Подготовка к практическим заданиям ✈️

Apache Spark является самым популярным фреймворком для обработки больших данных. Давайте перейдем к практике и попробуем Spark боем. Для работы возьмем реальные данные о выполненных авиа-рейсах (2015 Flight Delays and Cancellations). Авиасообщение является одним из самых сложных механизмов и требующим слаженной организации. Оно активно используется как для перевозки пассажиров так и грузов, а все возникающие инциденты должны решаться мгновенно - так сказать "на лету". Среднее количество самолетов, которые находятся в небе в определенный момент, обычно колеблется в пределах 11-12 тысяч. Это огромное количество данных.

Итак, мы с вами для упрощения будем работать с 50% от основной выборки и подготовили для вас следующие датасеты которые будут использоваться в практике:

1. airlines.parquet - файл с данными о авиалиниях
2. airports.parquet - файл с данными о аэропортах
3. flights.parquet - файл с данными о авиарейсах

Структура данных

Авиалинии

Колонка	Тип	Описание
IATA_CODE	String	Идентификатор авиалинии
AIRLINE	String	Название авиалинии

Аэропорты

Колонка	Тип	Описание
IATA_CODE	String	Идентификатор аэропорта
AIRPORT	String	Название аэропорта
CITY	String	Город
STATE	String	Штат/Округ
COUNTRY	String	Страна
LATITUDE	Float	Широта расположения
LONGITUDE	Float	Долгота расположения

Авиарейсы

Колонка	Тип	Описание
YEAR	Integer	Год полета
MONTH	Integer	Месяц полета
DAY	Integer	День полета
DAY_OF_WEEK	Integer	День недели полета [1-7] = [пн-вс]
AIRLINE	String	Код авиалиний
FLIGHT_NUMBER	String	Идентификатор рейса (просто ид)
TAIL_NUMBER	String	Номер рейса
ORIGIN_AIRPORT	String	Код аэропорта отправления
DESTINATION_AIRPORT	String	Код аэропорта назначения
SCHEDULED_DEPARTURE	Integer	Время запланированного отправления
DEPARTURE_TIME	Integer	Время фактического отправления WHEEL_OFF - TAXI_OUT
DEPARTURE_DELAY	Integer	Общая задержка отправления
TAXI_OUT	Integer	Время, прошедшее между отправлением от выхода на посадку в аэропорту отправления и вылетом
WHEELS_OFF	Integer	Момент времени, когда колеса самолета отрываются от земли
SCHEDULED_TIME	Integer	Запланированное количество времени, необходимое для полета
ELAPSED_TIME	Integer	AIR_TIME+TAXI_IN+TAXI_OUT
AIR_TIME	Integer	Время в воздухе. Промежуток времени между WHEELS_OFF и WHEELS_ON
DISTANCE	Integer	Расстояние между двумя аэропортами
WHEELS_ON	Integer	Момент времени, когда колеса самолета касаются земли

TAXI_IN	Integer	Время, прошедшее между посадкой на колеса и прибытием на посадку в аэропорту назначения
SCHEDULED_ARRIVAL	Integer	Планируемое время прибытия
ARRIVAL_TIME	Integer	Время когда самолет фактически прибыл в аэропорт (прибыл к гейту) WHEELS_ON+TAXI_IN
ARRIVAL_DELAY	Integer	Время задержки в прибытии ARRIVAL_TIME-SCHEDULED_ARRIVAL
DIVERTED	Integer	Флаг указывающий что рейс приземлился в аэропорту не по расписанию (0/1)
CANCELLED	Integer	Флаг указывающий что рейс был отменен (0/1)
CANCELLATION_REASON	String	Причина отмены рейса: A - Airline/Carrier; B - Weather; C - National Air System; D - Security
AIR_SYSTEM_DELAY	Integer	Время задержки из-за воздушной системы
SECURITY_DELAY	Integer	Время задержки из-за службы безопасности
AIRLINE_DELAY	Integer	Время задержки по вине авиакомпании
LATE_AIRCRAFT_DELAY	Integer	Время задержки из-за проблем самолета
WEATHER_DELAY	Integer	Время задержки из-за погодных условий

Полетели! Разберемся в процессе! 🛫🛫🛫

Задача №1

Постройте сводную таблицу отображающую топ 10 рейсов по коду рейса (TAIL_NUMBER) и числу вылетов за все время. Отсеките значения без указания кода рейса.

Пример вывода:

TAIL_NUMBER	count
N480HA	1763
N484HA	1654
N481HA	1434

Сохраните сводную таблицу в формате parquet.

В качестве решения задачи необходимо отправить файл с вашим кодом на языке Python.

Используйте за основу следующие шаблоны:

1) PySparkJob1.py - шаблон для задачи процесса преобразования данных.

Параметры запуска задачи:
flights_path - путь к файлу с данными
result_path - путь куда будет сохранен результат

Подсказки: groupBy, orderBy

Строка запуска:

python PySparkJob1.py --flights_path=flights.parquet --result_path=result # ИЛИ spark-submit PySparkJob1.py --flights_path=flights.parquet --result_path=result

Задача №2

Найдите топ 10 авиамаршрутов (ORIGIN_AIRPORT, DESTINATION_AIRPORT) по наибольшему числу рейсов, а так же посчитайте среднее время в полете (AIR_TIME).

Требуемые поля:

Колонка	Описание
ORIGIN_AIRPORT	Аэропорт вылета
DESTINATION_AIRPORT	Аэропорт прибытия
tail_count	Число рейсов по маршруту (TAIL_NUMBER)
avg_air_time	среднее время в небе по маршруту

Пример вывода:

	ORIGIN_AIRPORT	DESTINATION_AIRPORT	tail_count	avg_air_time
0	SFO	LAX	6771	56.071861
1	LAX	SFO	6759	54.985734

Используйте за основу следующие шаблоны:

1) PySparkJob2.py - шаблон для задачи процесса преобразования данных.

Параметры запуска задачи:
flights_path - путь к файлу с данными
result_path - путь куда будет сохранен результат

Подсказки: groupby, orderby

Строка запуска:

python PySparkJob2.py --flights_path=flights.parquet --result_path=result # ИЛИ spark-submit PySparkJob2.py --flights_path=flights.parquet --result_path=result

Задача №3

Аналитик попросил определить список аэропортов у которых самые больше проблемы с задержкой на вылет рейса. Для этого необходимо вычислить среднее, минимальное, максимальное время задержки и выбрать аэропорты только те где максимальная задержка (DEPARTURE_DELAY) 1000 секунд и больше. Дополнительно посчитать корреляцию между временем задержки и днем недели (DAY_OF_WEEK)

Требуемые поля:

Поле	Описание
ORIGIN_AIRPORT	Код аэропорта отправления
avg_delay	Среднее время задержки для аэропорта
min_delay	Минимальное время задержки для аэропорта
max_delay	Максимальное время задержки для аэропорта
corr_delay2day_of_week	Корреляция между временем задержки и днем недели

Пример вывода:

ORIGIN_AIRPORT	avg_delay	min_delay	max_delay	corr_delay2day_of_week
MSY	8.631696	-31	1255	-0.011314
GEG	4.144039	-23	1075	-0.000779

Используйте за основу следующие шаблоны:

1) PySparkJob3.py - шаблон для задачи процесса преобразования данных.

Параметры запуска задачи:

flights_path - путь к файлу с данными

result_path - путь куда будет сохранен результат

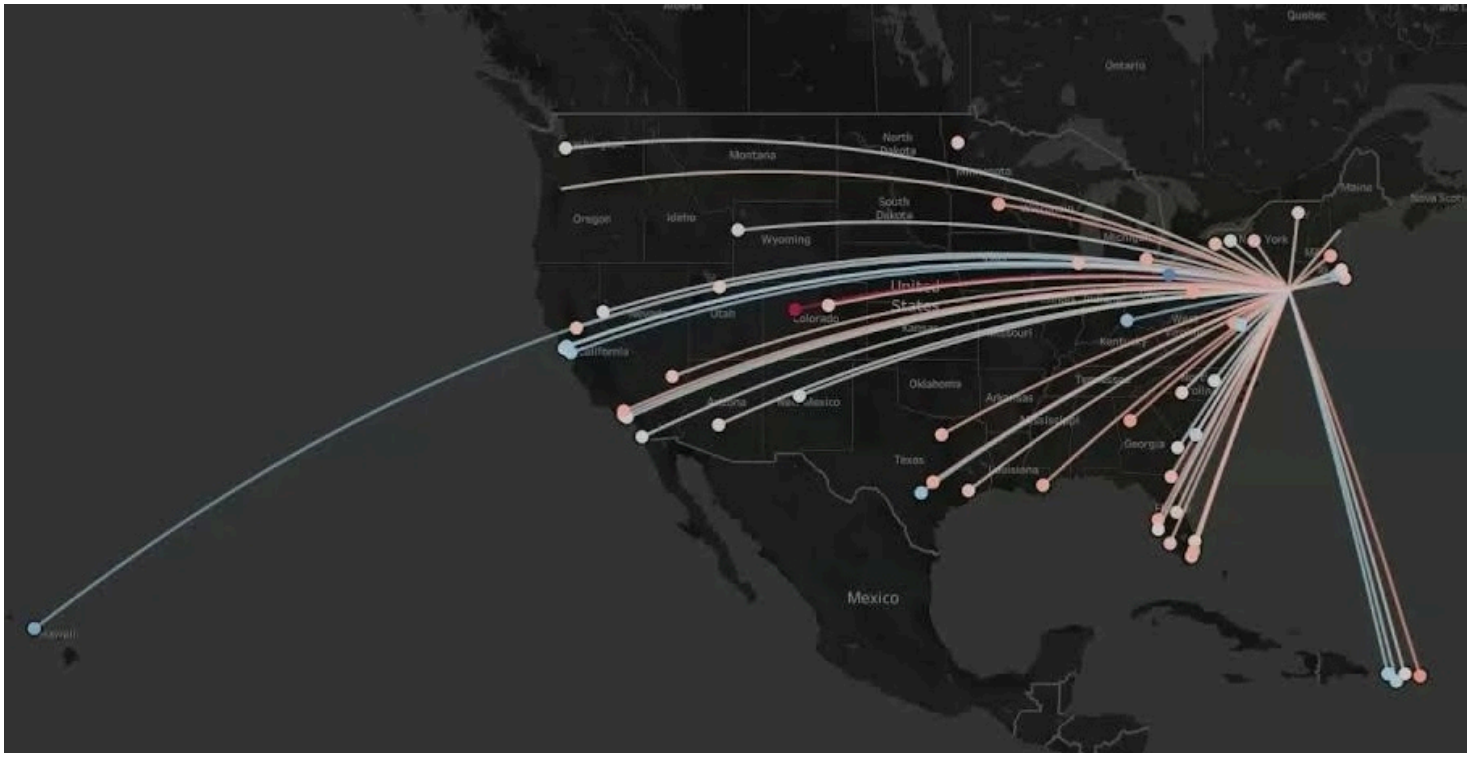
Подсказки: groupby, filter, corr

Строка запуска:

python PySparkJob3.py --flights_path=flights.parquet --result_path=result # ИЛИ spark-submit PySparkJob3.py --flights_path=flights.parquet --result_path=result

Задача №4

Для дашборда с отображением выполненных рейсов требуется собрать таблицу на основе наших данных.
Никакой дополнительной фильтрации данных не требуется.



Требуемые поля:

Поле	Описание
AIRLINE_NAME	Название авиалинии (airlines.AIRLINE)
TAIL_NUMBER	Номер рейса (flights.TAIL_NUMBER)
ORIGIN_COUNTRY	Страна отправления (airports.COUNTRY)
ORIGIN_AIRPORT_NAME	Полное название аэропорта отправления (airports.AIRPORT)
ORIGIN_LATITUDE	Широта аэропорта отправления (airports.LATITUDE)
ORIGIN_LONGITUDE	Долгота аэропорта отправления (airports.LONGITUDE)
DESTINATION_COUNTRY	Страна прибытия (airports.COUNTRY)
DESTINATION_AIRPORT_NAME	Полное название аэропорта прибытия (airports.AIRPORT)
DESTINATION_LATITUDE	Широта аэропорта прибытия (airports.LATITUDE)
DESTINATION_LONGITUDE	Долгота аэропорта прибытия (airports.LONGITUDE)

Пример вывода:

AIRLINE_NAME	TAIL_NUMBER	ORIGIN_COUNTRY	ORIGIN_AIRPORT_NAME	ORIGIN_LATITUDE	ORIGIN_LONGITUDE	DESTINATION_COUNTRY	DESTINATION_AIRPORT_NAME	DESTINATION_LATITUDE	DESTINATION_LONGITUDE
American Airlines Inc.	N787AA	USA	John F. Kennedy International Airport (New Yor...	40.63975	-73.77893	USA	Los Angeles International Airport	33.94254	-118.40807
American Airlines Inc.	N795AA	USA	Los Angeles International Airport	33.94254	-118.40807	USA	John F. Kennedy International Airport (New Yor...	40.63975	-73.77893
American Airlines Inc.	N798AA	USA	John F. Kennedy International Airport (New Yor...	40.63975	-73.77893	USA	Los Angeles International Airport	33.94254	-118.40807
American Airlines Inc.	N799AA	USA	Los Angeles International Airport	33.94254	-118.40807	USA	John F. Kennedy International Airport (New Yor...	40.63975	-73.77893
American Airlines Inc.	N376AA	USA	Dallas/Fort Worth International Airport	32.89595	-97.03720	USA	Honolulu International Airport	21.31869	-157.92241

Используйте за основу следующие шаблоны:

1) PySparkJob4.py - шаблон для задачи процесса преобразования данных.

Параметры запуска задачи:

flights_path - путь к файлу с данными о авиарейсах

airlines_path - путь к файлу с данными о авиалиниях

airports_path - путь к файлу с данными о аэропортах

result_path - путь куда будет сохранен результат

Подсказки: join, withColumnRenamed, withColumn

Строка запуска:

python PySparkJob4.py --flights_path=flights.parquet --airlines_path=airlines.parquet --airports_path=airports.parquet --result_path=result # ИЛИ spark-submit PySparkJob4.py --flights_path=flights.parquet --airlines_path=airlines.parquet --airports_path=airports.parque

Задача №5

Отдел аналитики интересует статистика по компаниям о возникших проблемах. Пришла задача построить сводную таблицу о всех авиакомпаниях содержащую следующие данные:

Колонка	Описание
AIRLINE_NAME	название авиалинии [airlines.AIRLINE]
correct_count	число выполненных рейсов без задержек
diverted_count	число рейсов выполненных с задержкой
cancelled_count	число отмененных рейсов
avg_distance	средняя дистанция рейсов
avg_air_time	среднее время в небе
airline_issue_count	число отмен из-за проблем с самолетом [CANCELLATION_REASON]
weather_issue_count	число отмен из-за погодных условий [CANCELLATION_REASON]
nas_issue_count	число отмен из-за проблем NAS [CANCELLATION_REASON]
security_issue_count	число отмен из-за службы безопасности [CANCELLATION_REASON]

Пример вывода:

	AIRLINE_NAME	correct_count	diverted_count	cancelled_count	avg_distance	avg_air_time	airline_issue_count	weather_issue_count	nas_issue_count	security_issue_count
0	Alaska Airlines Inc.	85805	217	343	1201.221108	158.330843	175	157	11	0
1	American Airlines Inc.	355623	1047	5427	1042.343900	139.916971	1432	3637	357	1

Используйте за основу следующие шаблоны:

1) PySparkJob5.py - шаблон для задачи процесса преобразования данных.

Параметры запуска задачи:

flights_path - путь к файлу с данными о рейсах

airlines_path - путь к файлу с данными об авиалиниях

result_path - путь куда будет сохранен результат

Подсказки: when/otherwise

Строка запуска:

python PySparkJob5.py --flights_path=flights.parquet --airlines_path=airlines.parquet --result_path=result # ИЛИ spark-submit PySparkJob5.py --flights_path=flights.parquet --airlines_path=airlines.parquet --result_path=result