Данные

В качестве данных взяты отзывы о мобильных телефонах с портала Яндекс.Маркет. В качестве данных для папки class были выбраны явно положительные отзывы, а для папки no_class -- явно отрицательные отзывы.
Было выбрано всего по 30 положительных и отрицательных отзывов. По 20 отзывов из каждой папки используется для обучения, по 10 (т.е. оставшиеся отзывы) -- для тестирования.

Классификатор

Наивный Байесовский классификатор реализован в классе NBClassifier. При создании экземпляра этого класса можно указать в параметрах конструктора:

- kind_of_partition - способ разделения. По умолчания текст разделяется на слова - модификатор ‘word’ (по пробельным символам и знакам препинаняи, слова длиной менее трех знаков отсеиваются). Вместо этого можно указать ‘ngram’, а в следующем параметре ngram, указывается длина нграммы, по умолчанию 4.
- ngram - длина граммы.
- stopwords - список слов-исключений.

У экземпляра этого класса есть два основных метода: train и test.

Метод **train** (folder_class, folder_not_class, model_file) - принимает первым аргументом папку с файлами классифицированными как принадлежащие к классу, затем папку с неклассовыми файлами, и имя файла, в который сохранится текущий обученный экземепляр NBClassifier.

Метод **test** (folder_class, folder_not_class, model_file) - аналогично методу train принимает теже параметры, но model_file можно не указывать, тогда используется текущий экземпляр класса. Возвращаются точность, полнота и f-мера.

Во время тестирования для каждого признака (слова или граммы), в конкретном классе, подсчитывается взвешенная вероятность. Взвешенная для того, чтобы сделать классификатор менее чувствительным к редко встречающимся словам. Для этого мы выбрали некую *предполагаемую вероятность*, которой будем использовать, когда информации о рассматриваемом признаке слишком мало, равную 0.5. И приписали вес предполагаемой вероятности, – 1. Это значит, что вес предполагаемой вероятности равен одному слову. Взвешенная вероятность – это средневзвешенное значение, обычной вероятности и предполагаемой вероятности, т.е. (1*0.5 + count*frequency) / (count+1), где count это число документов в этом классе, а frequency это количество вхождения данного признака в документ, отнесенного к этому классу.

Оценки рассчитываются по формулам:

- Бинарная классификация: точность, полнота
  - $P = \frac{tp}{tp+fp}, R = \frac{tp}{tp+fn}, Accuracy = \frac{tp+tn}{tp+fp+tn+fn}, F_1 = \frac{2PR}{P+R}$

	Решение классификатора	
	Класс	Не класс
Класс	*tp*	*fn*
Не класс	*fp*	*tn*

Результаты при случайном разбиении на обучение и контроль находятся в файле cross-validation_results.txt

Классификация в Weka

Предварительная обработка данных

Выборка была преобразована в файла типа arff, а затем данные были преобразованы в вектор слов с помощью фильтра StringToWordVector с такими параметрами (указаны отличные от параметров по умолчанию): IDFTransormation=True, TFTransformation=True, loweCaseTokens=True, minTermFreq=3.

Затем был применён фильтр AttributeSelection: estimator=ChiSquaredAttributeEval, search=Ranker.

В отранжированном списке аттрибутов сверху оказались довольно осмысленные слова: “хороший”, “отличный”, “доволен”.

Классификация

Для классифиации использовался алгоритм NaiveBayes с параметрами по умолчанию.

Результаты классификации

Алгоритм обучался на 70% выборки (Percentage Split):

Correctly Classified Instances 17
Incorrectly Classified Instances 1

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	0.143	0.917	1	0.957	class
0.857	0	1	0.857	0.923	no_class
0.944	0.087	0.949	0.944	0.944	Weighted Avg.

Кросс-валидация с folds=10

Correctly Classified Instances	56
Incorrectly Classified Instances	4

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.933	0.067	0.933	0.933	0.933	class
0.933	0.067	0.933	0.933	0.933	no_class
0.933	0.067	0.933	0.933	0.933	Weighted Avg.