Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение

высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики управления и технологий

Ли Александр Андреевич БД-241м

## **Практическая работа 3.1. "Анализ и визуализация больших данных. Машинное обучение на больших данных с использованием Apache Spark MLlib".**

Направление подготовки/специальность

38.04.05 - Бизнес-информатика

Бизнес-аналитика и большие данные

(очная форма обучения)

Вариант 12

Москва

2025

Цель

Познакомиться с понятием «большие данные» и способами их обработки;

Познакомиться с инструментом Apache Spark и возможностями, которые он предоставляет для обработки больших данных.

Получить навыки выполнения разведочного анализа данных использованием pyspark.

Ход работы

Открываем ноутбук [advanced\_pyspark\_for\_exploratory\_data\_analysis\_student.ipynb](https://github.com/BosenkoTM/PySpark/blob/main/advanced_pyspark_for_exploratory_data_analysis_student.ipynb).

Импортируем библиотеки

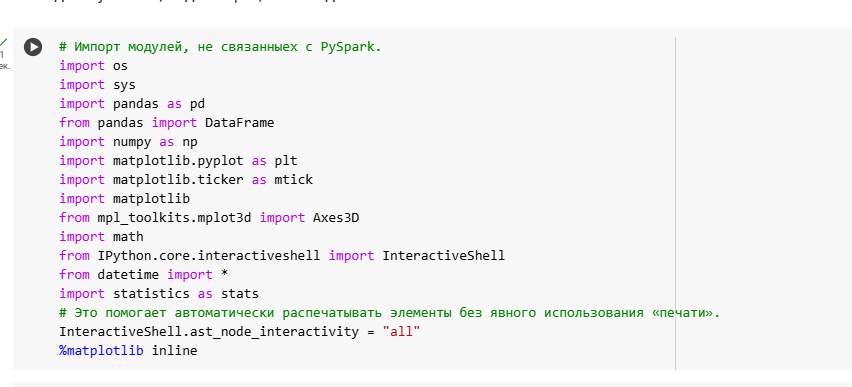


Рис.1 импорт библиотек

Подключаем google disk

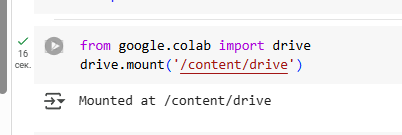


Рис.2 Подключаем google disk

Создаем директорию pyspark2024 и разархивируем в него данные



Рис.3 Разархивируем данные в директорию pyspark2024

Импортируем pyspark, создаем сеанс и cоздаем спарфрейм



Рис.4 Предварительная настройка pyspark

Смотрим какие данные получили из json файла.

Пол, сердцебиение, id, долгота и широта, скорость, вид спорта ,url, время и user\_id.



Рис.5 просмотр полученных данных

Описание фрейма данных

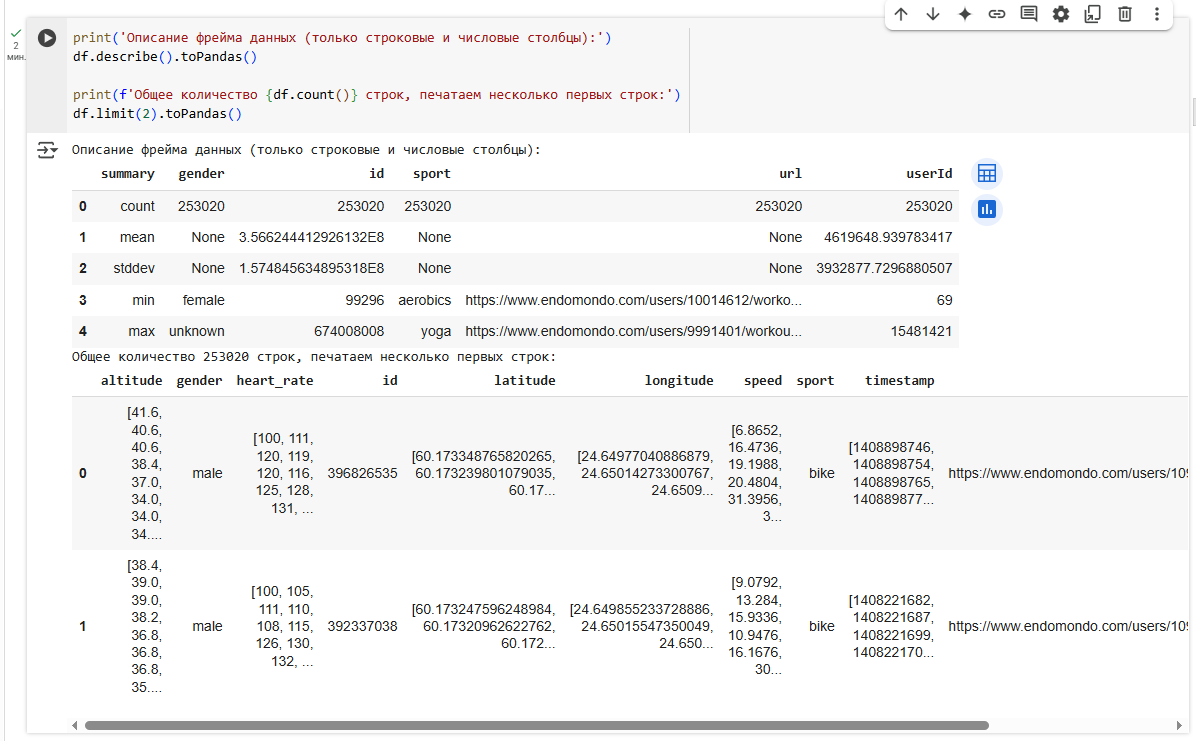


Рис.6 описание данных

Ищем пропущенные значения и аномалии.

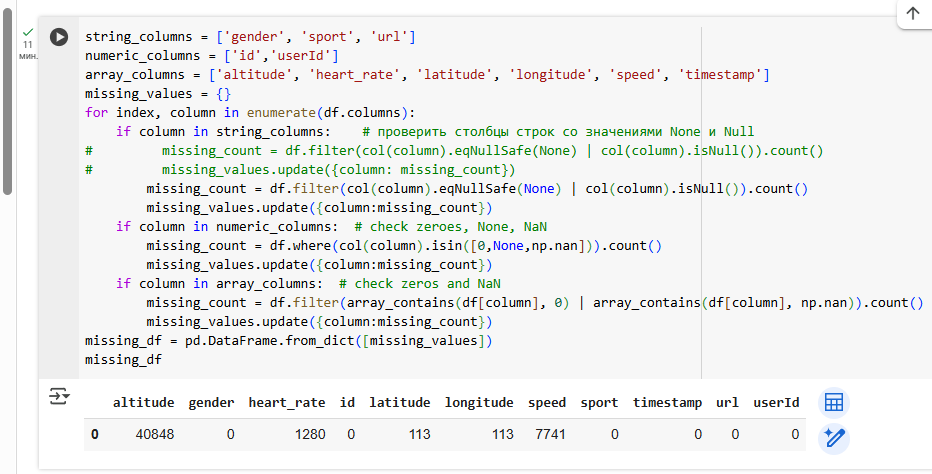


Рис.7 Поиск пропущенных значений

Создаем новый столбец для подсчета количества временных меток, записанных для каждой тренировки.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.8 подсчет кол-ва временных меток

Выводим количество пользователей с менее чем 50 тренировками и статистической сводкой

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рис.9 пользователей с менее чем 50 тренировок

Ленивая оценка pyspark.

Делаем анализ уникальных пользователей по каждому виду спорта в дата фрейме используя pyspark и выводим виды спорта по количеству их пользователей

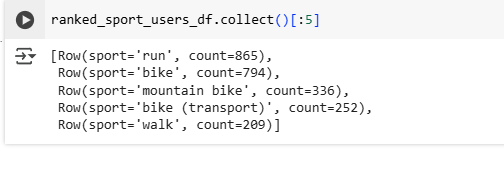


Рис.10 анализ пользователей по каждому спорту

Делаем исследовательский анализ данных.

Составим таблицу указав 5 лучших типов тренировок и сделаем визуализацию

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

Рис.11 визуализация 5 самых популярных видов спорта.

Смотрим данные в зависимости от пола

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис.12 просмотр данных в зависимости от пола

Unstack pyspark dataframe

Изменить форму таблицы в Pyspark, используя df.unstack(), чтобы получить правильное разделение , визуализируем процент участия по гендерному признаку.

Изображение выглядит как текст, снимок экрана, Параллельный, диаграмма

Автоматически созданное описание

Рис.13 Процент участия по гендерному признаку

Посмотрим на верхнее парето из 5 видов спорта, в которых больше всего участников.

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание

Рис.14 Виды спорта, в которых больше всего участников.

Анализируем **активность пользователей по количеству различных видов спорта,** в которых они участвовали.

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рис.15 анализ активности пользователей

Посмотрим на статистику по полу на коробчатой диаграмме:

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рис.16 коробчатая диаграмма

Для более детального наблюдения разобьем количество рекордов по каждому виду деятельности по каждому отдельному виду спорта.

Сделаем распределение тренировок по видам деятельности.

Изображение выглядит как диаграмма, План, линия, График

Автоматически созданное описание

Рис.17 распределение тренировок по видам деятельности

Далее, определим, сколько пользователей имеют более 10 тренировок, разобьем их по полу, и посчитаем, какой процент они составляют от общего числа всех пользователей.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис.18 анализ данных

Распарсим время, регистрируем функцию в реестре pyspark

Изображение выглядит как текст, Шрифт, число, снимок экрана

Автоматически созданное описание

Рис.19 преобразование столбца время

Выведем продолжительность каждой тренировки

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.20 продолжительность каждой тренировки

Изображение выглядит как диаграмма, текст, линия, План

Автоматически созданное описание

Рис.21 продолжительность каждой тренировки

Преобразуем объекты строк в устойчивый распределенный набор данных spark (rdd)

Изображение выглядит как текст, снимок экрана, число, Параллельный

Автоматически созданное описание

Рис.22 Преобразование в rdd

Визуализируем статистику по видам спорта, разделяя данные на 7 блоков по 7 видов спорта и построим графики.

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание

Рис.23 статистику по видам спорта

Изображение выглядит как текст, диаграмма, снимок экрана, График

Автоматически созданное описание

Рис.24 статистику по видам спорта

Изображение выглядит как текст, снимок экрана, График, диаграмма

Автоматически созданное описание

Рис.25 статистику по видам спорта

**Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание**

Рис.26 статистику по видам спорта

**Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание**

Рис.27 статистику по видам спорта

**Изображение выглядит как текст, диаграмма, снимок экрана, График

Автоматически созданное описание**

Рис.28 статистику по видам спорта

**Изображение выглядит как текст, диаграмма, снимок экрана, График

Автоматически созданное описание**

Рис.29 статистику по видам спорта

Построим гистограмму, чтобы посмотреть на распределение часов начала тренировок, сгруппированных по видам спорта и с разбивкой по полу.

**Изображение выглядит как диаграмма, План, карта, линия

Автоматически созданное описание**

Рис.30 распределение часов начала тренировок

Использование 2 переменных для определения критериев выборки:

максимальное количество пользователей по полу и максимальное количество тренировок по виду спорта

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

Рис.31 определения критериев выборки

Нормализуем время для всех тренировок, рассчитав продолжительность (в секундах) каждой записи временной метки из первой записи тренировки (первый элемент datetime списка в этой тренировке).  
Затем отображаем частоту сердечных сокращений в зависимости от этого нормализованного времени, группируя по видам спорта.

**Изображение выглядит как текст, снимок экрана, линия, диаграмма

Автоматически созданное описание**

Рис. 32 Рассчет продолжительность

У нас будет некоторая визуализация в трех столбцах с информацией о смещении/геометрии (longitude,latitude и altitude). Поскольку расположение каждого пользователя и тренировки отличается друг от друга, мы отображаем только несколько отдельных тренировок на 3D-графиках, чтобы просмотреть маршрут тренировки.

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

Визуализирует **трек каждой отдельной тренировки в виде 3D-маршрута** (по координатам: долгота, широта, высота). Он полезен для анализа маршрутов тренировок — особенно на открытом воздухе (например, бег, велосипед, лыжи).

**Изображение выглядит как текст, диаграмма, Шрифт, снимок экрана

Автоматически созданное описание**

Рис.33 3-D маршрут тренировки

**Индивидуальное задание вариант 12**

Вопрос 1   
Какие виды спорта, согласно анализу в разделах 5 и 6, являются самыми популярными и по числу пользователей, и по числу тренировок? Есть ли виды спорта, популярные в одной метрике, но не в другой?

1. **являются самыми популярными и по числу пользователей**
2. **others** (прочие) – 1000 пользователей
3. **run** – 870 пользователей
4. **bike** – 800 пользователей
5. **mountain bike** – 350 пользователей
6. **bike (transport)** – 250 пользователей
7. **являются самыми популярными и по числу тренировок**
8. **weight training**
9. **treadmill running**
10. **walk**
11. **treadmill walking**
12. **step counter**
13. **stair climbing**
14. **run**
15. **bike**
16. **mountain bike**
17. Есть ли виды спорта, популярные в одной метрике, но не в другой?

### **Популярные по числу пользователей, но не по числу тренировок**

1. **others**
2. **bike (transport)**
3. **walk**

### **Популярные по числу тренировок, но не по числу пользователей**

### **weight training**

1. **treadmill running** и **treadmill walking**
2. **step counter**, **stair climbing**

Вопрос 2

Интерпретируйте статистику по duration (раздел 7, df.select('duration').toPandas().describe().T). Какова медианная продолжительность тренировки? Что означает большая разница между 75-м перцентилем и максимумом?

1. Какова медианная продолжительность тренировки?

Медиана составляет 58.916668. Продолжительность половины всех тренировок 58.92 минуты, другая половина -дольше.

1. Что означает большая разница между 75-м перцентилем и максимумом?

**Большинство значений нормальные**, но есть **одиночные очень большие**.

В выборке есть один или несколько очень больших выбросов, которые не характерны для данных.

Вопрос 3

Объясните бизнес-ценность анализа интервалов записи данных (interval, раздел 8). Почему компании Endomondo может быть важно знать, как часто их трекеры записывают данные?

Интервалы записи данных показывают, как часто устройства собирают информацию о пользователе, что влияет на точность данных. Если интервал слишком большой, некоторые детали активности могут быть упущены, что приведет к неточным метрикам. Это может быть критично для некоторых видов спорта, таких как бег или велоспорт, где важно отслеживать изменения скорости и пульса в реальном времени. Если интервал записи данных слишком длинный, данные могут не отражать реальную активность.

Точные данные о частоте и интенсивности активности позволяют компании предоставлять персонализированные рекомендации по тренировкам. Низкочастотная запись данных может снизить эффективность этих рекомендаций, что влияет на удовлетворенность пользователей. Для профессиональных спортсменов или тех, кто требует более высокочастотных данных, компания может предложить платные планы с более частой записью данных, что способствует дополнительной монетизации.

Анализ интервалов записи данных помогает выявить, какие виды спорта требуют особого внимания при разработке новых функций и устройств. Например, экстремальные виды спорта, такие как парашютизм или скалолазание, могут потребовать более частой записи данных для повышения безопасности и точности.

Кроме того, если наблюдается большое разброс интервалов записи , это может указывать на проблемы с работой устройства или алгоритмами, что важно для устранения ошибок и повышения надежности продукта.

Для компании Endomondo важно понимать, как часто их трекеры записывают данные, потому что частота записи напрямую влияет на точность анализа активности. Для разных видов спорта требуется различная частота записи: для динамичных видов спорта необходима высокая частота, в то время как для более спокойных — меньшая. Анализ интервалов помогает понять, какие функции требуются пользователям разных категорий и как улучшить продукт.

Также, информация о частоте записи данных позволяет компании создавать премиум-планы, которые предоставляют более точные и подробные данные, что способствует выделению на фоне конкурентов и повышению прибыли от платных услуг.

**Задание 4**

**Напишите код PySpark**, чтобы вычислить среднюю продолжительность (duration) тренировок для пользователей мужского (male) и женского (female) пола отдельно.

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

Рис.34 средняя продолжительность тренировок для мужчин и женщин

**From pyspark.sql.functions import avg**

**Df\_male = df.filter(df.gender == ‘male’)**

**Df\_male\_avg = df\_male.select(avg(‘duration’))**

**Df\_male\_avg.show()**

**From pyspark.sql.functions import avg**

**Df\_male = df.filter(df.gender == ‘female’)**

**Df\_male\_avg = df\_male.select(avg(‘duration’))**

**Df\_male\_avg.show()**

**Вопросы**

1. **Фреймворк обработки больших данных Apache Spark, его назначение, функции и отличия от Hadoop [MapReduce](http://95.31.0.249/moodle/mod/resource/view.php?id=1348" \o "MapReduce).**

**Apache Spark** — это **фреймворк для быстрой обработки больших данных** в распределённых системах.

Назначение

Обрабатывать большие объёмы данных

Делать это быстро, используя операции в памяти (RAM), а не на диске

Поддерживать разные типы задач:

* Анализ данных
* Машинное обучение
* Стриминг (обработка данных в реальном времени)
* SQL-запросы
* Работа с графами

Основные функции

RDD (Resilient Distributed Dataset) — хранение и обработка данных

DataFrame и SQL — удобная работа с табличными данными

Spark MLlib — инструменты машинного обучения

Spark Streaming — работа с потоками данных

Отличия от Hadoop MapReduce:

Apache spark делает операции в ram из-за чего работает быстрее ,чем Hadoop mapreduce который записывает на диск.

Hadoop каждый шаг пишет на диск, spark многократные шаги не пересохраняет.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Понятие устойчивого распределенного набора данных (RDD). Понятие раздела RDD (partition). Способы создания RDD. Трансформации (transformations) и действия (actions). Кэширование (cache) данных в Spark.**

## Раздел RDD

Каждый RDD делится на **части (partitions)** — по одной на каждую "машину" или ядро.

## Способы создания RDD:

1. **Из списка/массива**:

sc.parallelize([1, 2, 3, 4])

1. **Из файла**:

sc.textFile("data.txt")

1. **Из базы данных или HDFS** — через коннекторы

## Transformations — преобразования

**Создают новый RDD, но не запускают вычисления.**

Примеры:

* map() — применить функцию к каждому элементу
* filter() — отфильтровать данные
* flatMap() — как map, но с разворачиванием списков
* distinct() — удалить дубликаты
* union() — объединить RDD
* groupByKey() / reduceByKey() — сгруппировать по ключам

## Actions — действия (запускают вычисления)

**Запускают настоящую работу — и возвращают результат.**

* collect() — собрать все данные
* count() — посчитать количество
* first() — первый элемент
* take(n) — взять первые n
* reduce() — объединить всё
* saveAsTextFile() — сохранить на диск

**Кэширование**

Когда используется один и тот же RDD много раз — Spark может сохранить его в памяти, чтобы не пересчитывать каждый раз.

Это ускоряет работу, особенно при сложных вычислениях или многократном использовании.

1. **Реализация концепции [MapReduce](http://95.31.0.249/moodle/mod/resource/view.php?id=1348" \o "MapReduce) в фреймворке Spark. Функции map, flatMap, mapValues, mapPartitions, reduce, reduceByKey.**

MapReduce — это концепция обработки данных в два этапа:

1. Map — применить функцию к каждому элементу
2. Reduce — объединить результаты по ключу

Map()- Применяет функцию к каждому элементу.

flatMap()- То же, что map, но **разворачивает** вложенные элементы.

mapValues()- Работает только с PairRDD (пары ключ-значение).  
Изменяет **только значение**, не трогая ключ.

mapPartitions()-Обрабатывает **сразу весь раздел** данных — а не по одному элементу.

### reduce()-Объединяет все элементы RDD **в один результат**.

### reduceByKey()-Объединяет значения по одинаковым ключам.

**Операция перемешивания данных (shuffle): причины возникновения, влияние на производительность. Класс Partitioner. Пути повышения производительности.**

Shuffle — это операция, при которой данные перемещаются между разделами и узлами кластера.

groupByKey() — Spark должен перенести данные с одинаковыми ключами в одно место.

Чтобы это сделать — он перемешивает данные между машинами

## Причины возникновения shuffle:

Shuffle **возникает**, когда операция требует:

**Объединения данных по ключу, сортировки, перераспределения данных**

## Влияние shuffle на производительность:

Shuffle - может сильно **замедлить** выполнение задач, так как является **тяжёлой операцией,** так как , данные **читаются и пишутся на диск,** передаются **по сети между узлами** и требуют **много памяти**

## Класс Partitioner

Partitioner — это класс, который управляет, **как именно данные распределяются по разделам.**

По умолчанию:

Для reduceByKey используется **HashPartitioner**

Можно задать свой Partitioner, чтобы **оптимизировать** работу

## Как уменьшить влияние shuffle (повышение производительности):

|  |
| --- |
| Использовать reducebykey, а не groupbykey |
| Настроить partitioner |
| Кэшировать данные |
| Использовать mappartitions |
| Избегать ненужных repartition и coalesce |
| Использовать broadcast для маленьких таблиц при join |

Заключение

Входы выполнения лабораторной работы , были получены знания о больших данных и способах их обработки. Познакомился с инструментом Apache Spark и возможностями, которые он предоставляет для обработки больших данных. Получил навыки выполнения разведочного анализа данных использованием pyspark.