

#### 4. Анализ тональности отзывов

Теперь, когда у вас уже получается делать неплохие решения задачи sentiment-анализа, пора научиться собирать для вашего алгоритма интерактивную демонстрацию в виде простенькой веб-странички, чтобы на него могли посмотреть в действии не только те, кто может запустить `ipython notebook`.

## Задание, оцениваемое сокурсниками: Демонстрация работы алгоритма

Срок сдачи задания: 18 июня 2016 года, 11:59 PDT

### Важная информация

Необходимо отправить это задание до завершения срока сдачи, июнь 18, 11:59 вечера PDT, поскольку другие учащиеся должны его оценить. Если поздно отправить задание, для его проверки может не оказаться сокурсников. Это затрудняет оценивание, а в некоторых случаях делает его невозможным. Чтобы избежать этих рисков, сдавайте задания вовремя.

1. [Инструкции](#)
2. [Моя работа](#)
3. [Обсуждения](#)

### Инструкции

Часто оказывается полезно представить результаты вашей работы в виде интерактивной демонстрации, в которую можно будет вводить какие-то данные и получать результат работы вашей модели. В этом задании вам предлагается реализовать такую демонстрацию для вашего алгоритма анализа тональности отзывов на фильмы, используя `flask`.

Даже если вы никогда не сталкивались с веб-программированием и `flask`'ом - познакомиться с ним совершенно не сложно. Сделать первые шаги вам поможет прекрасный Викиучебник по Flask: <https://ru.wikibooks.org/wiki/Flask>

Разработку демки для вашей задачи существенно упростит архив кода для аналогичной демки в аттаче (с которым все равно проще разбираться после основ Flask):

`simple_demo.zip`

В процессе вы столкнетесь с необходимостью сохранять в файл и загружать питоновский объект - обученный классификатор. Для этого вам пригодится библиотека `pickle`.

### Review criteria меньше

При оценке задания будет проверяться наличие скриншотов демки, кода и описания, как его запустить.