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

Кузьмина Дарья Юрьевна БД-241м

Инструменты хранения и анализа больших данных

**Лабораторная работа 1.2**

**Обработка данных с использованием Apache Spark и Python (PySpark)**

**Вариант 11**

Направление подготовки/специальность

38.04.05 - Бизнес-информатика

Бизнес-аналитика и большие данные

(очная форма обучения)

Руководитель дисциплины:

Босенко Т.М., доцент департамента

информатики, управления и технологий,

доктор экономических наук

Москва

2025

Содержание

[Введение 2](#_Toc192845116)

[Основная часть 2](#_Toc192845117)

[Заключение 16](#_Toc192845118)

## Введение

**Цель**

освоение основ работы с Apache Spark и его интеграцией с Python через библиотеку PySpark. Студенты научатся обрабатывать большие объемы данных, используя распределенные вычисления, а также научатся применять базовые операции с RDD (Resilient Distributed Datasets) и DataFrame, работать с SQL-запросами в Spark SQL, а также визуализировать результаты обработки данных.

**Задачи:**

1. Установить Apache Spark и PySpark.  
   Настроить рабочую среду для использования PySpark в Python, установить необходимые зависимости и настроить Spark на локальном компьютере или через облачную платформу.
2. Загрузка данных и их предварительная обработка.  
   Скачать или подготовить исходные данные для анализа (например, текстовые файлы или CSV). Загружать данные в Spark через RDD или DataFrame, выполнить предварительную обработку: очистка данных, фильтрация, преобразования.
3. Применение операций с RDD и DataFrame.  
   Научиться работать с RDD и DataFrame, выполнять такие операции как map, filter, reduce, groupBy, join и другие стандартные операции для обработки данных в распределенной среде.
4. Применение SQL-запросов через Spark SQL.  
   Использование SQL-запросов в Spark для извлечения и агрегации данных, создание временных таблиц и выполнение сложных запросов для анализа данных.
5. Визуализация результатов анализа данных.  
   Визуализировать полученные результаты с помощью библиотеки Python для визуализации данных (например, matplotlib или seaborn). Построить графики для лучшего представления результатов.
6. Подготовка отчета.  
   Оформить отчет по выполненной практике, в котором будет описан процесс выполнения работы, анализ полученных результатов и выводы. Включить ссылки на репозиторий и прикрепить сам отчет в формате PDF или Markdown.

## Основная часть

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

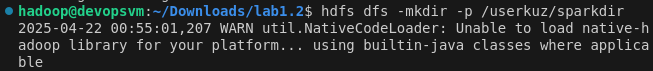
Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

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

Автоматически созданное описание



Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Понимаем, что на hadoop не установлены среда разработки для питона и идем с лицом лягухи в devops, чтобы оттуда уже выйти в hadoop, повторяя все вышеперечисленные команды.

Индивидуальное задание

|  |  |  |  |
| --- | --- | --- | --- |
| 11 | Анализ доставки: загрузить delivery.csv в HDFS, оценить эффективность логистики | SQL-анализ: рассчитать стоимость доставки по регионам и типам доставки | Построить карту загруженности по регионам |

Задание 1.

Создадим новую директорию под задание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Загрузим данные

Изображение выглядит как текст, чек, линия, Шрифт

Автоматически созданное описание

Откроем ноутбук, проверим подключение.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Посмотрим схему.

Результат выполнения задания.

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

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Потушили hadoop.

Проведен анализ данных о доставке заказов с использованием PySpark. Результаты включают:

Среднее время доставки для статуса "delivered" составляет 449.69 минут (предположительно). Это значение требует проверки на корректность данных, так как 7.5 часов на доставку может указывать на наличие выбросов или ошибок в данных (например, некорректные единицы измерения).

Процент доставленных заказов — 66.34%. Это приемлемый показатель, но для полноты анализа рекомендуется сравнить его с другими статусами (например, "canceled", "in transit").

Задание 2.

Изображение выглядит как текст, программное обеспечение, веб-страница, Веб-сайт

Автоматически созданное описание

Открыли файл, скачали данные

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Подключились к диску

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

В директории больше файлов нет

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Выполняем задание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Выполняем подбор признаков и преобразование категории.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Обучаем модель.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Проводим стратифицирование и разделение.

Изображение выглядит как текст, снимок экрана, Шрифт, веб-страница

Автоматически созданное описание

Модель оказалась непоказательной, возможно, я взяла неподходящие данные или ошиблась при подборе весов.

**Задание 3.**

Визуализация

В комментариях я все подписала!

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Выполняем основное заданиеИзображение выглядит как текст, снимок экрана

Автоматически созданное описание

Изображение выглядит как снимок экрана

Автоматически созданное описание

Карта получилась слишком маленькая, необходимо выполнить настройку фрейма.

Изображение выглядит как текст, карта, диаграмма, снимок экрана

Автоматически созданное описание

Готовый результат.

Изображение выглядит как текст, карта, диаграмма, снимок экрана

Автоматически созданное описание

Также было выполнено еще несколько полезных визуализаций в счет ошибки в обучении модели.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Изображение выглядит как текст, линия, снимок экрана, График

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт, Параллельный

Автоматически созданное описаниеИзображение выглядит как текст, линия, снимок экрана, График

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, диаграмма, Прямоугольник

Автоматически созданное описание

# Заключение

**Вывод**:

В ходе выполнения практической работы были успешно освоены базовые принципы работы с Apache Spark и PySpark, что позволило получить ключевые навыки обработки больших данных в распределенной среде. Установка и настройка Spark на локальной машине, а также интеграция с Python через PySpark, продемонстрировали гибкость экосистемы Spark и её совместимость с популярными инструментами анализа данных.

Работа с RDD и DataFrame, включая операции `map`, `filter`, `groupBy` и `join`, подтвердила эффективность Spark для параллельной обработки данных. Использование Spark SQL для выполнения SQL-запросов позволило объединить преимущества реляционных баз данных и распределенных вычислений, упростив агрегацию и анализ сложных наборов данных.

Визуализация результатов с помощью библиотек matplotlib/seaborn и оформление отчета показали, как этапы анализа данных — от предобработки до интерпретации — могут быть объединены в единый рабочий процесс. Важным аспектом стала работа с реальными данными, которая подчеркнула необходимость тщательной очистки и трансформации данных перед анализом.

Практика закрепила понимание преимуществ Spark: масштабируемость, скорость выполнения операций за счет распределенных вычислений и интеграция с экосистемой Python. Полученные навыки позволяют эффективно решать задачи анализа больших данных, что особенно актуально в условиях роста их объемов и сложности. Результаты работы подтверждают, что Spark является мощным инструментом для современных Data Science проектов, а его освоение открывает возможности для работы в области машинного обучения, ETL-процессов и реального времени аналитики.

Материалы практики (код, отчет) доступны в репозитории, что обеспечивает прозрачность и воспроизводимость результатов.