретных заболеваниях и мерах по охране здоровья.

GHO была создана примерно в 2010 году на основе Статистической информационной системы ВОЗ, которая была «модернизирована... чтобы предоставить вам больше данных, больше инструментов, больше возможностей для анализа и больше отчётов».

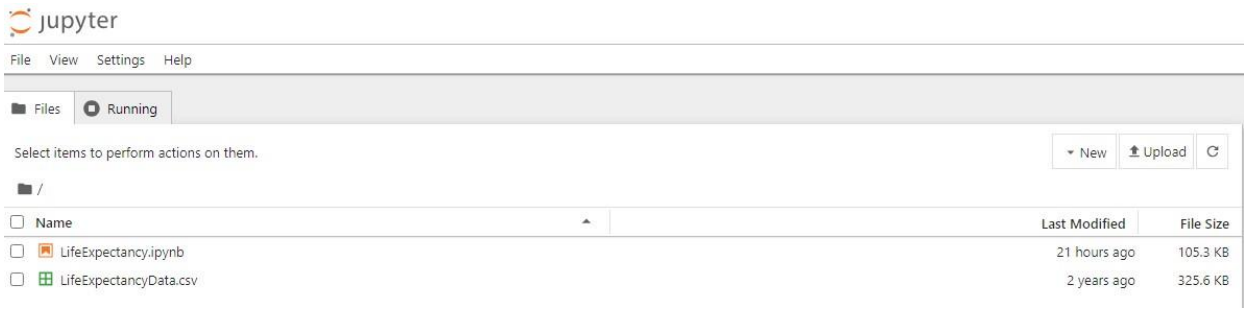
В декабре 2012 года ВОЗ объявила, что вносит улучшения в свою систему GHO, чтобы повысить её доступность и удобство использования «специалистами, такими как статистики, эпидемиологи, экономисты и исследователи в области общественного здравоохранения, а также всеми, кто интересуется глобальным здравоохранением».

2.2 Создание проекта и активация виртуального окружения

Теперь мы приступили к основной нашей практической работе. Вся наша работа будет вестись в Jupiter notebook с автоматическим использованием языка программирования Python.

Создадим на нашем компьютере директорию MyProject. Поместим туда файл LifeExpectancyData.csv, скачанный из сайта www.kaggle.ru. В данном файле содержится информация о факторах иммунизации, факторах смертности, экономических факторах, социальных факторах и других факторах, связанных со здоровьем, которые мы будем анализировать.

Запустим Jupiter notebook, перейдём в нашу вновь созданную директорию и создадим там файл LifeExpectancy.ipynb. В общем, это будет выглядеть следующим образом, что представлено ниже:



Все файлы будут лежать в одной директории MyProject.

2.3 Исследовательский анализ данных в Jupiter notebook

Мы приступаем к исследовательской части нашей работы. Откроем созданный файл LifeExpectancy.ipynb, и всю работу будем ввести в этом файле. У нас будет очень много кода на языке программирования Python. Поэтому в целях удобства и читаемости мы будем вставлять последовательно фото(скриншот) части нашего кода из Jupiter notebook и описывать его.

Начнём наше исследование. Ещё раз вспомним, что целью нашей работы является исследование ожидаемой продолжительность жизни населения разных стран и выяснение причин ее снижения. К этому мы будем двигаться постепенно, анализируя различные показатели и метрики.

Опишем обозначения из файла LifeExpectancyData.csv.

Информация об атрибутах:

"Country": "Country"
"Year": "Year"
"Status": "Developed or Developing status"
"Life_expect": "Life Expectancy in age"
"Adult_Mortality": "dult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population)"
"infant deaths": "Number of Infant Deaths per 1000 population"
"Alcohol": "Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol)"
"percentage expenditure": "Expenditure on health as a percentage of Gross Domestic Product per capita(%)"
"HepatitisB": "Hepatitis B (HepB) immunization coverage among 1-year-olds (%)"
"Measles": "Measles - number of reported cases per 1000 population"
"BMI": "Average Body Mass Index of entire population"
"under-five deaths": "Number of under-five deaths per 1000 population"
"Polio": "Polio (Pol3) immunization coverage among 1-year-olds (%)"
"Total_expenditure": "General government expenditure on health as a percentage of total government expenditure (%)"
"Diphtheria": "Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%)"
"HIV/AIDS": "Deaths per 1 000 live births HIV/AIDS (0-4 years)"
"GDP": "Gross Domestic Product per capita (in USD)"
"Population": "Population of the country"
"thinness_1-19_years": "Prevalence of thinness among children and adolescents for Age 10 to 19 (%)"
"thinness_5-9_years": "Prevalence of thinness among children for Age 5 to 9(%)"
"Income_composition_of_resources": "Human Development Index in terms of income composition of resources (index ranging from 0 to 1)"
"Schooling": "Number of years of Schooling(years)"

«Страна»: «Страна»
«Год»: «Год»
«Статус»: «Развитый или Развивающийся статус»
«Продолжительность жизни»: «Продолжительность жизни в возрасте»
«Смертность взрослых»: «Уровни смертности взрослых обоих полов (вероятность смерти от 15 до 60 лет на 1000 населения)»
«Младенческая смертность»: «Количество младенческих смертей на 1000 населения»
«Алкоголь»: «Учтенное потребление алкоголя на душу населения (15+) (в литрах чистого спирта)»
"Процентные расходы": "Расходы на здравоохранение в процентах от валового внутреннего продукта на душу населения (%)"
«Гепатит В»: «Охват иммунизацией против гепатита В (HepB) детей в возрасте 1 года (%)»
«Корь»: «Корь - количество зарегистрированных случаев на 1000 населения»
«ИМТ»: «Средний индекс массы тела всего населения»
«Смерть детей в возрасте до пяти лет»: «Число смертей детей в возрасте до пяти лет на 1000 населения»
«Полиомиелит»: «Охват иммунизацией против полиомиелита (Pol3) детей в возрасте 1 года (%)»
«Общие расходы»: «Общие государственные расходы на здравоохранение в процентах от общих государственных расходов (%)»
«Дифтерия»: «Охват иммунизацией дифтерийно-столбнячным анатоксином и коклюшем (АКДС-3) детей в возрасте 1 года (%)»
«ВИЧ/СПИД»: «Смертность на 1 000 живорожденных ВИЧ/СПИД (0–4 года)»
«ВВП»: «Валовой внутренний продукт на душу населения (в долларах США)»
«Население»: «Население страны»
«Худоба 1–19 лет»: «Распространенность худобы среди детей и подростков в возрасте от 10 до 19 лет (%)»
«Худоба 5-9 лет»: «Распространенность худобы среди детей в возрасте от 5 до 9 лет (%)»
«Доходная структура ресурсов»: «Индекс человеческого развития по доходной структуре ресурсов (индекс от 0 до 1)»
«Обучение»: «Количество лет обучения (лет)»

Импортируем необходимые библиотеки.

Импортируем исследуемый датасет LifeExpectancyData.csv, в котором представлена информация о различных факторах, влияющих на ожидаемую продолжительность жизни.

[4]:
```
data = pd.read_csv('LifeExpectancyData.csv')
data
```

[4]:

	Country	Year	Status	Life_expect	Adult_Mortality	infant_deaths	Alcohol	percentage_expenditure	HepatitisB	Measles	...	Polio	Total_expenditure	Di
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	8.16	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	8.18	
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	8.13	
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	8.52	
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	7.87	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2933	Zimbabwe	2004	Developing	44.3	723.0	27	4.36	0.000000	68.0	31	...	67.0	7.13	
2934	Zimbabwe	2003	Developing	44.5	715.0	26	4.06	0.000000	7.0	998	...	7.0	6.52	
2935	Zimbabwe	2002	Developing	44.8	73.0	25	4.43	0.000000	73.0	304	...	73.0	6.53	
2936	Zimbabwe	2001	Developing	45.3	686.0	25	1.72	0.000000	76.0	529	...	76.0	6.16	
2937	Zimbabwe	2000	Developing	46.0	665.0	24	1.68	0.000000	79.0	1483	...	78.0	7.10	

2938 rows × 22 columns

[4]:
```
data = pd.read_csv('LifeExpectancyData.csv')
data
```

[4]:

HepatitisB	Measles	...	Polio	Total_expenditure	Diphtheria	HIV_AIDS	GDP	Population	thinness_1-19_years	thinness_5-9_years	Income_composition_of_resources	Schooling
65.0	1154	...	6.0	8.16	65.0	0.1	584.259210	33736494.0	17.2	17.3	0.479	10.1
62.0	492	...	58.0	8.18	62.0	0.1	612.696514	327582.0	17.5	17.5	0.476	10.0
64.0	430	...	62.0	8.13	64.0	0.1	631.744976	31731688.0	17.7	17.7	0.470	9.9
67.0	2787	...	67.0	8.52	67.0	0.1	669.959000	3696958.0	17.9	18.0	0.463	9.8
68.0	3013	...	68.0	7.87	68.0	0.1	63.537231	2978599.0	18.2	18.2	0.454	9.5
...	...	...	...	...	...	...	...	...	...	...	...	...
68.0	31	...	67.0	7.13	65.0	33.6	454.366654	12777511.0	9.4	9.4	0.407	9.2
7.0	998	...	7.0	6.52	68.0	36.7	453.351155	12633897.0	9.8	9.9	0.418	9.5
73.0	304	...	73.0	6.53	71.0	39.8	57.348340	125525.0	1.2	1.3	0.427	10.0
76.0	529	...	76.0	6.16	75.0	42.1	548.587312	12366165.0	1.6	1.7	0.427	9.8
79.0	1483	...	78.0	7.10	78.0	43.5	547.358878	12222251.0	11.0	11.2	0.434	9.8

Проверим типы данных и есть ли отсутствующие данные в датасете.

```
[6]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   Country                          2938 non-null   object 
 1   Year                             2938 non-null   int64  
 2   Status                           2938 non-null   object 
 3   Life_expect                      2928 non-null   float64
 4   Adult_Mortality                  2928 non-null   float64
 5   infant_deaths                    2938 non-null   int64  
 6   Alcohol                          2744 non-null   float64
 7   percentage_expenditure           2938 non-null   float64
 8   HepatitisB                       2385 non-null   float64
 9   Measles                          2938 non-null   int64  
 10  BMI                              2904 non-null   float64
 11  under_five_deaths                2938 non-null   int64  
 12  Polio                            2919 non-null   float64
 13  Total_expenditure                2712 non-null   float64
 14  Diphtheria                       2919 non-null   float64
 15  HIV_AIDS                         2938 non-null   float64
 16  GDP                              2490 non-null   float64
 17  Population                       2286 non-null   float64
 18  thinness_1-19_years              2904 non-null   float64
 19  thinness_5-9_years               2904 non-null   float64
 20  Income_composition_of_resources  2771 non-null   float64
 21  Schooling                        2775 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

Видно, что пропусков и отсутствующих данных не имеется.

Проведём тщательное исследование данных. Убедимся, что в этих данных нет дубликатов

```
[7]: data.duplicated().any()
```

```
[7]: False
```

Выведем основные статистические характеристики.

```
[8]: data.describe()
```

[8]:

	Year	Life_expect	Adult_Mortality	infant_deaths	Alcohol	percentage_expenditure	HepatitisB	Measles	BMI	under_five_deaths	
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.000000	2938.000000	2904.000000	2938.000000	2919
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.940461	2419.592240	38.321247	42.035739	82
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.070016	11467.272489	20.044034	160.445548	23
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.000000	0.000000	1.000000	0.000000	3
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.000000	0.000000	19.300000	0.000000	78
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.000000	17.000000	43.500000	4.000000	93
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.000000	360.250000	56.200000	28.000000	97
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000	212183.000000	87.300000	2500.000000	99

```
[8]: data.describe()
```

[8]:

deaths	Polio	Total_expenditure	Diphtheria	HIV_AIDS	GDP	Population	thinness_1-19_years	thinness_5-9_years	Income_composition_of_resources	Schooling
000000	2919.000000	2712.00000	2919.000000	2938.000000	2490.000000	2.286000e+03	2904.000000	2904.000000	2771.000000	2775.000000
035739	82.550188	5.93819	82.324084	1.742103	7483.158469	1.275338e+07	4.839704	4.870317	0.627551	11.992793
445548	23.428046	2.49832	23.716912	5.077785	14270.169342	6.101210e+07	4.420195	4.508882	0.210904	3.358920
000000	3.000000	0.37000	2.000000	0.100000	1.681350	3.400000e+01	0.100000	0.100000	0.000000	0.000000
000000	78.000000	4.26000	78.000000	0.100000	463.935626	1.957932e+05	1.600000	1.500000	0.493000	10.100000
000000	93.000000	5.75500	93.000000	0.100000	1766.947595	1.386542e+06	3.300000	3.300000	0.677000	12.300000
000000	97.000000	7.49250	97.000000	0.800000	5910.806335	7.420359e+06	7.200000	7.200000	0.779000	14.300000
000000	99.000000	17.60000	99.000000	50.600000	119172.741800	1.293859e+09	27.700000	28.600000	0.948000	20.700000

Видно, что в некоторых колонках есть отсутствующие значения.

Перепроверим данные. Выведем количество строк и столбцов.

```
[11]: print("Number of Rows:", data.shape[0])
      print("Number of Columns:", data.shape[1])
```

```
Number of Rows: 2938
Number of Columns: 22
```

Проверим данные на пропущенные значения.

```
[13]: data.isnull().sum()
```

```
[13]: Country                          0
      Year                             0
      Status                           0
      Life_expect                     10
      Adult_Mortality                 10
      infant_deaths                    0
      Alcohol                        194
      percentage_expenditure           0
      HepatitisB                     553
      Measles                          0
      BMI                             34
      under_five_deaths                0
      Polio                           19
      Total_expenditure              226
      Diphtheria                      19
      HIV_AIDS                         0
      GDP                            448
      Population                     652
      thinness_1-19_years             34
      thinness_5-9_years              34
      Income_composition_of_resources    167
      Schooling                      163
      dtype: int64
```

Обрабатываем пропущенные значения, заполняя пропуски средним значением по каждому столбцу.

```
[14]: imputer = SimpleImputer(missing_values=np.nan, strategy='mean', fill_value=None)
data['Life_expect']=imputer.fit_transform(data[['Life_expect']])
data['Adult_Mortality']=imputer.fit_transform(data[['Adult_Mortality']])
data['Alcohol']=imputer.fit_transform(data[['Alcohol']])
data['HepatitisB']=imputer.fit_transform(data[['HepatitisB']])
data['BMI']=imputer.fit_transform(data[['BMI']])
data['Polio']=imputer.fit_transform(data[['Polio']])
data['Total_expenditure']=imputer.fit_transform(data[['Total_expenditure']])
data['Diphtheria']=imputer.fit_transform(data[['Diphtheria']])
data['GDP']=imputer.fit_transform(data[['GDP']])
data['Population']=imputer.fit_transform(data[['Population']])
data['thinness_1-19_years']=imputer.fit_transform(data[['thinness_1-19_years']])
data['thinness_5-9_years']=imputer.fit_transform(data[['thinness_5-9_years']])
data['Income_composition_of_resources']=imputer.fit_transform(data[['Income_composition_of_resources']])
data['Schooling']=imputer.fit_transform(data[['Schooling']])
```

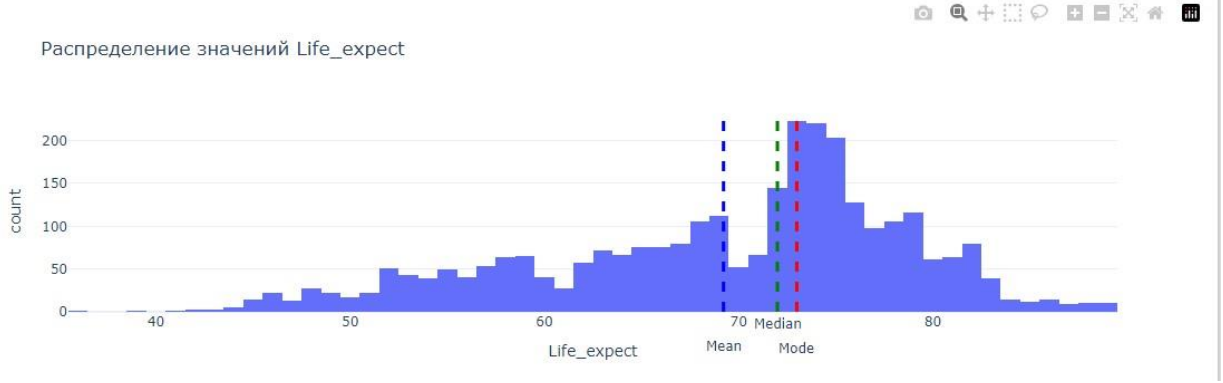
Еще раз перепроверим данные.

```
[15]: data.isnull().sum()
```

```
[15]: Country                          0
      Year                             0
      Status                           0
      Life_expect                      0
      Adult_Mortality                  0
      infant_deaths                    0
      Alcohol                          0
      percentage_expenditure           0
      HepatitisB                       0
      Measles                          0
      BMI                              0
      under_five_deaths                0
      Polio                            0
      Total_expenditure                0
      Diphtheria                       0
      HIV_AIDS                         0
      GDP                              0
      Population                       0
      thinness_1-19_years              0
      thinness_5-9_years               0
      Income_composition_of_resources  0
      Schooling                        0
      dtype: int64
```

Мы получили все нули во всех колонках, что означает отсутствующих значений нет. Теперь перейдём к визуализации данных. Построим гистограмму с распределением значений колонки Life_expect - это количество лет ожидаемой продолжительности жизни.

[6]:
```
fig = px.histogram(data, x='Life_expect', template='plotly_white', title = 'Распределение значений Life_expect')

median = data['Life_expect'].median()
mean = data['Life_expect'].mean()
mode = data['Life_expect'].mode()[0]


fig.add_vline(x=median, line_width=3, line_dash="dash", line_color="green")
fig.add_vline(x=mean, line_width=3, line_dash="dash", line_color="blue")
fig.add_vline(x=mode, line_width=3, line_dash="dash", line_color="red")

fig.add_annotation(x=median, y=0, xref="x", yref="paper", text="Median", showarrow=False, yshift=-20)
fig.add_annotation(x=mean, y=0, xref="x", yref="paper", text="Mean", showarrow=False, yshift=-40)
fig.add_annotation(x=mode, y=0, xref="x", yref="paper", text="Mode", showarrow=False, yshift=-42)

fig.show()
```

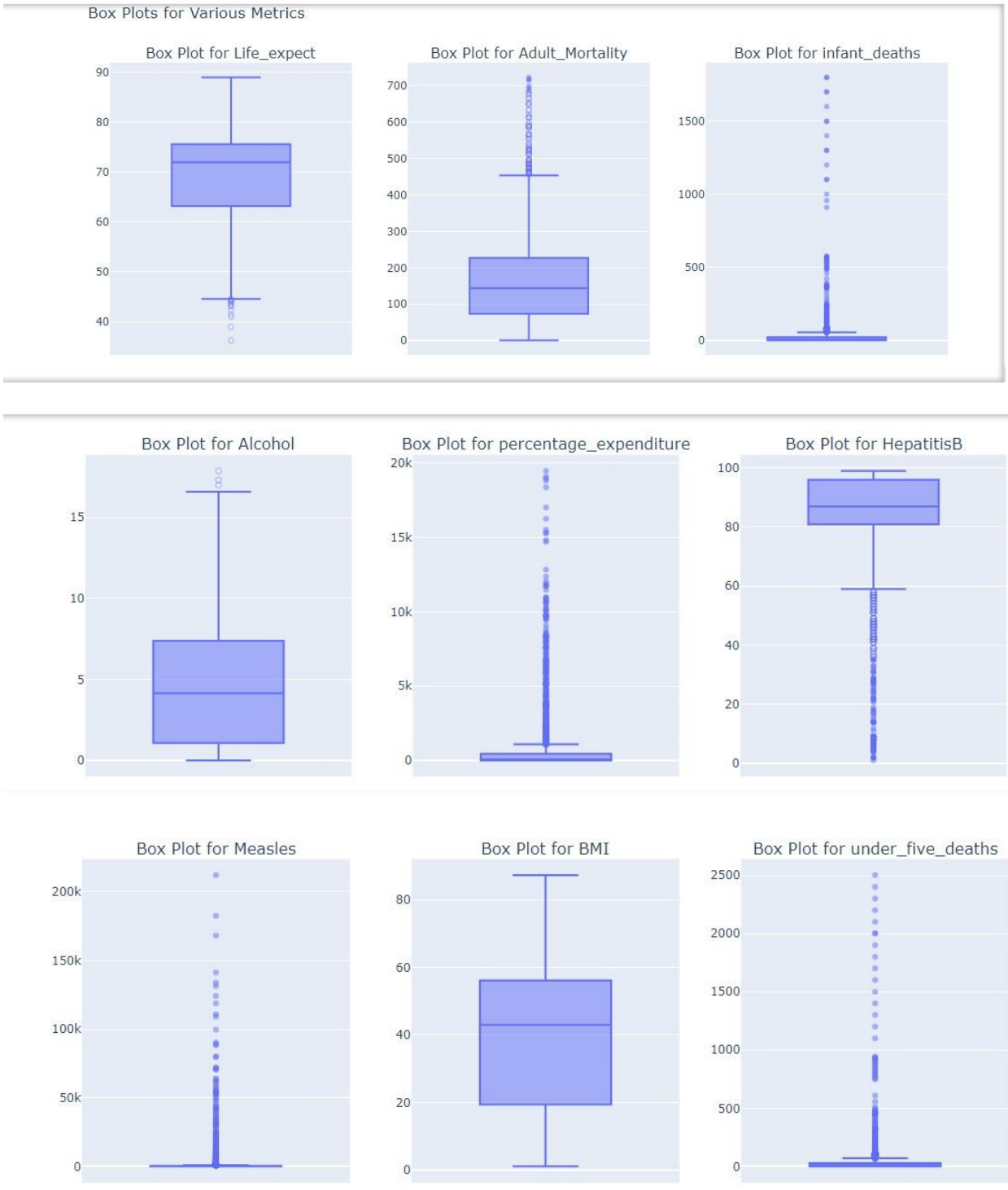


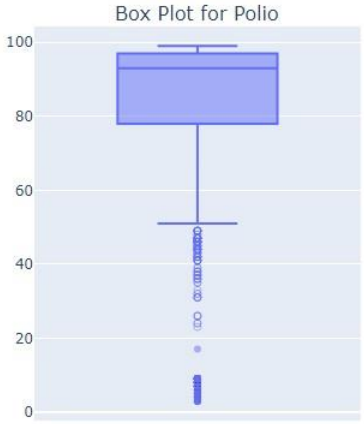
Видно, что средняя ожидаемая продолжительность жизни человека – такой же средний показатель, как и температура по больнице. Кто-то доживает до глубокой старости, кто-то умирает молодым. В среднем в мире получается чуть больше 69 лет. А также видно, как мало людей (статистически говоря) доживают до 85 лет и что после 85 лет ожидаемая продолжительность жизни человека в мире резко снижается.

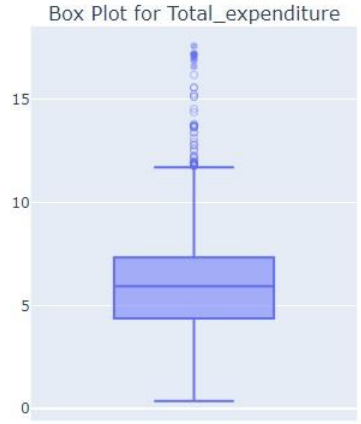
Отобразим сетку из subplot-ов, в каждом из которых отображается box-plot для одного из числовых столбцов датасета. Box-plot позволяет быстро визуализировать и сравнить распределение значений для нескольких числовых показателей одновременно. Это может быть полезно для выявления выбросов, асимметрии, центральной тенденции и разброса данных в разных столбцах. Выбираем только колонки с численным типом.

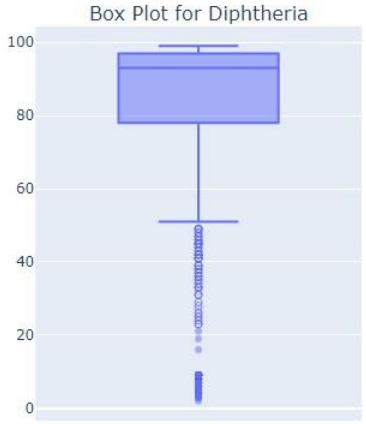
[7]:
```
numerical_columns = data.select_dtypes(exclude = object).columns.tolist()
numerical_columns.remove("Year") # удаляю колонку "Year", т.к она имеет категориальный признак
```

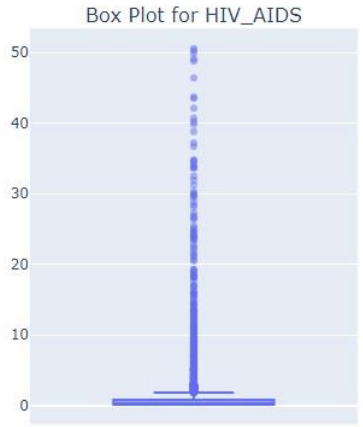
[9]:
```
cols = 3  # Три колонки для подграфиков
rows = (len(numerical_columns) + cols - 1) // cols  # Количество строк
fig = make_subplots(rows=rows, cols=cols, subplot_titles=[f'Box Plot for {col}' for col in numerical_columns])

for index, column in enumerate(numerical_columns):
    row = index // cols + 1
    col = index % cols + 1
    temp_fig = px.box(data, y=column, points='suspectedoutliers', title=f'Box Plot for {column}')
    for trace in temp_fig.data:
        trace.marker.opacity = 0.5  # Устанавливаем прозрачность точек
        fig.add_trace(trace, row=row, col=col)

# Регулировка макета subplot-ов
fig.update_layout(height=600 * rows, showlegend=False, title_text="Box Plots for Various Metrics")

# Отображение фигуры
fig.show()
```

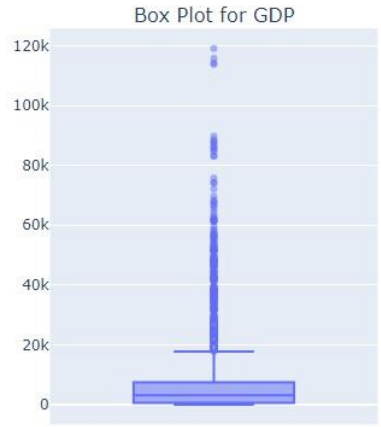


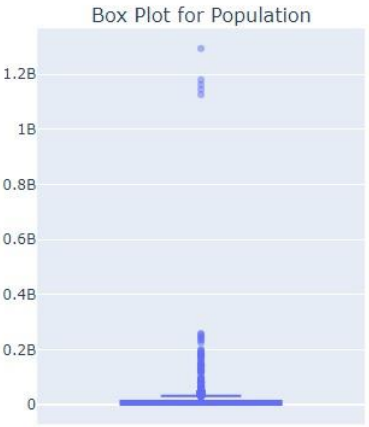

Box Plot for Polio
100
80
60
40
20
0

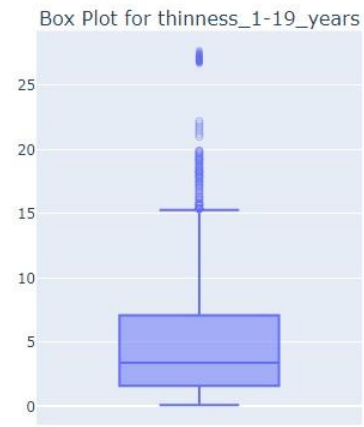

Box Plot for Total_expenditure
15
10
5
0

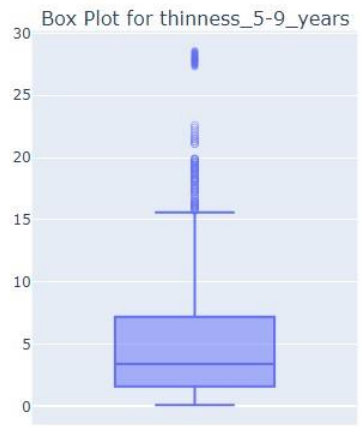

Box Plot for Diphtheria
100
80
60
40
20
0

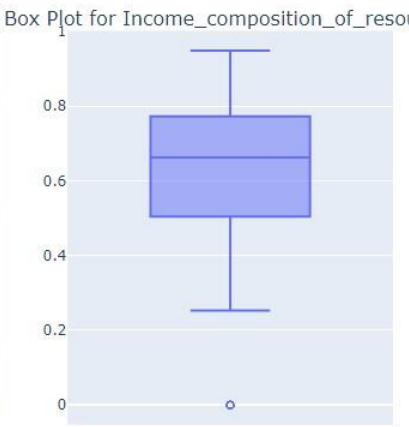

Box Plot for HIV_AIDS
50
40
30
20
10
0

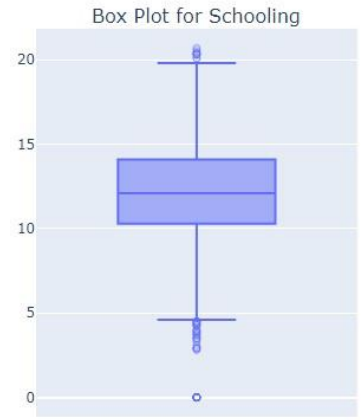

Box Plot for GDP
120k
100k
80k
60k
40k
20k
0


Box Plot for Population
1.2B
1B
0.8B
0.6B
0.4B
0.2B
0


Box Plot for thinness_1-19_years
25
20
15
10
5
0


Box Plot for thinness_5-9_years
30
25
20
15
10
5
0


Box Plot for Income_composition_of_resources
1
0.8
0.6
0.4
0.2
0



Посмотрим на страны с минимальными уровнями образования и доходности ресурсов (в основном это африканские страны либо островные государства).

[9]:
```
rows_with_min = data[(data['Schooling'] == data['Schooling'].min()) &
 (data['Income_composition_of_resources'] == data['Income_composition_of_resources'].min())]

rows_with_min
```

[9]:

	Country	Year	Status	Life_expect	Adult_Mortality	infant_deaths	Alcohol	percentage_expenditure	HepatitisB	Measles	...	Polio	Total_expenditure	D
74	Antigua and Barbuda	2005	Developing	74.6	147.0	0	8.15	1455.608186	99.0	0	...	98.0	4.41	
75	Antigua and Barbuda	2004	Developing	74.4	149.0	0	7.28	22.862952	97.0	0	...	97.0	4.21	
76	Antigua and Barbuda	2003	Developing	74.2	151.0	0	7.16	1158.065259	99.0	0	...	99.0	4.53	
77	Antigua and Barbuda	2002	Developing	74.0	153.0	0	7.21	927.407585	99.0	0	...	93.0	4.41	
78	Antigua and Barbuda	2001	Developing	73.8	154.0	0	7.51	163.767698	96.0	0	...	99.0	4.48	
79	Antigua and Barbuda	2000	Developing	73.6	156.0	0	7.27	1127.743470	NaN	0	...	96.0	4.13	
335	Bosnia and Herzegovina	2000	Developing	74.6	116.0	0	3.64	165.616864	NaN	43	...	87.0	7.90	
849	Equatorial Guinea	2000	Developing	52.7	336.0	3	4.46	14.954513	NaN	0	...	41.0	2.73	
1714	Micronesia (Federated States of)	2000	Developing	67.0	185.0	0	2.23	0.000000	87.0	0	...	85.0	7.88	
1744	Montenegro	2003	Developing	73.5	134.0	0	0.01	495.078296	NaN	0	...	NaN	8.91	
1745	Montenegro	2002	Developing	73.4	136.0	0	0.01	36.480240	NaN	0	...	NaN	8.33	
1746	Montenegro	2001	Developing	73.3	136.0	0	0.01	33.669814	NaN	0	...	NaN	8.23	
1747	Montenegro	2000	Developing	73.0	144.0	0	0.01	274.547260	NaN	0	...	NaN	7.32	
2414	South Sudan	2010	Developing	55.0	359.0	27	NaN	0.000000	NaN	0	...	NaN	NaN	
2415	South Sudan	2009	Developing	54.3	369.0	27	NaN	0.000000	NaN	0	...	NaN	NaN	
2416	South Sudan	2008	Developing	53.6	377.0	27	NaN	0.000000	NaN	0	...	NaN	NaN	
2417	South Sudan	2007	Developing	53.1	381.0	27	NaN	0.000000	NaN	0	...	NaN	NaN	

[9]:
```
rows_with_min = data[(data['Schooling'] == data['Schooling'].min()) &
 (data['Income_composition_of_resources'] == data['Income_composition_of_resources'].min())]

rows_with_min
```

[9]:

HepatitisB	Measles	...	Polio	Total_expenditure	Diphtheria	HIV_AIDS	GDP	Population	thinness_1-19_years	thinness_5-9_years	Income_composition_of_resources	Schooling
99.0	0	...	98.0	4.41	99.0	0.1	11371.938950	NaN	3.5	3.4	0.0	0.0
97.0	0	...	97.0	4.21	97.0	0.1	1352.837400	NaN	3.5	3.4	0.0	0.0
99.0	0	...	99.0	4.53	99.0	0.1	9739.825560	NaN	3.5	3.5	0.0	0.0
99.0	0	...	93.0	4.41	98.0	0.1	9386.716452	NaN	3.6	3.5	0.0	0.0
96.0	0	...	99.0	4.48	97.0	0.1	9358.154162	NaN	3.6	3.5	0.0	0.0
NaN	0	...	96.0	4.13	95.0	0.1	9875.161736	NaN	3.7	3.6	0.0	0.0
NaN	43	...	87.0	7.90	85.0	0.1	1461.755200	376676.0	3.3	3.2	0.0	0.0
NaN	0	...	41.0	2.73	34.0	1.9	172.684910	614323.0	1.7	1.6	0.0	0.0
87.0	0	...	85.0	7.88	85.0	0.1	NaN	NaN	0.3	0.3	0.0	0.0
NaN	0	...	NaN	8.91	NaN	0.1	2789.173500	612267.0	2.4	2.4	0.0	0.0
NaN	0	...	NaN	8.33	NaN	0.1	216.243274	69828.0	2.5	2.5	0.0	0.0
NaN	0	...	NaN	8.23	NaN	0.1	199.583957	67389.0	2.5	2.6	0.0	0.0
NaN	0	...	NaN	7.32	NaN	0.1	1627.428930	6495.0	2.6	2.7	0.0	0.0
NaN	0	...	NaN	NaN	NaN	4.0	1562.239346	167192.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	4.2	1264.789980	967667.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	4.2	1678.711862	9263136.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	4.2	NaN	88568.0	NaN	NaN	0.0	0.0

79	Antigua and Barbuda	2000	Developing	73.6	156.0	0	7.27	1127.743470	NaN	0	...	96.0	4.13
335	Bosnia and Herzegovina	2000	Developing	74.6	116.0	0	3.64	165.616864	NaN	43	...	87.0	7.90
849	Equatorial Guinea	2000	Developing	52.7	336.0	3	4.46	14.954513	NaN	0	...	41.0	2.73
1714	Micronesia (Federated States of)	2000	Developing	67.0	185.0	0	2.23	0.000000	87.0	0	...	85.0	7.88
1744	Montenegro	2003	Developing	73.5	134.0	0	0.01	495.078296	NaN	0	...	NaN	8.91
1745	Montenegro	2002	Developing	73.4	136.0	0	0.01	36.480240	NaN	0	...	NaN	8.33
1746	Montenegro	2001	Developing	73.3	136.0	0	0.01	33.669814	NaN	0	...	NaN	8.23
1747	Montenegro	2000	Developing	73.0	144.0	0	0.01	274.547260	NaN	0	...	NaN	7.32
2414	South Sudan	2010	Developing	55.0	359.0	27	NaN	0.000000	NaN	0	...	NaN	NaN
2415	South Sudan	2009	Developing	54.3	369.0	27	NaN	0.000000	NaN	0	...	NaN	NaN
2416	South Sudan	2008	Developing	53.6	377.0	27	NaN	0.000000	NaN	0	...	NaN	NaN
2417	South Sudan	2007	Developing	53.1	381.0	27	NaN	0.000000	NaN	0	...	NaN	NaN
2418	South Sudan	2006	Developing	52.5	383.0	28	NaN	0.000000	NaN	0	...	NaN	NaN
2419	South Sudan	2005	Developing	51.9	383.0	28	NaN	0.000000	NaN	0	...	NaN	NaN
2420	South Sudan	2004	Developing	51.4	383.0	29	NaN	0.000000	NaN	0	...	NaN	NaN
2421	South Sudan	2003	Developing	58.0	383.0	29	NaN	0.000000	NaN	0	...	NaN	NaN
2422	South Sudan	2002	Developing	52.0	382.0	30	NaN	0.000000	NaN	0	...	NaN	NaN
2423	South Sudan	2001	Developing	49.6	381.0	30	NaN	0.000000	NaN	0	...	NaN	NaN
2424	South Sudan	2000	Developing	48.9	38.0	31	NaN	0.000000	NaN	0	...	NaN	NaN
2616	Timor-Leste	2000	Developing	58.7	276.0	3	0.50	49.069672	NaN	0	...	NaN	3.26
2712	Turkmenistan	2000	Developing	63.8	224.0	7	2.90	88.243635	NaN	113	...	98.0	3.94

26 rows × 22 columns

NaN	0	...	96.0	4.13	95.0	0.1	9875.161736	NaN	3.7	3.6	0.0	0.0
NaN	43	...	87.0	7.90	85.0	0.1	1461.755200	376676.0	3.3	3.2	0.0	0.0
NaN	0	...	41.0	2.73	34.0	1.9	172.684910	614323.0	1.7	1.6	0.0	0.0
87.0	0	...	85.0	7.88	85.0	0.1	NaN	NaN	0.3	0.3	0.0	0.0
NaN	0	...	NaN	8.91	NaN	0.1	2789.173500	612267.0	2.4	2.4	0.0	0.0
NaN	0	...	NaN	8.33	NaN	0.1	216.243274	69828.0	2.5	2.5	0.0	0.0
NaN	0	...	NaN	8.23	NaN	0.1	199.583957	67389.0	2.5	2.6	0.0	0.0
NaN	0	...	NaN	7.32	NaN	0.1	1627.428930	6495.0	2.6	2.7	0.0	0.0
NaN	0	...	NaN	NaN	NaN	4.0	1562.239346	167192.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	4.2	1264.789980	967667.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	4.2	1678.711862	9263136.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	4.2	NaN	88568.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	4.1	NaN	8468152.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	3.9	NaN	818877.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	3.8	NaN	7787655.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	3.5	NaN	751642.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	3.3	NaN	7237276.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	3.0	NaN	6974442.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	NaN	NaN	2.7	NaN	67656.0	NaN	NaN	0.0	0.0
NaN	0	...	NaN	3.26	NaN	0.1	422.286330	87167.0	12.2	12.2	0.0	0.0
NaN	113	...	98.0	3.94	97.0	0.1	643.175180	4516131.0	3.6	3.6	0.0	0.0

Организация Объединенных Наций делит страны на две основные категории: развитые и развивающиеся страны. Классификация стран основана на экономическом статусе, таком как ВВП, ВНП, доход на душу населения, индустриализация, уровень жизни и т.д.

Развитые страны относятся к суверенным государствам, экономика которых значительно прогрессировала и обладает большой технологической инфраструктурой по сравнению с другими странами.

Страны, которые переживают начальный уровень промышленного развития наряду с низким доходом на душу населения, известны как развивающиеся страны. В этих странах нет здоровой и безопасной среды для жизни, низкий валовой внутренний продукт, высокий уровень неграмотности, неустойчивый государственный долг, неравное распределение доходов, высокий уровень смертности и рождаемости, недоедание как матери, так и ребенка, что связано с высокой младенческой смертностью, плохими условиями жизни, высоким уровнем безработицы и бедности.

Развитые страны предоставляют свободную, здоровую и безопасную атмосферу для жизни, тогда как в развивающихся странах этого не хватает.

Посмотрим на уровень жизни в зависимости от статуса развития страны.

```
[10]: fig = px.box(data, x= 'Status', y='Life_expect', color='Status',template='plotly_white',title='Life expectancy Based on Countries status')
      fig.show()
```

По графику видно, что страны, где отмечается низкая продолжительность жизни, экономически не развиты.

Худшие показатели неизменно представляют страны Африки. Наличие огромного числа проблем сказывается на цифрах. Зараженность вирусом иммунодефицита в ряде бедных африканских стран составляет 20-25% населения. Отсутствие полноценной системы медицинского обслуживания — является причиной высокого уровня младенческой и детской смертности.

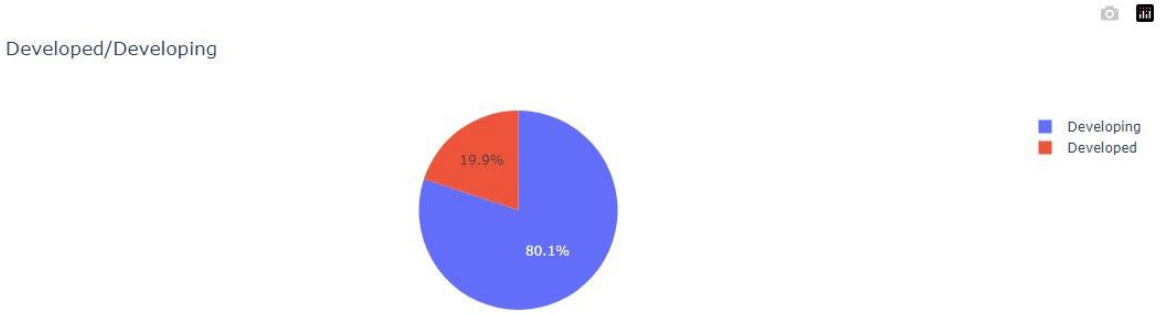
В экономически развитых странах количество населения меньше, но ожидаемая продолжительность жизни выше, по сравнению с развивающимися странами.

Определим количество стран по их статусу (развитая или развивающаяся страна).

```
[11]: data['Status'].value_counts()

[11]: Status
      Developing    2426
      Developed      512
      Name: count, dtype: int64
```

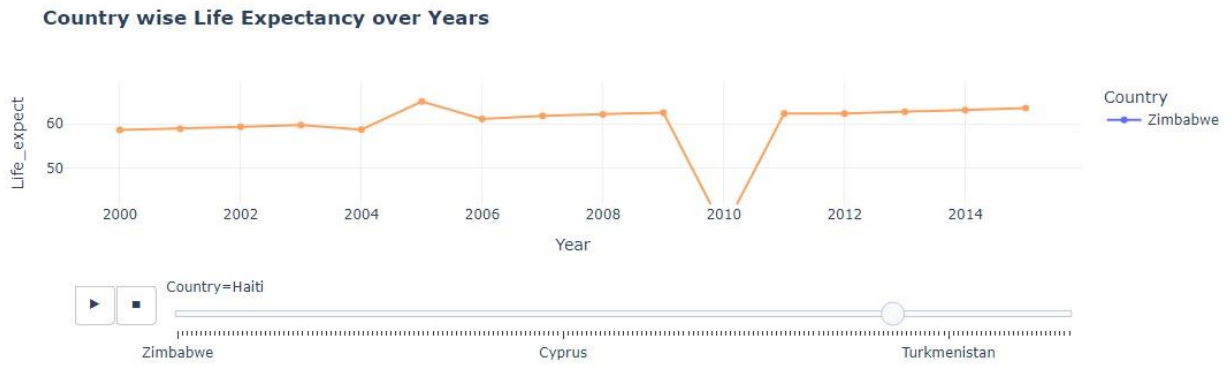
Создадим круговую диаграмму, показывающую соотношение развитых и развивающихся стран.

```
[12]: fig = px.pie(data, values='Life_expect', names='Status',template='plotly_white',title='Developed/Developing')
fig.show()
```



Построим линейную диаграмму, отображающую по отдельности каждую страну и ее уровень ожидаемой продолжительности жизни в зависимости от года.

```
[13]: fig=px.line(data.sort_values(by='Year'),x='Year',y='Life_expect',animation_frame='Country',animation_group='Year',
            color='Country',markers=True,template='plotly_white',title='<b> Country wise Life Expectancy over Years')
fig.show()
```



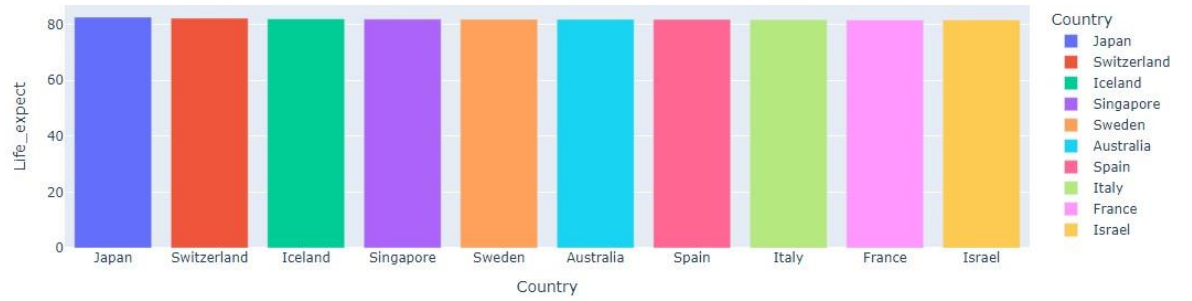
По графику видно, например, что в 2010 году уровень ожидаемой продолжительности жизни в Гаити сократился почти в 2 раза по сравнению с предыдущим. Это связано с произошедшим 12 января землетрясением, в результате которого погибло по разным оценкам от ста тысяч до полмиллиона человек.

Страны с самой низкой ожидаемой продолжительностью жизни.

```
[14]: dataGroup = data.groupby('Country')['Life_expect'].median().sort_values(ascending = True).reset_index().head(10)

      px.bar(data_frame = dataGroup, x='Country', y = 'Life_expect', color='Country').show()
```

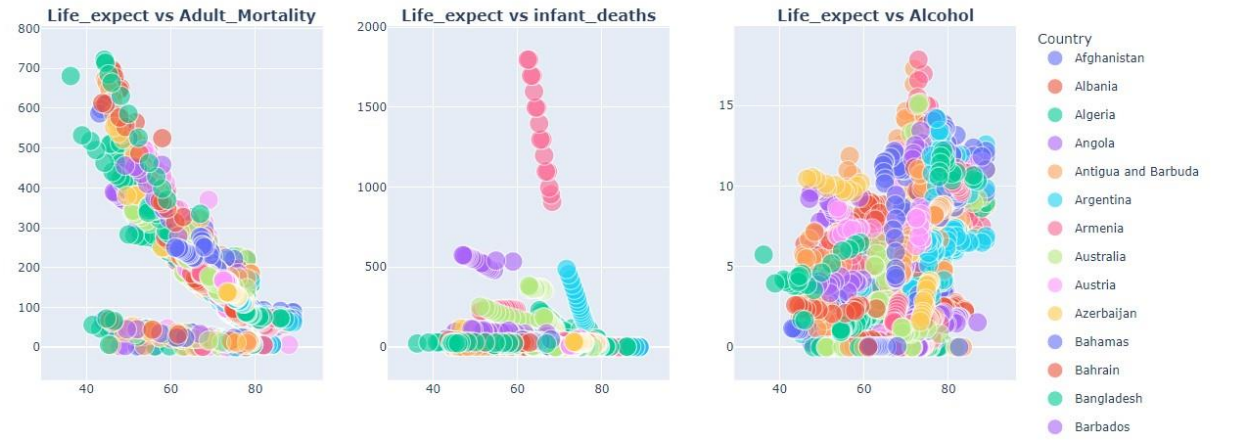
Страны с самой высокой ожидаемой продолжительностью жизни.

```
[15]: dataGroup = data.groupby('Country')['Life_expect'].median().sort_values(ascending = False).reset_index().head(10)

      px.bar(data_frame = dataGroup, x='Country', y = 'Life_expect', color='Country').show()
```

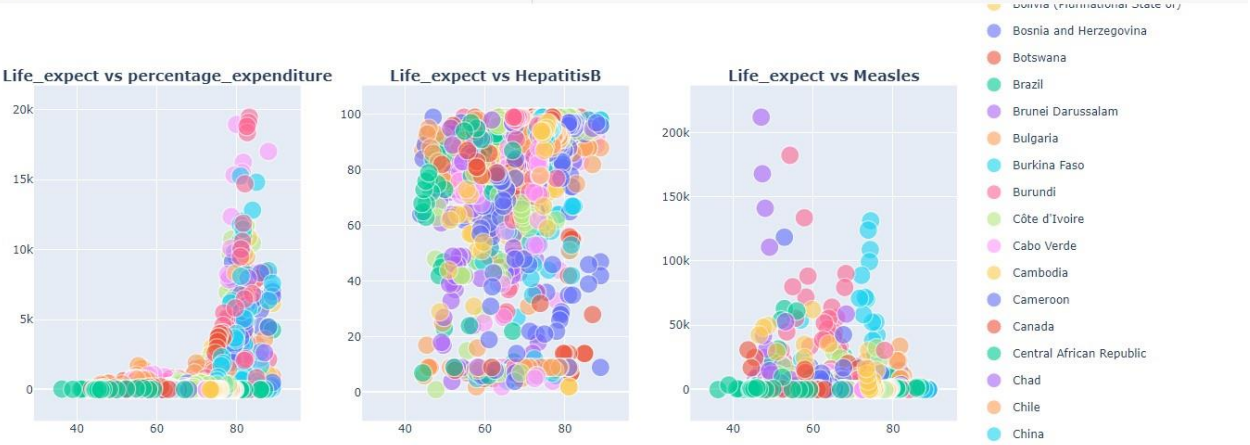


Построим диаграмму рассеивания для колонки 'Life_expect'(ожидаемая продолжительность жизни) со всеми другими численными колонками.

```
# Отображаем диаграммы
fig.show()
```




```
# Отображаем диаграммы
fig.show()
```




```
# Отображаем диаграммы
fig.show()
```

Приведенный выше график объясняет следующие тенденции: Ожидаемая продолжительность жизни снижается с увеличением детской смертности. Страны с высоким потреблением алкоголя отличаются высокой продолжительностью жизни. Между ИМТ (средний индекс массы тела всего населения) и ожидаемой продолжительностью жизни существует линейная зависимость. Между ВВП и ожидаемой продолжительностью жизни существует

четкая линейная зависимость. Страны с высокой численностью населения имеют несколько более низкую ожидаемую продолжительность жизни. По мере увеличения общего дохода страны ожидаемая продолжительность жизни также увеличивается. (Если вы богаты, то, как ожидается, проживете долгую жизнь). И последнее, но не менее важное: образование, как и ожидалось, влияет на продолжительность жизни.

Построим и оценим корреляционную матрицу для всех значений.

```
[24]: plt.figure(figsize = (16,8))
heatMap = sns.heatmap(data.select_dtypes(exclude = object).drop('Year', axis=1).corr(),
            annot = True, fmt = ".2f", linewidths = 0.2,cmap='BuPu')
heatMap.set_title('Correlation matrix', pad=10)
plt.show()
```

В итоге получим следующий рисунок:

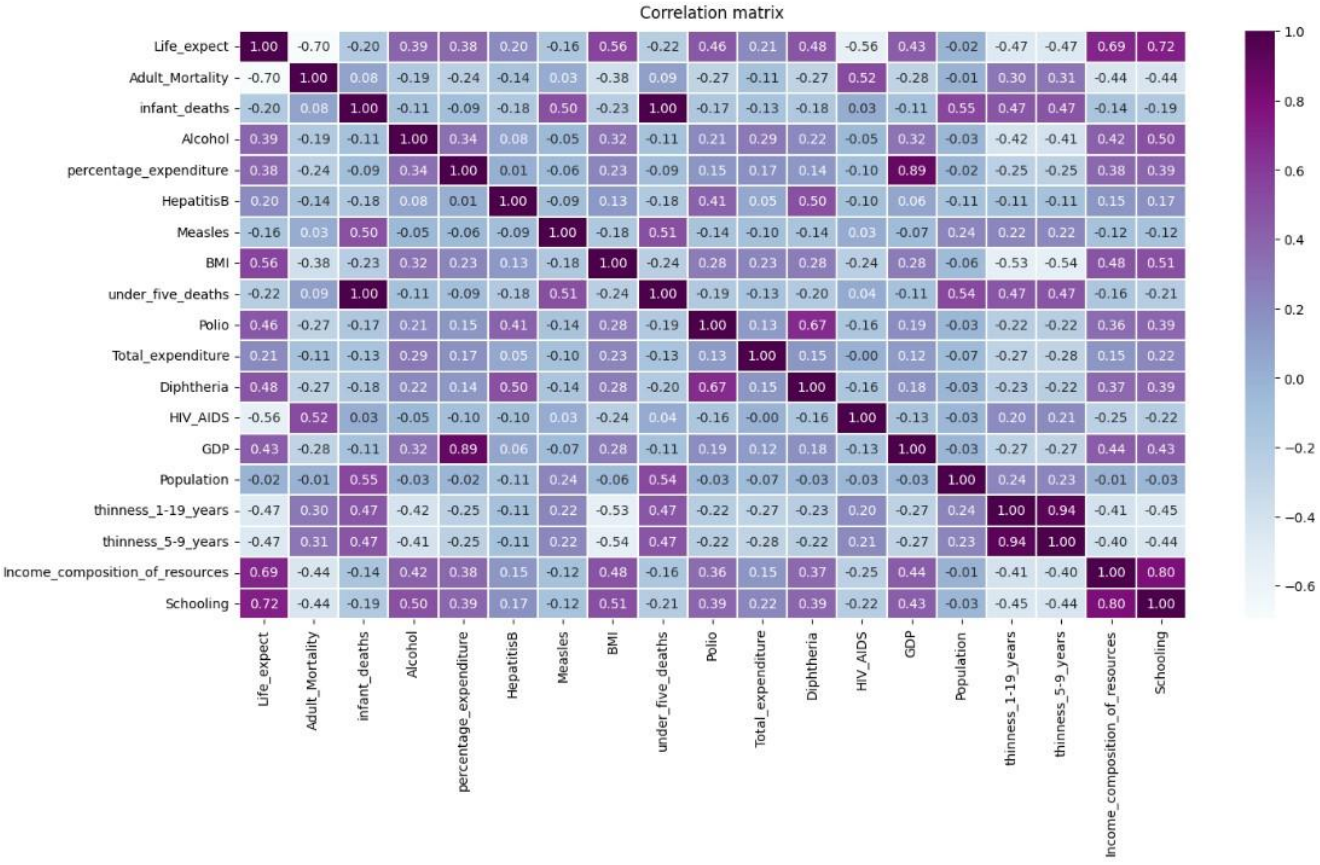


График корреляции показывает, что рост ожидаемой продолжительности жизни состоит из снижения смертности в раннем возрасте, а затем снижения смертности в пожилом возрасте.

Экономический рост оказывает наибольшее воздействие, если он используется для финансирования соответствующих социальных услуг, таких как здравоохранение, эпидемиологическая защита и базовое образование. К числу других фундаментальных факторов, способствующих повышению ожидаемой продолжительности жизни, относятся улучшение питания.

Расходы на здравоохранение также являются намного более важными в развивающихся экономиках, где существующие медицинские учреждения настолько просты, что каждый дополнительный доллар, вливаемый в сектор, может иметь значительный эффект. Некоторые инфекционные (ВИЧ, корь и т.д.) заболевания требуют внимания со стороны правительства, с тем чтобы их можно было контролировать и/или искоренять. Младенцы и маленькие дети, как правило, очень уязвимы к таким заболеваниям, и уровень младенческой смертности особенно сильно коррелирует с общей продолжительностью жизни в развивающихся странах.

Глава 3. Применение моделей машинного обучения для прогнозирования ожидаемой продолжительности жизни

3.1 Выбор модели машинного обучения

Теперь мы подошли к основной части нашей работы – создание предиктивной модели машинного обучения. Возникает невольный вопрос: “Какую модель следует выбирать?”

Сделаем небольшое напоминание, что при выборе модели машинного обучения необходимо учитывать следующие факторы:

1. Цель задачи: определить, какую задачу мы хотим решить (классификация, регрессия, кластеризация и т. д.).

2. Объем и характер данных: учесть размер выборки, количество признаков, их тип (категориальные или числовые) и распределение.

3. Время обучения и предсказания: учесть требования к скорости работы модели.

4. Интерпретируемость: если важно понимать, как модель принимает решения, то лучше выбрать модели с хорошей интерпретируемостью.

5. Регуляризация: если имеется много признаков, которые могут быть нерелевантными, рассмотреть модели с возможностью регуляризации.

6. Разреженность данных: если есть разреженные данные (многие нулевые значения), выбрать модели, которые умеют с ними хорошо работать.

7. Проверка модели: использовать кросс-валидацию и метрики качества, чтобы выбрать наилучшую модель.

8. Наличие предобученных моделей: если у нас ограниченные ресурсы, то лучшем рассмотреть использование предобученных моделей для нашей задачи.

Исходя из этих факторов, можно выбрать наиболее подходящую модель машинного обучения для нашей конкретной задачи. Важно также помнить, что иногда эффективно использовать не одну модель, а ансамбль моделей для повышения качества предсказаний.

В нашем случае данные размечены, что означает классифицированы и соотнесены к определённому классу. Здесь уместно использовать обучение с учителем. Выбор наш падёт на модели на основе линейной регрессии, деревьев, метод K-ближайших соседей (KNN).

Данные модели машинного обучения имеют несколько преимуществ:

1. Интерпретируемость: легко понимать и интерпретировать, поскольку они представляют собой простую логическую структуру. Это делает их полезными для принятия решений и объяснения прогнозов.

2. Универсальность: могут быть использованы как для задач классификации, так и для задач регрессии. Они могут быть применены к данным с различными типами признаков, включая категориальные признаки.

3. Работа с нелинейными зависимостями: могут обрабатывать сложные нелинейные зависимости в данных без необходимости их линеаризации.

4. Устойчивость к выбросам: хорошо устойчивы к выбросам и шуму в данных.

Также модели на основе деревьев, линейной регрессии, метода K-ближайших соседей (KNN) могут быть применены в различных областях, включая:

1. Классификация: часто используются для классификации объектов на основе их признаков. Например, они могут быть применены в медицине для диагностики заболеваний, в финансах для прогнозирования рисков и мошенничества, или в маркетинге для выявления целевой аудитории.

2. Регрессия: могут использоваться для прогнозирования непрерывных значений. Например, они могут быть применены в финансовой аналитике для прогнозирования цен на акции или в промышленности для прогнозирования времени до отказа оборудования.

3. Кластеризация: могут использоваться для кластеризации данных и выявления групп похожих объектов.

4. Ранжирование: могут быть использованы для ранжирования объектов в зависимости от их значимости или релевантности.

Использование этих моделей даст целый спектр ответов, которые мы хотим узнать.

Исследуем несколько моделей на основе классификации: линейная регрессия, случайный лес, метод K-ближайших соседей (KNN), а далее сравним результаты и выберем из них наилучшую с меньшей ошибкой предсказания.

Для начала займемся предварительной обработкой данных.

Импортируем библиотеки.

```
[44]: from sklearn.preprocessing import LabelEncoder, MinMaxScaler
      from sklearn.ensemble import RandomForestRegressor
      from sklearn.linear_model import LinearRegression
      from sklearn.metrics import mean_absolute_error,r2_score
      from sklearn.ensemble import GradientBoostingRegressor
      from sklearn.svm import SVR
      from sklearn.neighbors import KNeighborsRegressor
```

Выполним разбиение данных на тренировочную и тестовую выборку (train_test_split), откладывая в сторону 20% данных для тестирования. В решениях мы используем random_state=42.

```
[46]: train, test =  train_test_split(data, test_size = 0.2, random_state = 42)
```

Посмотрим на размеры разбитых частей (тренировочной и тестовой).

```
[48]: print(train.shape, test.shape)
      (2350, 22) (588, 22)
```

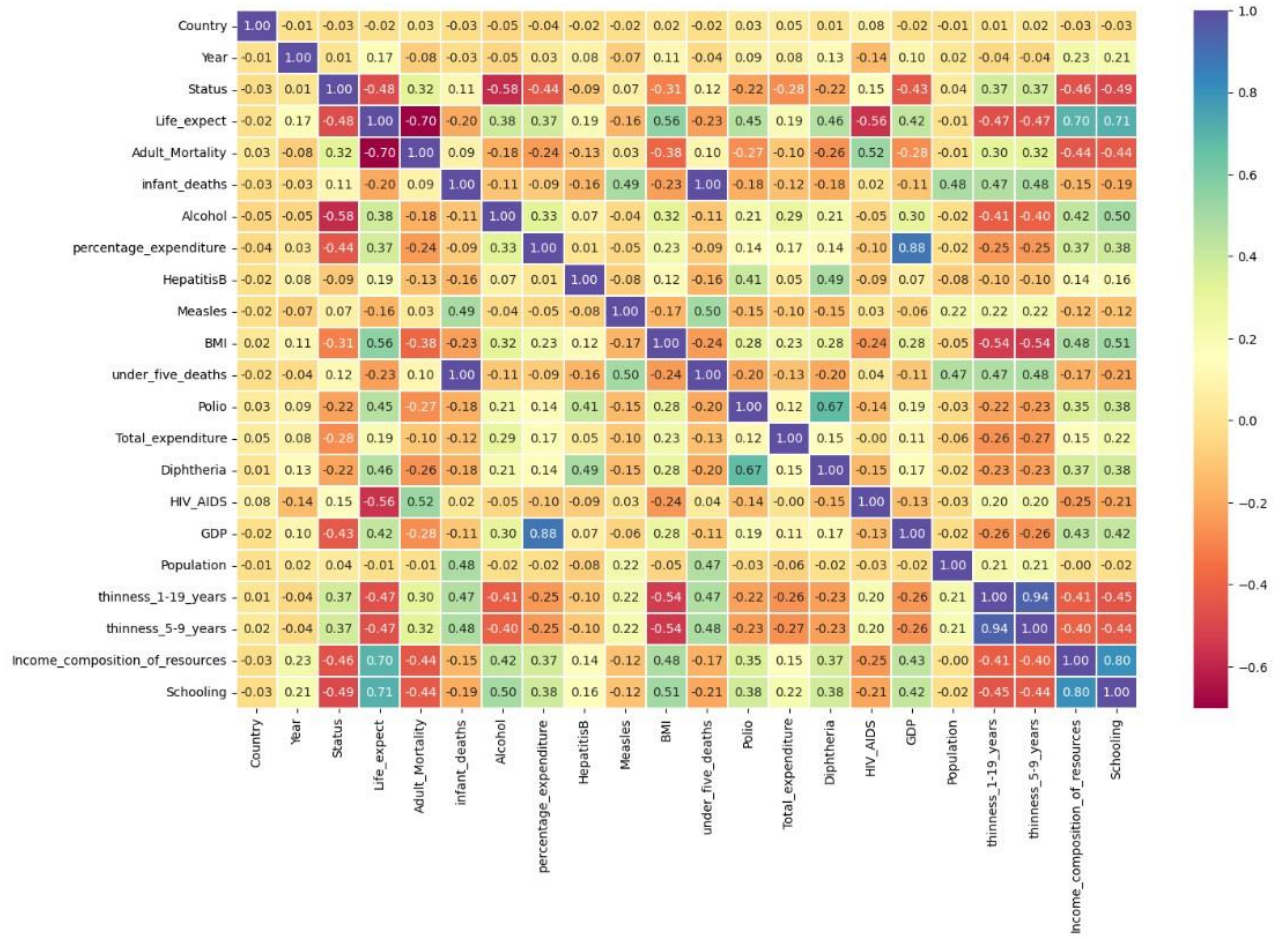
Используя labelEnconder заменим существующие текстовые данные новыми закодированными, переводим:

```
[51]: encoder = LabelEncoder()
      for column in ["Country", "Status"]:
          train[column] = encoder.fit_transform(train[column])
          test[column] = encoder.fit_transform(test[column])
```

Построим корреляционную матрицу для всех параметров.

```
[55]: plt.figure(figsize = (16,10))
      sns.heatmap(train.corr(), annot = True, fmt = ".2f", linewidths = 0.2,cmap='Spectral')
      plt.show()
```

В итоге получим следующий рисунок:



В очередной раз подтверждается тот факт, что ожидаемая продолжительность жизни - это не только статистический показатель, но и отображение состояния страны в целом. То, сколько в среднем живет человек зависит от большого количества факторов. Это и уровень достатка, и качество питания, образ жизни и наличие вредных привычек, уровень медицины и т.д.

Проверяем значения тренировочной выборки.

```
[57]: train.head()
```

[57]:

	Country	Year	Status	Life_expect	Adult_Mortality	infant_deaths	Alcohol	percentage_expenditure	HepatitisB	Measles	...	Polio	Total_expenditure	Diphtheria
456	27	2007	1	72.3	126.0	0	5.28	345.463714	96.000000	0	...	98.0	4.30	98.0
462	27	2001	1	73.0	152.0	0	3.81	150.743486	80.940461	0	...	91.0	5.19	9.0
2172	143	2011	1	74.6	143.0	0	10.43	0.000000	99.000000	0	...	99.0	7.58	99.0
2667	174	2013	1	74.9	13.0	3	1.29	594.645310	98.000000	16	...	98.0	7.26	98.0
381	23	2002	1	74.8	95.0	0	0.13	941.703687	99.000000	0	...	99.0	3.40	94.0

5 rows × 22 columns

```
[57]: train.head()
```

[57]:

patitisB	Measles	...	Polio	Total_expenditure	Diphtheria	HIV_AIDS	GDP	Population	thinness_1-19_years	thinness_5-9_years	Income_composition_of_resources	Schooling
.000000	0	...	98.0	4.30	98.0	0.6	3112.285712	4.864380e+05	8.1	8.0	0.602	11.9
.940461	0	...	91.0	5.19	9.0	0.8	1268.884564	4.437160e+05	9.4	9.3	0.562	11.0
.000000	0	...	99.0	7.58	99.0	0.1	7483.158469	1.275338e+07	4.3	4.3	0.733	12.9
.000000	16	...	98.0	7.26	98.0	0.1	4199.472530	1.114558e+06	6.4	6.3	0.720	14.7
.000000	0	...	99.0	3.40	94.0	0.1	16846.219800	1.275338e+07	6.7	6.1	0.820	13.3

Проверяем значения тренировочной выборки.

```
[61]: test.head()
```

[61]:

	Country	Year	Status	Life_expect	Adult_Mortality	infant_deaths	Alcohol	percentage_expenditure	HepatitisB	Measles	...	Polio	Total_expenditure	Diphtheria
2546	155	2006	1	73.7	123.0	8	0.97	122.652333	83.000000	517	...	83.0	3.78	8.0
650	37	2006	0	75.9	113.0	0	11.83	1555.651986	80.940461	1	...	96.0	6.95	96.0
1740	105	2007	1	74.2	125.0	0	4.98	678.518894	9.000000	0	...	92.0	6.74	92.0
177	11	2014	1	76.8	7.0	0	1.57	367.255674	98.000000	46	...	98.0	4.98	98.0
1377	82	2000	1	51.9	428.0	77	1.51	0.681686	80.940461	21002	...	8.0	4.68	82.0

5 rows × 22 columns

```
[61]: test.head()
```

[61]:

patitisB	Measles	...	Polio	Total_expenditure	Diphtheria	HIV_AIDS	GDP	Population	thinness_1-19_years	thinness_5-9_years	Income_composition_of_resources	Schooling
.000000	517	...	83.0	3.78	8.0	0.1	1762.246170	1.891498e+07	6.4	6.3	0.636	11.4
.940461	1	...	96.0	6.95	96.0	0.1	11363.418450	4.440000e+02	1.8	1.8	0.783	13.9
.000000	0	...	92.0	6.74	92.0	0.1	5957.145693	6.158750e+05	2.1	2.2	0.762	13.6
.000000	46	...	98.0	4.98	98.0	0.1	24983.379200	1.275338e+07	6.1	6.0	0.820	14.5
.940461	21002	...	8.0	4.68	82.0	18.1	43.979713	3.145483e+06	9.2	9.1	0.448	8.4

3.2 Создание модели машинного обучения

Разделение обучения на независимую и зависимую переменную.

```
[64]: Train = train[['Year', 'Status', 'Adult_Mortality', 'Alcohol', 'HepatitisB', 'Measles', 'BMI',
                     'under_five_deaths', 'Polio', 'Total_expenditure', 'HIV_AIDS','GDP',
                     'thinness_1-19_years', 'Schooling']]
      Target = train["Life_expect"]
```

Разделение тестирования на независимую и зависимую переменную.

```
[66]: x_test = test[['Year', 'Status', 'Adult_Mortality', 'Alcohol', 'HepatitisB', 'Measles', 'BMI',
                    'under_five_deaths', 'Polio', 'Total_expenditure', 'HIV_AIDS','GDP',
                    'thinness_1-19_years', 'Schooling']]
      y_test = test["Life_expect"]
```

Применяем масштабирование данных к независимым переменным (minMax).

MinMaxScaler не уменьшает влияние выбросов, но линейно масштабирует их до фиксированного диапазона, где наибольшая встречающаяся точка данных соответствует максимальному значению, а наименьшая - минимальному значению.

```
[68]: scaler = MinMaxScaler()

      Train = scaler.fit_transform(Train)
      x_test= scaler.transform(x_test)
```

3.3 Функция выбора модели машинного обучения

```
[70]: # Создадим пустой датафрейм для моделирования
      Modeling = pd.DataFrame(columns=['Model', 'Training Score', 'Test R2 Score'])

      # Функция выбора модели
      def select_model(model_name):
          global Modeling  # Access the global DataFrame

          # Инициализация модели
          model = model_name

          # Обучение модели на тренировочных данных
          model.fit(Train, Target)

          # Рассчитаем оценку модели на тренировочных данных
          train_score = model.score(Train, Target)
          print(f"Score of the {model_name} model on the training data is: {train_score}")

          # Прогноз на основе тестовых данных
          predictions = np.round(model.predict(x_test), decimals = 1)

          # Оценка R^2 для тестовых данных
          test_r2_score = r2_score(y_test, predictions)
          print(f"R2 score of the {model_name} model on the test data is: {test_r2_score}")

          model_scores = pd.DataFrame({'Model': [model_name], 'Training Score': [train_score], 'Test R2 Score': [test_r2_score]})

          # Объединение фрейм данных model_scores с фреймом данных моделирования
          Modeling = pd.concat([Modeling, model_scores], ignore_index = True)
```

Начнем с традиционной линейной регрессии.

3.3.1 Применение функции для модели Линейной регрессии

Линейная регрессия является основным методом для задач регрессии и используется для моделирования зависимости между входными и выходными переменными. Эта самый простой метод регрессии. Одним из его достоинств является лёгкость интерпретации результатов.

```
[71]: select_model(LinearRegression())

Score of the LinearRegression() model on the training data is: 0.7952977827768264
R2 score of the LinearRegression() model on the test data is: 0.8010472509274107
```

где R2 - коэффициент детерминации. Он используется для оценки производительности модели линейной регрессии. Это величина изменения выходного зависимого атрибута, которая предсказуема из входных независимых переменных. Используется для проверки того, насколько хорошо наблюдаемые результаты воспроизводятся моделью, в зависимости от отношения общего отклонения результатов, описанных моделью. Можно сказать, что 80% изменчивости зависимого выходного атрибута может быть объяснено моделью, в то время как остальные 20% изменчивости все еще не учтены. Значение R2 равное 1 указывает на то, что прогнозируемые значения совпадают с фактическими.

3.3.2 Применение функции для модели K-ближайших соседей

Метод K-ближайших соседей (KNN) — является одним из наиболее простых и эффективных алгоритмов в задачах классификации и регрессии. Он основан на идее, что точки данных, которые расположены близко друг к другу в пространстве признаков, скорее всего, относятся к одному и тому же классу.

```
[72]: # Параметр n_neighbors принимает количество параметров, которые мы используем для придания зависимой переменной нового значения. По умолчанию это 5
select_model(KNeighborsRegressor(n_neighbors = 5))

Score of the KNeighborsRegressor() model on the training data is: 0.9254778732635724
R2 score of the KNeighborsRegressor() model on the test data is: 0.8979645785791343
```

Значение R2, полученное этим методом, выше по сравнению с предыдущим методом Линейной регрессии, что указывает на лучшие результаты.

3.3.3 Применение функции для модели случайного леса (RandomForestRegressor)

Модель случайный лес (RandomForestRegressor) – это алгоритм машинного обучения, который использует комбинацию нескольких деревьев

принятия решений для решения задач классификации или регрессии. Он работает путем создания леса случайных деревьев во время обучения и совершения предсказаний на основе голосования или усреднения результатов всех деревьев. Случайный лес (RandomForestRegressor) обладает высокой точностью, устойчив к переобучению и способен обрабатывать как категориальные, так и количественные данные.

[76]:
```
# мы создаём объект RandomForestRegressor с:
# 'n_estimators':  = 100, то есть с 100 деревьями решений
# 'max_depth':  = 8 - максимальная глубина деревьев в лесу ограничена 8 уровнями. Это значит, что деревья не будут расти дальше 8 уровней вниз.
# 'min_samples_split': = 5 - минимальное количество выборок, необходимое для разбиения узла, равно 5. Это означает, что для создания нового узла в дереве

select_model(RandomForestRegressor(n_estimators = 100, max_depth=8, min_samples_split=5))
```

```
Score of the RandomForestRegressor(max_depth=8, min_samples_split=5) model on the training data is: 0.9748512514061026
R2 score of the RandomForestRegressor(max_depth=8, min_samples_split=5) model on the test data is: 0.9575145038519549
```

Результаты экспериментов показывают, что базовая модель случайного леса (RandomForestRegressor) демонстрирует более высокую точность предсказаний, чем модели Линейной регрессии и K-ближайших соседей.

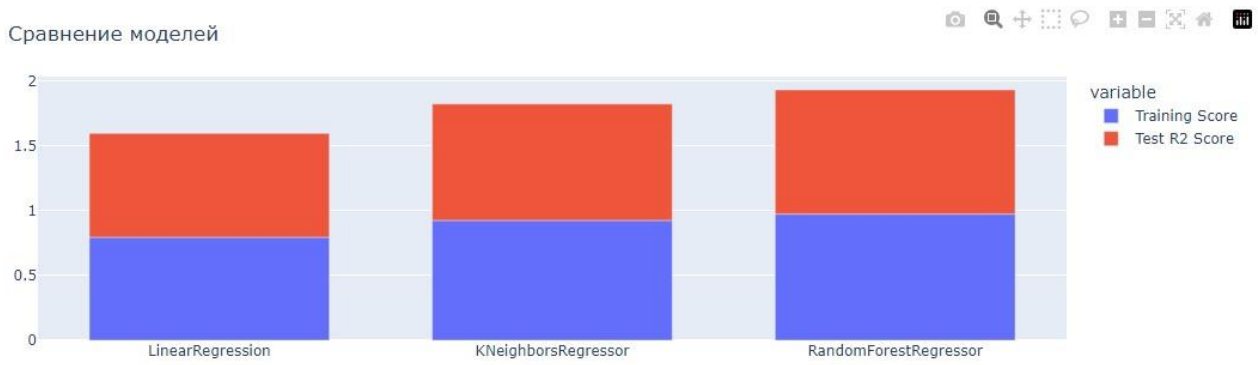
3.4 Оценка работы моделей машинного обучения

Для оценки работы моделей машинного обучения построим диаграмму

[97]:
```
# Переименовываем модели
names = ['LinearRegression', 'KNeighborsRegressor', 'RandomForestRegressor']

# loop over list to rename model
for i in range(3):
    Modeling.rename(index = {i : names[i]}, inplace = True)

# drop model column
Modeling.drop(columns= "Model", inplace = True)
```

[98]:
```
# Построим диаграмму оценки работы моделей
fig = px.bar(
    Modeling,
    x = Modeling.index,
    y = Modeling.columns,
)
fig.update_layout(
    bargap=0.3,
    title="Сравнение моделей",
)

fig.update_xaxes(title_text='')
fig.update_yaxes(title_text='')

fig.show()
```

В итоге получим следующий график:



Если сделаем вывод, то можно сказать, что модель случайный лес (RandomForestRegressor) работает лучше остальных моделей и демонстрирует более высокую точность предсказаний.

Полученное значение коэффициента детерминации R2 = 0.958311498492542 говорит о том, что регрессионная модель, использованная для предсказания ожидаемой продолжительности жизни, является очень хорошей и объясняет около 95,83% вариации в этой зависимой переменной. Это отличный результат, указывающий на высокую предсказательную способность модели.

3.5 Общий вывод

Ожидаемая продолжительность жизни предельно точно отражает современные реалии и ближайшие перспективы развития. Ее увеличение отражает прогресс в социальном и экономическом развитии, а также в области здравоохранения, в частности успех в борьбе со смертельными детскими заболеваниями, материнской смертностью и в последнее время со смертностью в пожилом возрасте. Она дает возможность иначе взглянуть на то, что такое пожилой возраст, а также на то, как может складываться вся наша жизнь.

Подтверждается закономерность: по мере роста экономического благосостояния стран увеличивается ожидаемая продолжительность жизни

населения. Значит, динамикой этого в известных пределах может «управлять» государство.

Взаимосвязь между расходами на здравоохранение и ожидаемой продолжительностью жизни достаточно сложная. Все указывает на то, что дальнейшее увеличение продолжительности жизни невозможно без серьезных научных прорывов. Для увеличения срока жизни необходимо разработать новые лекарства, направленные на замедление процессов старения, а не просто лечить уже существующие заболевания.

Расходы на здравоохранение оказываются значимым фактором, однако надо признать, что более существенную роль в достижении результатов играет общее экономическое развитие. Оказалось, что для бедных стран важнее всего доступность питания и расходы на здравоохранение, а для богатых — распространенность вредных привычек, что является негативным побочным эффектом роста доходов.

Пагубное воздействие на здоровье вредных привычек после достижения определенного уровня личного благосостояния, могут частично перевешивать улучшения, способствующие росту благосостояния. Низкая физическая активность, повышенное потребление алкоголя, табака, сахара и животных жиров - все это характерные черты современного развитого общества, главным образом потому, что высокий доход на душу населения сделал такое потребление возможным. Курение, нездоровое питание, недостаточная физическая активность и злоупотребление алкоголем являются основными факторами риска, оказывающими биологическое воздействие на здоровье и демографические структуры.

Несмотря на то, что в прошлом было проведено множество исследований факторов, влияющих на продолжительность жизни, с учётом демографических переменных, структуры доходов и уровня смертности. Можно утверждать, что в развивающихся странах, где рабочая сила, как правило, очень дешева, имеет смысл инвестировать в трудоемкие услуги, такие как санитария и здравоохранение, поскольку их чистая стоимость для общества на самом деле

очень низка, а выгоды могут быть существенными. Между тем, борьба с ожирением и другими вредными привычками – это наиболее насущные проблемы в развитых странах, может добавить больше здоровых лет к средней продолжительности жизни в сегодняшних богатых странах.

Заключение

В данном исследовании были рассмотрены факторы, влияющие на ожидаемую продолжительность жизни с применением алгоритмов машинного обучения. Целью исследования было выявить факторы, влияющие на снижение ожидаемой продолжительности жизни в современном обществе.

Для достижения этой цели был проведен анализ базы данных Global Health Observatory (GHO) за 2000–2015 годы по 193 странам, включающий информацию, связанную с иммунизацией, факторы смертности, экономические факторы и социальные факторы. Затем были применены различные алгоритмы машинного обучения, такие как модель случайного леса (RandomForestRegressor), линейная регрессия, метод K-ближайших соседей (KNN) для построения моделей прогнозирования оттока.

Результаты исследования показали, что на снижение ожидаемой продолжительности жизни влияют несколько ключевых факторов. Она может снизится из-за таких проблем, как голод, война, болезни и плохое здоровье.

Таким образом, применение алгоритмов машинного обучения для поможет определить прогнозирующий фактор, который способствует снижению значения ожидаемой продолжительности жизни. Это поможет подсказать стране, какой области следует уделить особое внимание, чтобы эффективно увеличить ожидаемую продолжительность жизни ее населения.

Список используемой литературы

1. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. – М.: ДМК Пресс, 2015
2. Саттон Р.С., Барто Э.Дж. Обучение с подкреплением: Введение. 2-е изд. – М.: ДМК Пресс, 2020
3. Шалев-Шварц Ш., Бен-Давид Ш. Идеи машинного обучения: от теории к алгоритмам. – М.: ДМК Пресс, 2019.
4. Agrawal R. et al. Fast discovery of association rules // Advances in Knowledge Discovery and Data Mining. AAAI Press, 1996. P. 307–328.
5. Han J. et al. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach // Data Mining and Knowledge Discovery. 2004. Vol. 8, № 1. P. 53–87.
6. Tipping M. The Relevance Vector Machine // Advances in Neural Information Processing Systems / ed. Solla S., Leen T., Müller K. MIT Press, 2000. Vol. 12.
7. Метод релевантных векторов [Электронный ресурс]. – Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=RVM
8. Quinlan J.R. C4.5: programs for machine learning. San Mateo, Calif: Morgan Kaufmann Publishers, 1993. 302 p.
9. Breiman L. et al. Classification and Regression Trees. Wadsworth, Belmont, California, 1984
10. Rabiner L.R. A tutorial on hidden Markov models and selected applications in speech recognition // Proc. IEEE. 1989. Vol. 77, № 2. P. 257–286.
11. Документация Python https://docs.python.org/3.10/
12. Документация Pandas https://pandas.pydata.org/docs/
13. Документация Python https://scikit-learn.ru/

Приложение 1. Содержание файла LifeExpectancyData.csv
https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who

Приложение 2. Содержание файла LifeExpectancyData.ipynb
https://github.com/Valeria28-10/Diplom/blob/main/LifeExpectancyData.csv