Тестовое задание

ML&Texts

NB

Перед началом выполенения хотим предпуредить вас о двух важных моментах:

1. После этого набора заданий мы хотим провести еще интервью с вами. Это нужно для того, чтобы познакомиться, рассказать о мастерской и ответить на ваши вопросы. Летняя Школа придерживается довольно строгого формата, и нам нужно соответствовать. К сожалению, мы вас выбираем не только по техническим навыкам, но по тому, что нызывается софт-скилы. Поэтому технических вопросов там не будет, мы просто пообщаемся :) О дате интервью мы вам отдельно напишем на почту и согласуем удобное время.

2. Если вы считаете, что в прошлом году выполнили задание идеально, мы учтем его, но пришлите его еще раз, чтобы мы понимали, что вы еще хотите участововать. Если сомневаетесь, присылайте новый вариант.

Задача 1

Дан корпус текстов (на выбор корпус 1 "паблик "Летней школы" вконтакте с комментариями" (если ваших мощностей компьютера не хватает для всех данных, используйте часть, но обязательно это укажите в решении) или корпус 2 "COVID-19 Open Research Dataset"). Найдите и посчитайте в датасете самые частотные слова. Изобразите частотный список на графике.

Если вы работаете с дампом паблика, то, возможно, стоит отделить расчеты для комментариев (кстати, мы не уверены в том, что в дамп попали **все** комментарии :)) от расчетов для самих постов в паблике.

Если вы работаете с биомедицинским датасетом, то, возможно, стоит отделить расчеты для абстрактов статей от расчетов для текстов статей. В этом задании нам в первую очередь интересно то, что вы думаете про текстовые данные и на что в них вам интересно смотреть — поэтому жестких ограничений нет: анализируйте в датасете то текстовое, что захотите.

Пожалуйста, проявите фантазию, и сделайте вывод по полученному анализу.

Используйте для решения R или Python. Код нужно выслать в виде .R.(или Rmd) или .py (или .ipynb).

Далее выберите одну на выбор (2.0 или 2.1), но не обманывайтесь легкостью первой задачи, она с подвохом.

Задача 2.0

Напишите рекурсию, которая принимает на вход определенное число (N), и затем печатает числа от 1 до N. Важное условие: числа не должны храниться в оперативной памяти (то есть внутри функции не должны создаваться доп. переменные.)

Используйте для решения R или Python. Код нужно выслать в виде .R.(или Rmd) или .py (или .ipynb).

Задача 2.1

Представьте себе пользовательскую папку на компьютере. Что-то типа `C:\Users\username` или `/home/username`. В ней наверняка есть еще какие-то папки, и в них тоже какие-то папки, и т.д. А в папках еще и файлы какие-то иногда лежат. Ваша задача — получить общий частотный список слов для папки и всего текстового, что в ней есть.

Ниже приведен пример данных и интуиция того, что нужно для них получить. Еще ниже — пояснения и конкретные формулировки скользких мест задачи.

Например, для такой папки:

```
/letnyayashkola/dirs_task
├── dir1
│   ├── cat_says.txt
│   └── dir2
├── dir3
│   └── dog_says.txt
├── dir4
│   └── dir5
│       ├── cat_says.txt
│       └── dir6
├── dir7
│   ├── dir8
│   └── good_boi_says_sometimes.txt
└── dir9
    └── dir10
        └── dir11
            ├── panther_says.txt
            └── tiger_says.txt
```

С примерно таким содержимым:

```
first 2 words from dir9/dir10/dir11/tiger_says.txt :
meow
meow
first 2 words from dir9/dir10/dir11/panther_says.txt :
meow
roar
first 2 words from dir7/good_boi_says_sometimes.txt :
woof
bark
first 2 words from dir1/cat_says.txt :
meow
meow
first 2 words from dir4/dir5/cat_says.txt :
meow
meow
first 2 words from dir3/dog_says.txt :
woof
woof
```

Частотный список, кажется, будет таким, что meow — самое частотное слово, woof — второе по частоте, и так далее. Что-то типа такого: `{"meow": 15, "woof": 10, "roar": 6, "bark": 3, ..}`

В задаче стоит считать словами все, что отделено друг от друга пробелами.

Пунктуации не существует.

В папках бывают текстовые и бинарные файлы: нас интересуют только текстовые. Все текстовые файлы записаны в `utf-8`. Как понять по файлу, текстовый он или бинарный, — можно погуглить. **Гугление вообще приветствуется** при решении задач. Скрытых папок не существует. Данных достаточно мало для того, чтобы ограничений по памяти или глубине рекурсии не возникало. **Подвохов здесь не запланировано**.

Для решения задачи создайте у себя на компьютере подобную структуру и напишите код, для получения частотного списка. Вы вольны проявить фантазию в решении и использовать для этого к примеру рекурсию.

Используйте для решения R или Python. Код нужно выслать в виде .R.(или Rmd) или .py (или .ipynb).

Задание 3

Скачайте из проекта gutenberg текст "Женское международное движение: Сборник статей"(или напрямую здесь) Используя список русских стопслов, уберите из текста служебные слова, посчитайте и визуализируйте 20 самых частотных слов текста.

Используйте для решения R или Python. Код нужно выслать в виде .R.(или Rmd) или .py (или .ipynb).

Задание 4

Вам нужно сохранить у себя на компьютере статьи из Википедии(1, 2, 3). Файл, естественно, должен быть в `utf-8`. Программа должна читать этот файл и заменять в нем все формы слова A на соответствующие формы слова B (слова A и B тоже указаные здесь). То, что получится, она должна записывать в другой текстовый файл. Все входные и выходные файлы сохраните в своей папке с кодом.

Заменяться должны только формы этого слова. Т. е. если Вам нужно заменить слово «кит» на слово «кот», слово «китовый» на слово «котовый» заменяться не должно. При замене нужно пользоваться функцией `re.sub`. Если слово было написано с большой буквы, то и после замены оно должно быть написано с большой буквы.

Используйте для решения R или Python. Код нужно выслать в виде .R.(или Rmd) или .py (или .ipynb).

Задача 5*

ЗАДАЧА СО ЗВЕЗДОЧКОЙ

Эта задача с повышенной сложностью, поэтому ее выполнение не будет влиять на результат. НО, мы бы хотели, чтобы вы ее выполнили, для того, чтобы примерно понимать Ваш уровень.

20 человек участвуют в голосовании и выбирают одного из 10 кандидатов. Какая вероятность того, что хотя бы один кандидат не получит ни одного голоса, если голосующие не имеют предпочтений и случайно выбирают кандидатов?

Решите задачу на R или Python, используя метод Монте-Карло.Гуглить и использовать интернет можно и нужно. Код нужно выслать в виде .R.(или Rmd) или .py (или .ipynb).

Напутствие

Нам бы очень хотелось, чтобы вы постарались решить всё, но мы понимаем, что задания могут показаться сложными, поэтому решите сколько сможете, и если у вас возникают сложности, пишите их прямо в своем ответе, мы постараемся прислать фидбек с пояснением сложным моментов.

Решения присылать сюда: nlp@letnyayashkola.org до 30го(включительно) мая.