**Лабораторная работа №5 опциональная**

**Олейников Михаил Николаевич (olejnikov)**

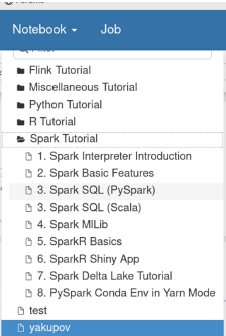
данная работа опциональная в рамках курса, но ее решение меня очень обрадует!

В текущей работе вам потребуется использовать продукт RT.WareHouse, включая Apache Zeppelin + pySpark который развернут в рамках нашего курса. Если появятся вопросы, пожалуйста напишите в общий учебный чат или личным сообщением преподавателю

**Цель задания**: исследовать приложенные датасеты для поиска **2 инсайтов** по данных (Data Insights). Инсайт - это представление данных в виде визуализации с применением предварительной группировки данных, которое представляет интересные моменты в данных. (например количество товара которое было продано в

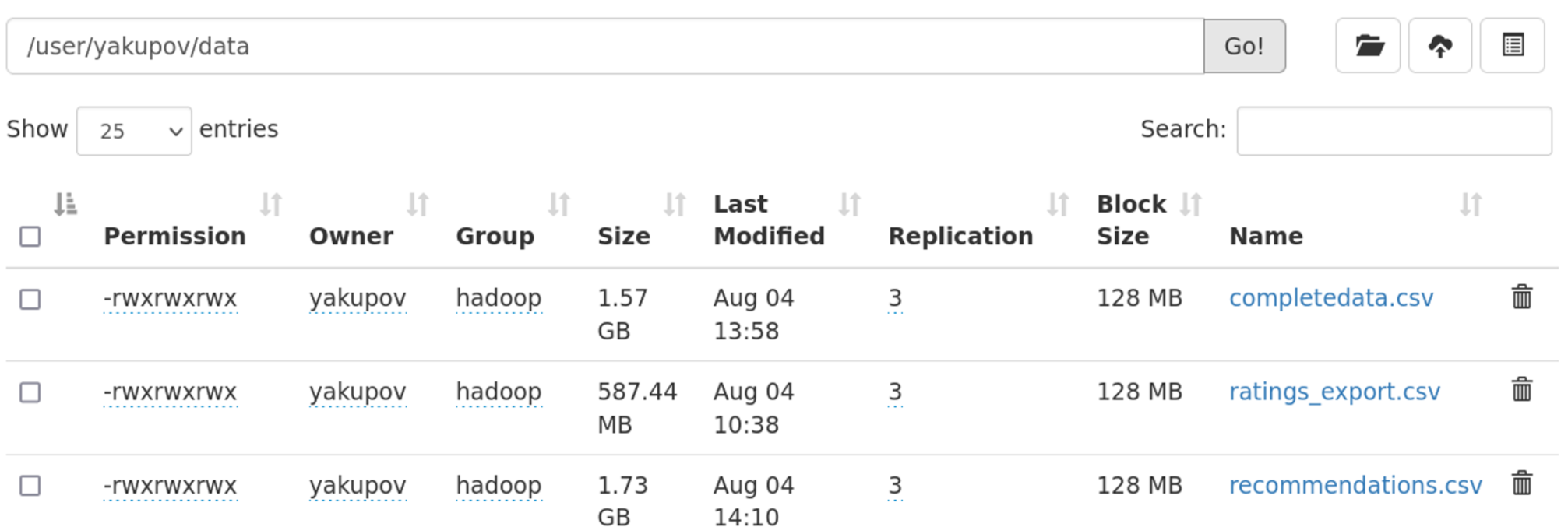
прошлом сезоне).

1. Соединитесь с Apache Zeppelin ( http://vm-dlake2-m-2:8180 ) из сессии JumpHost
2. Вы можете посмотреть на документацию использования pySpark непосредственно в Apache Zeppelin (выбрав пункт меню в Notebook)



1. Создайте свой Notebook, указав ему имя **фамилияинициалы** (например **yakupovas**). При создании выберите интерпретатор **Spark**
2. Вы можете выбрать один из трех представленных файлов для анализа, соединившись с ними из вашего Notebook к HDFS кластеру

Доступные файлы (все файлы в HDFS : /user/yakupov/data):



- **rating\_export.csv** - датасет оценок (rating\_val) по фильмам (movie\_id) от пользователей (user\_id)

- **completedata.csv** - набор данных о внутренних рейсах США с добавленными данными об аэропортах, самолетах и текущей погоде

- FL\_DATE - дата вылета

- DEP\_HOUR - час вылета

- MKT\_UNIQUE\_CARRIER - Уникальный код перевозчика

- MKT\_CARRIER\_FL\_NUM - Номер рейса перевозчика

- OP\_UNIQUE\_CARRIER - Уникальный код действующего перевозчика

- OP\_CARRIER\_FL\_NUM - Номер рейса действующего перевозчика

- TAIL\_NUM - номер / регистрация

- ORIGIN - код аэропорта вылета

- DEST - код аэропорта назначения

- DEP\_TIME - полное время вылета

- **recommendations.csv** - набор данных рекомендаций игр на Steam от пользователей

- app\_id - номер игры в Steam

- helpful - как много пользователей оценили рекомендацию полезной для себя

- funny - как много пользователей оценили рекомендацию смешной

- date - дата публикации рекомендации

- is\_recommended - комментатор рекомендует игру?

- hours - как много часов играл комментатор в игру

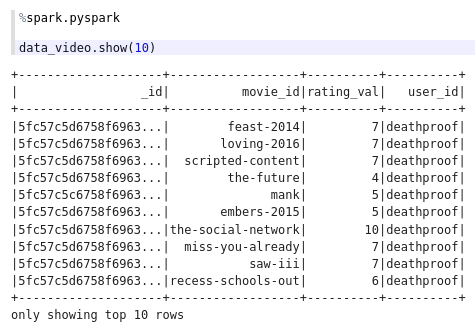
- user\_id - анонимизированный пользователь

- review\_id - внутренний номер рекомендации

- вы можете взять свой собственный датасет (размером не более 1.5 Gb)

(Предоставить преподавателю) приложите пожалуйста скрины по найденным вами 2 инсайтам (это может быть плоская таблица или диаграмма/график) в doc файл и прикрепите к лабораторной работе.

Открываем датафрейм, структуируем и т.д.



Выведем 5 самых популярных фильмов. Считаем что это фильмы, которым максимальное количество пользователей поставили самый высокий рейтинг.

spark.sql("\

select movie\_id, count(user\_id) \

from dv\_tab \

where rating\_val = (\

select max(rating\_val) \

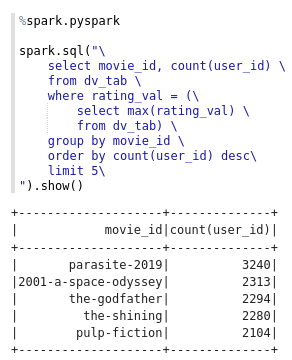
from dv\_tab) \

group by movie\_id \

order by count(user\_id) desc\

limit 5\

").show()



То же самое, но не напрямую sql, а через PySpark

import pyspark.sql.functions as F

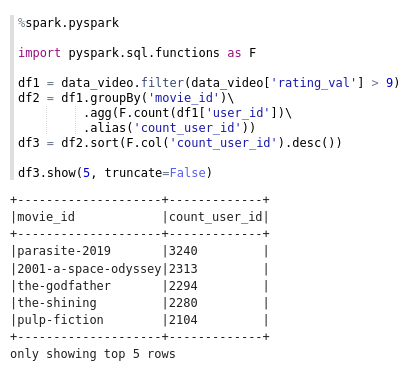
df1 = data\_video.filter(data\_video['rating\_val'] > 9)

df2 = df1.groupBy('movie\_id')\

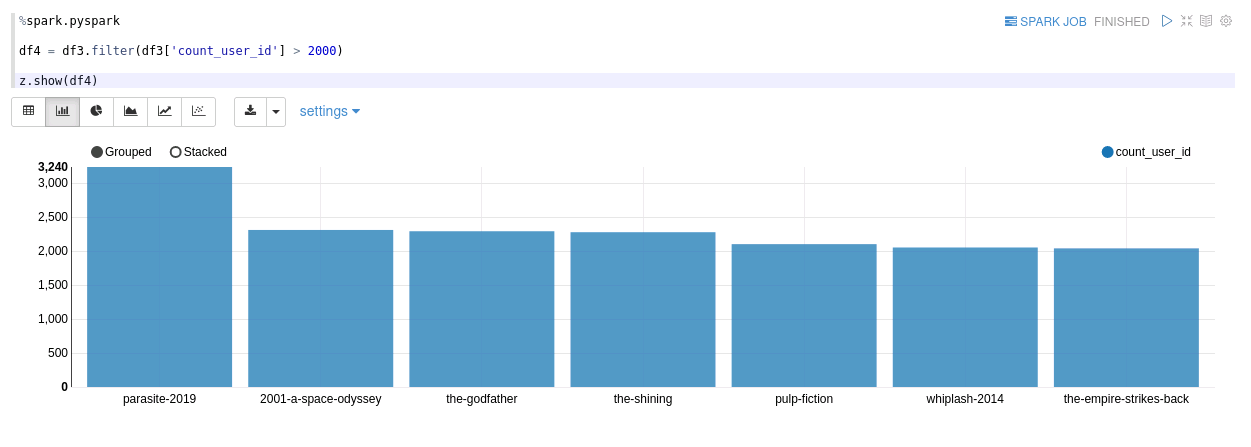
.agg(F.count(df1['user\_id']).alias('count\_user\_id'))

df3 = df2.sort(F.col('count\_user\_id').desc())

df3.show(5, truncate=False)



Я бы конечно побаловался визуализацией, но библиотек нет, поставить их невозможно, т.к. pip тоже не установлен. А с тепловой картой seaborn интересно было бы поиграться.



В качестве второго инсайта найдем наиболее активных пользователей.

