BIGDATA TEAM

HW #05: Spark RDD

автор задания: BigData Team, коллективная работа.

1. Описание задания

В данном ДЗ нужно решить **2 задачи**. Задачи общие для всех. Решение надо выполнить с помощью Apache Spark, можно использовать только RDD API.

2. Критерии оценивания

Веса задач:

1. 40%
2. 60%

Балл за задачу складывается из:

- **60%** - правильное решение задачи
- **20%** - поддерживаемость и читаемость кода
  - в общем случае см. Clean Code и Google Python Style Guide
- **20%** - эффективность решения (такие как потребляемые CPU-ресурсы, скорость выполнения (в предположении свободного кластера)).

Discounts (скидки и другие акции):

- **100%** за плагиат в решениях (всем участникам процесса)
- **100%** за посылку решения после hard deadline
- **30%** за посылку решения в после soft deadline и до hard deadline
- **5%** за каждую новую посылку (одна дополнительная посылка бесплатно)

3. Описание данных

3.1 Дамп Википедии

`en_articles_part`:

- Путь на кластере: полный[1] датасет - **`/data/wiki/en_articles_part`**
- Формат: текст
- В каждой строке находятся следующие поля, разделенные знаком табуляции:
  1. `INT` - id статьи,
  2. `STRING` - текст статьи,

Пример:

[1] Да, здесь нет ошибки, работаем на части данных, чтобы побыстрее познакомиться со Spark RDD

```
12    Anarchism         Anarchism is often defined as a political
philosophy which holds the state to be undesirable, unnecessary, or
harmful.
```

3.2 Стоп-слова

`stop_words_en`:

- Путь на кластере: **`/data/stop_words/stop_words_en-xpo6.txt`**
- Формат: одно стоп-слово на строчку

Пример:

```
...
wherein
whereupon
wherever
...
```

4.1 Задача #1 (Task ID: spark.bigram): народные биграммы

Найдите все пары двух последовательных слов (биграмм), где первое слово:

```
narodnaya
```

Для каждой пары подсчитайте количество вхождений в тексте статей Википедии. Выведите все пары с их частотой вхождений в лексикографическом порядке. Формат вывода:

```
word_pair <tab> count
```

Условия:

- для однозначности вычислений, выделите слова из статьи с помощью регулярного выражения **`re.findall(r"\w+", text)`**;
- привести все слова к нижнему регистру;
- слова в паре объединить символом нижнего подчеркивания “_”;
- отсортировать слова в выводе по алфавиту;
- решение должно отрабатывать в течение 3х минут на свободном кластере (3 ноды x 8 CPU x 16GB RAM).

Пример вывода:

```
...
crazy_zoo 42
red_apple 100500
```

...

4.2 Задача #2 (Task ID: spark.collocation): коллокации

Коллокация - это комбинации слов, которые часто встречаются вместе. Например, «high school» или «Roman Empire». Чтобы определить, является ли пара слов коллокацией, можно воспользоваться метрикой NPMI - нормализованная точечная взаимная информация.

Чтобы рассчитать NPMI, введем несколько определений:

1. `P(a)` - вероятность увидеть слово “a” в датасете.
   `P(a) = num_of_occurrences_of_word_"a" / total_number_of_words`
   `total_number_of_words` - общее количество слов в тексте
2. `P(ab)` - вероятность увидеть пару слов “a” и “b”, идущих подряд.
   `P(ab) = num_of_occurrences_of_pair_"ab" / total_number_of_word_pairs`
   `total_number_of_word_pairs` - общее количество пар
3. `PMI(a,b) = ln( P(ab) / [P(a) x P(b)] )`
4. `NPMI(a,b) = PMI(a,b) / -ln(P(ab))` - величина PMI нормализованная в диапазон [-1, 1];

Примеры и комментарии:

- значение NPMI равное “-1” будет означать, что пара слов никогда не встречается в датасете. Например, такие пары как **“green idea”** или **“sleeps furiously”** никогда не встречаются вместе, поэтому `P(ab) = 0`, следовательно `PMI(a,b) = -inf, NPMI = -1`;
- значение NPMI равное “0” будет означать, что слова в паре встречается абсолютно независимо друг от друга. Рассмотрим пример **"the doors"**: “the” может встретится рядом с любым словом. Таким образом, `P(ab) = P(a) x P(b)` и `PMI(a,b) = ln(1) = 0, NPMI = 0`;
- значение NPMI равное “1” будет означать, что это идеальная коллокация. Предположим, что **"Roman Empire"** - это уникальная комбинация, и за каждым появлением “Roman” следует “Empire”, и, наоборот, каждому появлению “Empire” предшествует “Roman”. В этом случае `P(ab) = P (a) = P(b)`, поэтому `PMI(a,b) = -ln(P(a)) = -ln(P(b))`, следовательно `NPMI = 1`.

Условия:

- найти самые популярные коллокации в Википедии;
- для однозначности вычислений, выделите слова из статьи с помощью регулярного выражения **`re.findall(r"\w+", text)`**;

- привести все слова к нижнему регистру;
- удалить стоп-слова;
- слова в паре объединить символом нижнего подчеркивания “_”;
- отфильтровать биграммы, которые встретились **не реже** 500 раз (т.е. проводим все необходимые join'ы и считаем NPMI только для них, НО оценку вероятности встретить биграмму, считаем на полном датасете);
- отсортировать слова в выводе по значению NPMI;
- вывести **TOP-39** самых популярных коллокаций и их значения NPMI (округляем до 3-го знака после запятой, см. round);
- решение должно отрабатывать в течение 3х минут на свободном кластере (3 ноды x 8 CPU x 16GB RAM).

Формат вывода:

```
word_pair <tab> npmi
```

Пример вывода:

```
...
south_africa      0.619
roman_empire      0.603
...
```

5. Правила оформления задания

Оформление задания:

- Код задания (Short name): **HW5:Spark-RDD**.
- Выполненное ДЗ запакуйте в архив **X5BD2020Q3**_<Surname>_<Name>_HW#**.zip**, например, для Алексея Драля -- **X5BD2021Q1**_Dral_Alexey_**HW5**.zip. Если ваше решение лежит в папке my_solution_folder, то для создания архива hw.zip на Linux и Mac OS выполните команду[2]:
  - zip -r hw.zip my_solution_folder/*
- На Windows 7/8/10: необходимо выделить все содержимое директории `my_solution_folder/` нажать правую кнопку мыши на одном из выделенных объектов, выбрать в открывшемся меню "Отправить >", затем "Сжатая ZIP-папка". Теперь можно переименовать архив.
- Решения заданий должно содержаться в одной папке.
- PySpark-скрипты для запуска решений следует называть `task_<Surname>_<Name>_<#task_ID>.py`:

[2] Флаг -r значит, что будет совершен рекурсивный обход по структуре директории

  - решение задачи #1 должно называться "task_*_bigram.py" и его можно запустить с помощью команды:
    - PYSPARK_DRIVER_PYTHON=python3.6 PYSPARK_PYTHON=python3.6 spark-submit "task_*_bigram.py"
  - решение задачи #2 должно называться "task_*_collocation.py" и его можно запустить с помощью команды:
    - PYSPARK_DRIVER_PYTHON=python3.6 PYSPARK_PYTHON=python3.6 spark-submit "task_*_collocation.py"
  - скрипты выводят на экран (STDOUT) указанное в задании число строк в нужном формате
- **Вывод STDOUT задач нужно сохранить в соответствующих файлах в архиве посылке домашнего задания (например, `task_*_bigram.out`).**[3]
- Перед проверкой убедитесь, что дерево вашего архива выглядит так:
  - **| X5BD2021Q1_<Surname>_<Name>_HW5.zip**
  - **| ---- task_<Surname>_<Name>_bigram.py**
  - **| ---- task_<Surname>_<Name>_bigram.out**
  - **| ---- task_<Surname>_<Name>_collocation.py**
  - **| ---- task_<Surname>_<Name>_collocation.out**
  - При несовпадении дерева вашего архива с представленным деревом, ваше решение будет невозможно автоматически проверить, а значит, и оценить его.
- Для того, чтобы сдать задание необходимо:
  - Зарегистрироваться и залогиниться в сервисе Everest
  - Перейти на страницу приложения: BDT-grader-X5-BD
  - Выбрать вкладку Submit Job (если отображается иная).
  - Выбрать в качестве “Task” значение: **HW5:Spark-RDD**[4]
  - Загрузить в качестве “Task solution” файл с решением
  - В качестве Sender ID указать тот, который был выслан по почте
- Если Вы видите надпись "You are not allowed to run this application" во вкладке Submit Job в Everest, то на данный момент сдача закрыта (нет доступных для сдачи домашних заданий, по техническим причинам или другое). Попробуйте, пожалуйста, еще раз через некоторое время. Если Вы еще ни разу не сдавали, у коллег сдача работает, но Вы видите такое сообщение, сообщите нам об этом.
- Ситуации:
  * система оценивания показывает оценку (Grade) < 0, а отчет (Grading report) не помогает решить проблему (пример помощи: в случае неправильно указанного Sender ID система вернет -2 и информацию о том, что его нужно поправить);
  * показывает 0 и в отчете (Grading report) не указано, какие тесты не пройдены.

[3] Для подготовки архива с решением и выводом результатов запуска можно воспользоваться командой “tee”
[4] Сервисный ID: spark.rdd.onsite_hw

Если Вы столкнулись с какой-то из них присылайте ссылку на выполненное задание (Job) на почту с темой письма “Short name. ФИО.”. Например: “**HW5:Spark-RDD**. Иванов Иван Иванович.”
Пример ссылки: https://everest.distcomp.org/jobs/67893456230000abc0123def
Внимание: Если до дедлайна остается меньше суток, и Вы знаете (сами проверили или коллеги сообщили), что сдача решений сломана, обязательно сдайте свое решение и напишите письмо, как написано выше, чтобы мы видели, какое решение Вы имели до дедлайна и смогли его оценить.

- Перед отправкой задания, оставьте, пожалуйста, отзыв о домашнем задании по сссылке: http://rebrand.ly/x5bd2021q1_feedback_hw05. Это позволит нам скорректировать учебную нагрузку по следующим заданиям (в зависимости от того, сколько часов уходит на решение ДЗ), а также ответить на интересующие вопросы.

Любые вопросы / комментарии / предложения можно писать в телеграм-канал курса или на почту bigdata_x52021q1@bigdatateam.org .

Всем удачи!