Кафедра АД, «Автоматическая обработка текстов»

Задание 1
Классификация текстов: спам-фильтр для SMS

Описание

В этом задании вам предстоит взять открытый датасет с SMS-сообщениями, размеченными на спам ("spam") и не спам ("ham"), построить на нем классификатор текстов на эти два класса, оценить его качество с помощью кросс-валидации, протестировать его работу на отдельных примерах, и посмотреть, что будет происходить с качеством, если менять параметры вашей модели.

Организационные вопросы

Для сдачи задания выложите IPython/Jupiter notebook с кодом на github или nbviewer, и пришлите на почту xead@yandex-team.ru письмо со ссылкой на код и ответами, которые вы получили в пунктах 5-11. Тема письма должна иметь вид [АД Тексты 2016] Фамилия Имя – Задание 1 – классификация текстов.

Задание

1. Загрузите датасет по ссылке:
   http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/smsspamcollection.zip (описание датасета можно посмотреть здесь)

2. Считайте датасет в Python (можете сразу грузить все в память, выборка небольшая), выясните, что используется в качестве разделителей и как проставляются метки классов.

3. Подготовьте для дальнейшей работы два списка: список текстов в порядке их следования в датасете и список соответствующих им меток классов. В качестве метки класса используйте 1 для спама и 0 для "не спама".

4. Используя sklearn.feature_extraction.text.CountVectorizer со стандартными настройками, получите из списка текстов матрицу признаков X.

5. Оцените качество классификации текстов с помощью LogisticRegression() с параметрами по умолчанию, используя sklearn.cross_validation.cross_val_score и посчитав среднее арифметическое качества на отдельных fold'ах. Параметр cv задайте равным 10. В качестве метрики качества используйте f1-меру.

Получившееся качество – ответ в этом пункте.

6.А теперь обучите классификатор на всей выборке и спрогнозируйте с его помощью класс для следующих сообщений:

"FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk time to use from your phone now! Subscribe6GB"
"FreeMsg: Txt: claim your reward of 3 hours talk time"
"Have you visited the last lecture on physics?"
"Have you visited the last lecture on physics? Just buy this book and you will have all materials! Only 99$"
"Only 99$"
Выпишите через пробел прогнозы классификатора (0 – не спам, 1 – спам)

7. Задайте в CountVectorizer параметр ngram_range=(2,2), затем ngram_range=(3,3), затем ngram_range=(1,3). Во всех трех случаях измерьте получившееся в кросс-валидации значение f1-меры, округлите до второго знака после точки, и выпишете результаты через пробел в том же порядке. В данном эксперименте мы пробовали добавлять в признаки n-граммы для разных диапазонов n - только биграммы, только триграммы, и, наконец, все вместе - униграммы, биграммы и триграммы. Обратите внимание, что статистики по биграммам и триграммам намного меньше, поэтому классификатор только на них работает хуже. В то же время это не ухудшает результат сколько-нибудь существенно, если добавлять их вместе с униграммами, т.к. за счет регуляризации линейный классификатор не склонен сильно переобучаться на этих признаках.

8. Повторите аналогичный п.7 эксперимент, используя вместо логистической регрессии MultinomialNB(). Обратите внимание, насколько сильнее (по сравнению с линейным классификатором) наивный Байес страдает от нехватки статистики по биграммам и триграммам.

9. Попробуйте использовать в логистической регрессии в качестве признаков Tf*idf из TfidfVectorizer на униграммах. Повысилось или понизилось качество на кросс-валидации по сравнению с CountVectorizer на униграммах? Обратите внимание, что результат перехода к tf*idf не всегда будет таким - если вы наблюдаете какое-то явление на одном датасете, не надо сразу же его обобщать на любые данные.

10. * Попробуйте получить как можно более высокое качество на кросс-

валидации. Напишите, что пробовали и какое качество получилось.

11. Какие наблюдения и выводы можно сделать из этого задания?