Наивный байесовский классификатор документов в Excel

Самое распространенное применение *наивного Байеса* — классификация документов. Является ли это электронное письмо спамом или наоборот, долгожданной новостью? Эта запись в Twitter — благодушная или сердитая? Нужно ли передавать этот перехваченный звонок по сотовому для дальнейшего исследования федеральным агентам? Вы предоставляете «данные для обучения», например, классифицированные примеры документов, обучающему алгоритму, который в дальнейшем сможет «разбить» новые документы на те же категории, используя имеющиеся знания.[1]

Самый распространенный подход к классификации документов — это использование модели *набор слов* в сочетании с наивным байесовским классификатором. Модель *набор слов* воспринимает документ как беспорядочное множество слов. «Джонни съел сыр» для него то же самое, что «сыр съел Джонни» — и то и другое состоит из одного и того же множества слов: {«Джонни», «съел», «сыр»}.

Небольшое введение в теорию вероятностей. Выражение $\mathrm{p}()$ используется для обозначения вероятности. Например, $\mathrm{p}(\mathrm{A}) = 0{,}2$ означает, что событие A произойдет с вероятностью 20%. Выражения, типа $\mathrm{p}(\mathrm{A}|\mathrm{B})$ используются для обозначения условных вероятностей. Например, $\mathrm{p}(\mathrm{A}|\mathrm{B}) = 0{,}3$ означает, что вероятность события A, при условии, что случилось событие B, составляет 30%. Совместная вероятность $\mathrm{p}(\mathrm{A}, \mathrm{B})$ используется для обозначения вероятности того, что события A и B произойдут одновременно. Если события A и B независимы, то $\mathrm{p}(\mathrm{A}, \mathrm{B}) = \mathrm{p}(\mathrm{A}) * \mathrm{p}(\mathrm{B})$. Если события A и B зависимы, то $\mathrm{p}(\mathrm{A}, \mathrm{B}) = \mathrm{p}(\mathrm{A}) * \mathrm{p}(\mathrm{B}|\mathrm{A})$.

Для удобства манипулирования условными вероятностями Томас Байес доказал теорему (подробнее см. Идеи Байеса для менеджеров):

$$p(X|Y) = \frac{p(Y|X) \times p(X)}{p(Y)}$$

В качестве примера мы изучаем твиты о сервисе для отправки электронных писем – Mandrill.com. При поиске по ключевому слову – *mandrill* – помимо полезных, появляются также ссылки, не имеющие отношения к делу. Наша задача – отфильтровать релевантные твиты. Допустим, ранее мы накопили базу, включающую 300 твитов: 150 – о приложении Mandrill.com, и 150 – других.

Каждый твит мы разбиваем на отдельные слова (называемые жетонами — *token*). Нам важны две вероятности:

$\mathrm{p}\,(\text{приложение} \mid \text{слово1}, \text{слово2}\ \ldots)$
$\mathrm{p}\,(\text{другое} \mid \text{слово1}, \text{слово2}, \ldots)$

Это вероятность того, что твит либо о приложении, либо о чем-то другом, при том, что мы обнаруживаем слова: «слово1», «слово2» и т.д.

Если

$$(1)\ \mathrm{p}\,(\text{приложение} \mid \text{слово1}, \text{слово2}, \ldots) > \mathrm{p}\,(\text{другое} \mid \text{слово1}, \text{слово2}, \ldots)$$

то данный твит — о Mandrill.com. Но как же вычислить эти вероятности? Первый шаг — использование теоремы Байеса, которая позволяет переписать условную вероятность приложения как:

$$(2)\ \mathrm{p}\,(\text{прилож.} \mid \text{слово1}, \text{слово2}, \ldots) = \frac{\mathrm{p}\,(\text{прилож.}) * \mathrm{p}\,(\text{слово1}, \text{слово2}, \ldots \mid \text{прилож.})}{\mathrm{p}\,(\text{слово1}, \text{слово2}, \ldots)}$$

Точно так же

$$(3)\ \mathrm{p}\,(\text{другое} \mid \text{слово1}, \text{слово2}, \ldots) = \frac{\mathrm{p}\,(\text{другое}) * \mathrm{p}\,(\text{слово1}, \text{слово2}, \ldots \mid \text{другое})}{\mathrm{p}\,(\text{слово1}, \text{слово2}, \ldots)}$$

---

[1] Написано по материалам книги Джона Формана Много цифр: Анализ больших данных при помощи Excel. – М.: Альпина Паблишер, 2016. – С. 101–128

Подставив (2) и (3) в (1) и умножив обе части на *p(слово1, слово2, …)*, получим условие (1) в виде:

(4) p(прилож.) * p(слово1,слово2,…| прилож.) > p(другое) * p(слово1,слово2,…| другое)

Применяемое для анализа правило апостериорного максимума (MAP) позволяет, во-первых, не обращать внимание на различие значений p (прилож.) и p (другое), а во-вторых, считать вероятность вхождения слов в твит независимым (хотя это и не так), и заменить:

p(слово1,слово2,…| прилож.) –> p(слово1| прилож.) * p(слово2| прилож.) * …
p(слово1,слово2,…|другое) –> p(слово1|другое) * p(слово2|другое) * …

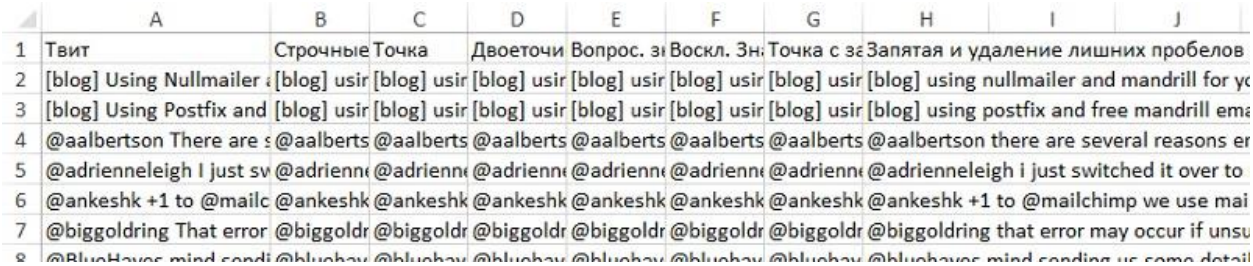
В окончательном виде мы будем сравнивать две величины:

(5) p(слово1| прилож.) * p(слово2| прилож.) * … > p(слово1|другое) * p(слово2|другое) * …

Предположение о независимости позволяет разбить совместную условную вероятность набора слов при известном классе на вероятности нахождения каждого слова в данном классе. Считая слова независимыми, мы вносим в алгоритм MAP множество ошибок, но, в конце концов, они не влияют на правильность выбора между набором, относящимся к приложению и другими твитам.

Осталось решить две проблемы: что делать с редкими словами, и как победить исчезающе малые величины, появляющиеся при перемножении большого числа вероятностей, близких к нулю? Принято добавлять единицу к каждому значению (даже нулевому). Это называется дополнительным сглаживанием и часто используется для приспособления неведомых ранее слов к модели наборов слов. А вместо умножения используется сложение логарифмов. Например, у вас есть произведение: 0,2 * 0,8. Прологарифмируйте его: ln(0,2 * 0,8) = ln(0,2) + ln(0,8).

Итак, все объяснения даны, и можно перейти к Excel. На первых двух листах книги с примерами содержатся по 150 твитов, относящихся к приложению Mandrill.com (рис. 1) и к другим темам. Последовательно в оригинальном тексте твитов все буквы заменяются на строчные, а затем знаки препинания – на пробелы. Например, формула в ячейке E2 =ПОДСТАВИТЬ(D2;"?";" ") – заменяет в тексте, содержащемся в ячейке D2, все знаки вопроса на пробелы.



	A	B	C	D	E	F	G	H	I	J
1	Твит	Строчные	Точка	Двоеточи	Вопрос. зн	Воскл. Зн	Точка с за	Запятая и удаление лишних пробелов		
2	[blog] Using Nullmailer a	[blog] usir	[blog] usir	[blog] usir	[blog] usir	[blog] usir	[blog] usir	[blog] using nullmailer and mandrill for yo		
3	[blog] Using Postfix and	[blog] usir	[blog] usir	[blog] usir	[blog] usir	[blog] usir	[blog] usir	[blog] using postfix and free mandrill ema		
4	@aalbertson There are s	@aalberts	@aalberts	@aalberts	@aalberts	@aalberts	@aalberts	@aalbertson there are several reasons er		
5	@adrienneleigh I just sv	@adrienne	@adrienne	@adrienne	@adrienne	@adrienne	@adrienne	@adrienneleigh i just switched it over to		
6	@ankeshk +1 to @mailc	@ankeshk	@ankeshk	@ankeshk	@ankeshk	@ankeshk	@ankeshk	@ankeshk +1 to @mailchimp we use mai		
7	@biggoldring That error	@biggoldr	@biggoldr	@biggoldr	@biggoldr	@biggoldr	@biggoldr	@biggoldring that error may occur if unsu		
8	@BlueHaves mind sendi	@bluehav	@bluehav	@bluehav	@bluehav	@bluehav	@bluehav	@bluehaves mind sending us some detail		

Рис. 1. Удаление лишних знаков в базе твитов о приложении

Теперь нам необходимо сосчитать, сколько раз каждое слово используется в записях данной категории. Для этого нужно собрать все слова из твитов каждой базы в одном столбце. Предполагая, что каждый твит содержит не более 30 слов, и собираясь присвоить каждому жетону отдельную строку, вам понадобится 150 * 30 = 4500 строк. Создайте новый лист, назовите его *Жетоны_прил.* Назовите ячейку A1 *Твиты*. Скопируйте в буфер значения H2:H151 с листа *Приложение.* Выделите на листе *Жетоны_прил.* область A2:A4501 и кликните *Вставить –> Специальная вставка –> значения* (рис. 2). Нажмите Ok. Обратите внимание: так как вы вставляете 150 твитов в 4500 строк, Excel повторяет все за вас. Это означает, что если вы выделите первое слово из первого твита в строке 2, этот самый твит повторится для выделения второго слова в строке 152, третьего — в 302 и т.д.

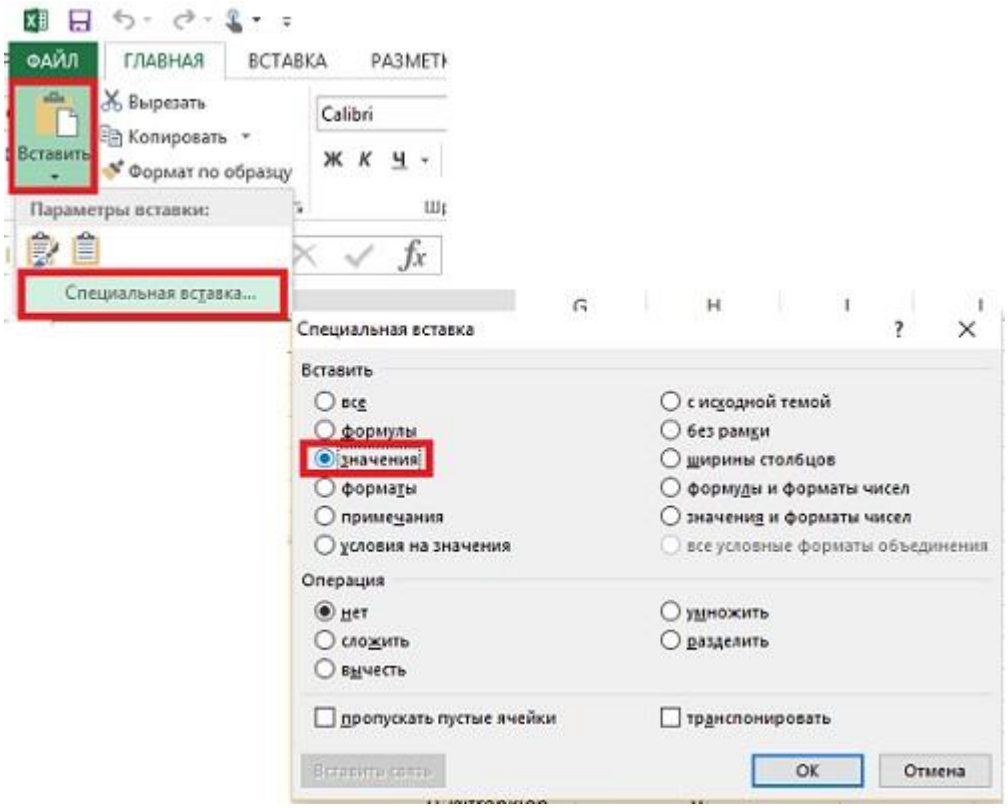
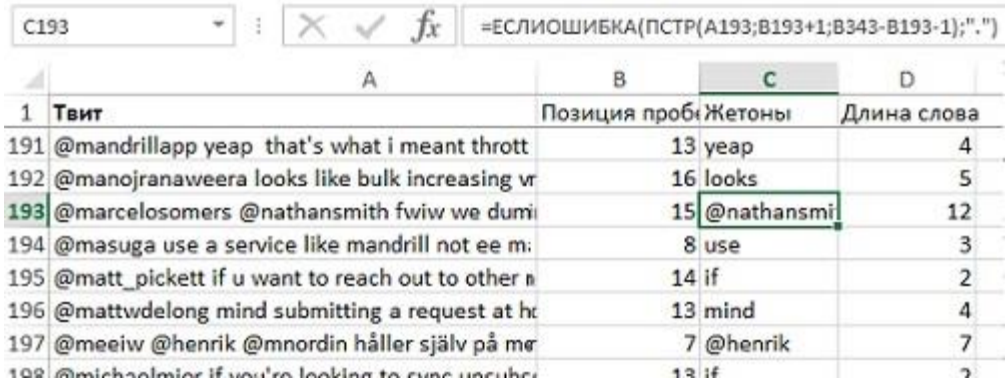


Рис. 2. Специальная вставка

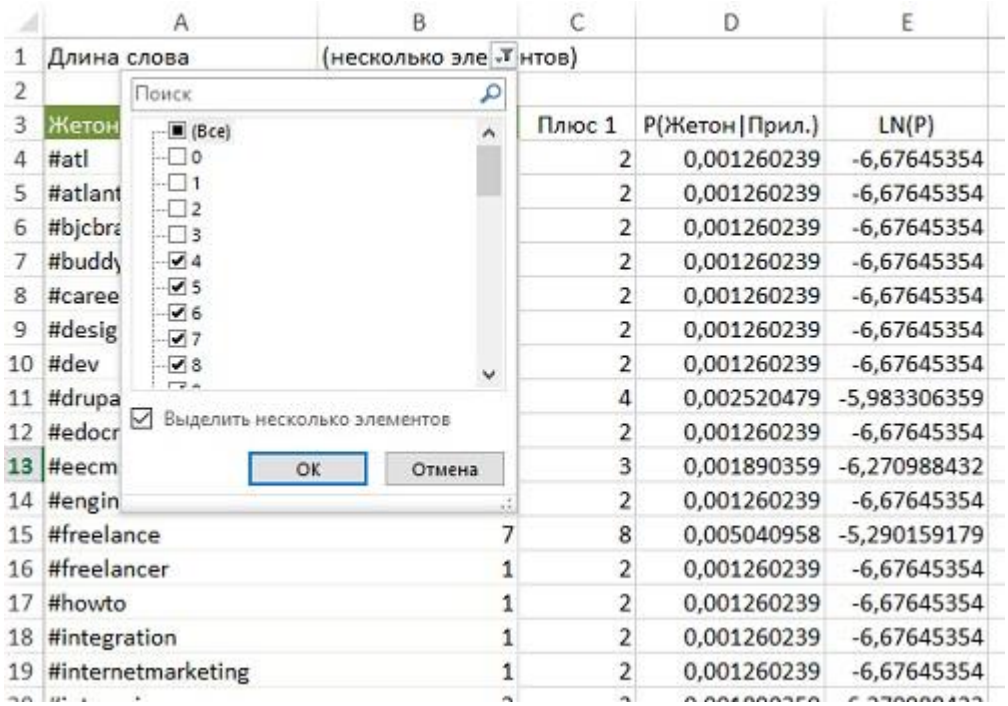
Изучите формулы в столбцах B:D листа *Жетоны_прил.*, чтобы понять механику последовательного извлечения жетонов из твита (рис. 3). Аналогично создайте лист *Жетоны_др.* для базы твитов, не относящихся к приложению Mandrill.com.



	A	B	C	D
1	Твит	Позиция проб	Жетоны	Длина слова
191	@mandrillapp yeap that's what i meant thrott	13	yeap	4
192	@manojranaweera looks like bulk increasing vr	16	looks	5
193	@marcelosomers @nathansmith fwiw we dum	15	@nathansmi	12
194	@masuga use a service like mandrill not ee m	8	use	3
195	@matt_pickett if u want to reach out to other n	14	if	2
196	@mattwdelong mind submitting a request at ht	13	mind	4
197	@meeiw @henrik @mnordin håller själv på mo	7	@henrik	7

Рис. 3. Фрагмент листа *Жетоны_прил.,* на котором извлекаются жетоны из базы данных твитов, относящихся к приложению Mandrill.com

Теперь на базе листа *Жетоны_прил.* следует создать сводную таблицу, которая подсчитает число вхождений каждого жетона. С помощью фильтра сводной таблицы исключите слова длиною до 4 символов, а также добавьте столбцы для подсчета логарифма частоты вхождения жетона (рис. 4). Повторите операцию для листа *Жетоны_др.*



	A	B	C	D	E
1	Длина слова	(несколько эле	нтов)		
2					
3	Жетон		Плюс 1	P(Жетон|Прил.)	LN(P)
4	#atl		2	0,001260239	-6,67645354
5	#atlant		2	0,001260239	-6,67645354
6	#bjcbra		2	0,001260239	-6,67645354
7	#buddy		2	0,001260239	-6,67645354
8	#caree		2	0,001260239	-6,67645354
9	#desig		2	0,001260239	-6,67645354
10	#dev		2	0,001260239	-6,67645354
11	#drupa		4	0,002520479	-5,983306359
12	#edocr		2	0,001260239	-6,67645354
13	#eecm		3	0,001890359	-6,270988432
14	#engin		2	0,001260239	-6,67645354
15	#freelance	7	8	0,005040958	-5,290159179
16	#freelancer	1	2	0,001260239	-6,67645354
17	#howto	1	2	0,001260239	-6,67645354
18	#integration	1	2	0,001260239	-6,67645354
19	#internetmarketing	1	2	0,001260239	-6,67645354

Рис. 4. Логарифмированные вероятности для жетонов, относящихся к приложению

Теперь, когда модель классификатора «обучена», настало время ее использовать. На листе *Тест* размещены 20 твитов, которые нужно классифицировать. Они также обработаны (как и на рис. 1). Поместим подготовленные твиты на лист *Классификация*. Выделите D2:D21 и выберите *ДАННЫЕ* –> *Текст по столбцам*. В появившемся окне выберите *С разделителями* и нажмите *Далее*. На втором шаге выберите знаки табуляции и пробела в качестве разделителей, а также *Считать последовательные разделители одним* (рис. 5). Ограничитель строк установите *(нет)*. Нажмите *Далее*. На последнем шаге *Формат данных столбца* установите *общий*. Нажмите *Готово*.

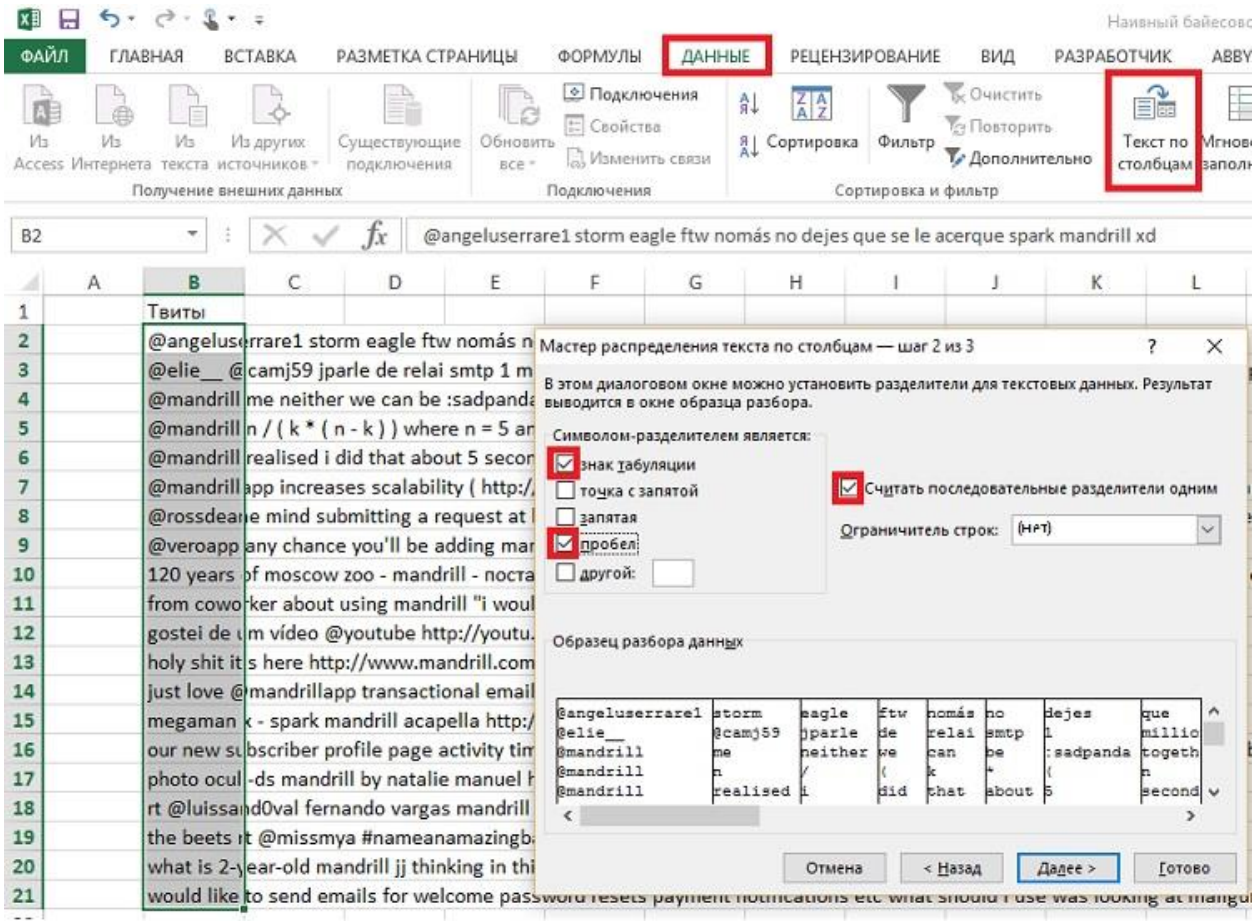


Рис. 5. Разделение тестовых твитов по столбцам

Процедура разбросает твиты по столбцам всего листа до столбца AI (рис. 6).

	A	B	C	D	E	F	G	H	I	J
1	Класс	Жетоны								
2		@angelus	storm	eagle	ftw	nomás	no	dejes	que	se
3		@elie__	@camj59	jparle	de	relai	smtp	1	million	de
4		@mandrill	me	neither	we	can	be	:sadpanda	together	:(
5		@mandrill	n	/	(	k	*	(	n	-
6		@mandrill	realised	i	did	that	about	5	seconds	after
7		@mandrill	increases	scalability	(	http://bit.l	)	then	decreases	pricing
8		@rossdear	mind	submitting	a	request	at	http://help	with	account
9		@veroapp	any	chance	you'll	be	adding	mandrill	support	to

Рис. 6. Жетоны из тестовых твитов

Теперь с помощью функции ВПР извлечем данные о логарифмах вероятностей вхождения тестовых жетонов в два набора данных (приложение / другие). Сравним суммы, и сделаем вывод о принадлежности тестов к тому или иному классу (рис. 7). Выделены цветом твиты, разность логарифмов по которым менее 1. Подробнее с формулами можно ознакомиться на листе *Классификация.*

	A	B	C	D	E
1	Класс	Разность LN(P)	Жетоны		
2	Другое	-5,16	@angelus	storm	eagle
3	Приложение	20,56	@elie__	@camj59	jparle
4	Другое	-3,42	@mandrill	me	neither
5	Другое	-4,71	@mandrill	n	/
6	Приложение	2,03	@mandrill	realised	i
7	Приложение	11,49	@mandrill	increases	scalability
8	Приложение	22,14	@rossdear	mind	submitting
9	Приложение	5,12	@veroapp	any	chance
10	Другое	-3,35	120	years	of
11	Приложение	10,52	from	coworker	about
12	Другое	-10,19	gostei	de	um
13	Приложение	0,91	holy	shit	it's
14	Приложение	17,56	just	love	@mandril
15	Другое	-16,06	megaman	x	-
16	Приложение	9,81	our	new	subscribe
17	Приложение	0,03	photo	oculi-ds	mandrill
18	Другое	-5,88	rt	@luissand	fernando
19	Другое	-4,24	the	beets	rt
20	Приложение	2,59	what	is	2-year-old
21	Приложение	13,86	would	like	to
22					
23	Приложение				
24	-57,33		-7,79	-7,79	-7,79
25	-102,67		-6,27	-5,58	-6,68
26	-29,63		-6,27	0,00	-7,79
27	-46,77		-6,27	0,00	0,00
28	47,54		6,27	6,68	0,00

Рис. 7. Классифицированные тестовые твиты

Вот и все. Модель построена, предположения сделаны.