ЦИФРОВАЯ КУЛЬТУРА

УНИВЕРСИТЕТ ИТМО

Классификация текстов

Высшая школа цифровой культуры

Университет ИТМО

dc@itmo.ru

Оглавление

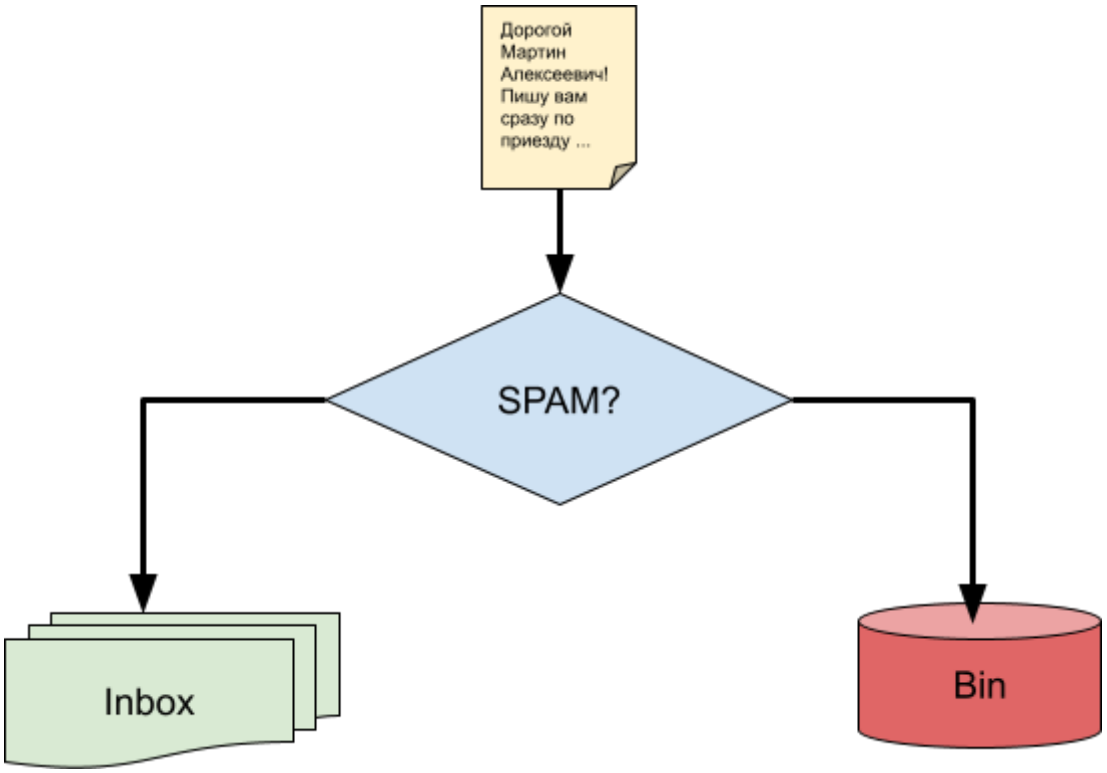
КЛАССИФИКАЦИЯ ТЕКСТОВ

Одна из ключевых задач компьютерной лингвистики и обработки естественного языка - классификация текстов. В общем виде, классификация текстов - это отнесение одного документа к одной из нескольких категорий на основании его содержания.

Прежде чем мы погрузимся в детали и подробнее рассмотрим подходящие для её решения методы машинного обучения, давайте посмотрим на примеры классификации текстов в повседневной жизни.

ОПРЕДЕЛЕНИЕ СПАМА

Каждый день мы получаем десятки писем, но не все из них нам интересны: кроме нужных писем в ящик попадают сообщения от мошенников с фишинговыми ссылками, рекламные рассылки, письма счастья.



Как правило, в почтовый сервер интегрирован спам-фильтр, который автоматически сортирует входящие сообщения, и перемещает в папку Спам все те, которые кажутся ему подозрительными. Как это происходит?

Допустим, человек получает такое письмо. Опытный пользователь сразу поймет, что это спам. Но как? В письме много подозрительных особенностей. Во-первых, в письме много раз упоминается денежное вознаграждение или выигрыш: “денежный сертификат”, “сертификат на сумму 75 159 руб”. Во-вторых, в тексте содержатся настойчивые просьбы быстрее его получить и многократный повтор фразы “заберите свои деньги”. Кроме этого, получателя точно насторожат отсутствие личного обращения (“Поздравляем”), ссылка и почтовый адрес подозрительного вида, а также точное время отправки письма (“23:00”). Объединив все эти признаки, можно классифицировать письмо как спам.

АНАЛИЗ ТОНАЛЬНОСТИ

Ещё одна популярная задача классификации - анализ тональности, определение того, какую эмоциональную нагрузку несет текст - положительную, отрицательную или нейтральную. Такими текстами могут быть обзоры фильмов, отзывы на товары или просто комментарии к новостям на сайте.

Например, на этом слайде приведены отзывы пользователей на мобильные телефоны. Сразу понятно, что первый пользователь доволен покупкой - он использует такие слова и фразы как “отличный”, “классный”, “работает шустро”, “современная прошивка”, “на отличном уровне”. Второй отзыв точно отрицательный: автор пишет, что никому не рекомендует телефон, не может его видеть, и вообще сначала хотел “выбросить в окно”. Задачу сортировки таких отзывов также можно решить автоматически.

ОПРЕДЕЛЕНИЕ ТЕМ НОВОСТЕЙ

Ещё одна область применения классификации текстов - определение тем новостей. Давайте посмотрим на эти заголовки. Кажется, человек, который владеет контекстом, легко определит, к какой теме относятся новости - к новостям спорта или новостям рынка криптовалют. Проще всего ориентироваться по ключевым словам: “фигурное катание”, “болельщики” или “альткоины”. Ключевыми словами могут быть не только термины, но и названия стран, марок или имена персоналий, связанных с темой (Тарасова, Крейг Райт). Выделение именованных сущностей -- отдельная задача, заслуживающая нескольких лекций.

ПРИМЕРЫ ЗАДАЧИ КЛАССИФИКАЦИИ

Для каких других задач можно применять классификацию? На самом деле количество задач в реальной жизни, которое можно решать в реальной жизни, очень велико. В общем и целом, классификация - это определение категории текста по его содержанию. Можно разделять тексты по авторству (какой пользователь написал сообщение на форуме), либо определять возраст или пол автора текста. Также можно классифицировать тексты по их языку или кодировке. Классификация применяется для ограничения тем при выдаче в поисковых системах, подборе контекстной рекламы, определении сообщений ботов в онлайн-дискуссиях.

ФОРМАЛЬНАЯ ПОСТАНОВКА ЗАДАЧИ

Давайте определим задачу классификации с точки зрения машинного обучения. Важно понимать, что классификация текстов - это не их кластеризация. При классификации текстов категории документов уже заранее определены (например, темы новостей или тональность отзывов), в то время как при кластеризации нет информации ни о возможных категориях текстов, а часто и о количестве таких категорий.

Формально задачу можно описать следующим образом. Есть множество документов D - все тексты в выборке, и множество категорий C - все возможные категории, которые можно присвоить документу. Есть неизвестная целевая функция Ф, которая предсказывает категории текстов. Задача - на основе данных построить классификатор Ф’, который будет максимально близок к Ф и сможет прогнозировать категории тех текстов, которые модель не видела при обучении.

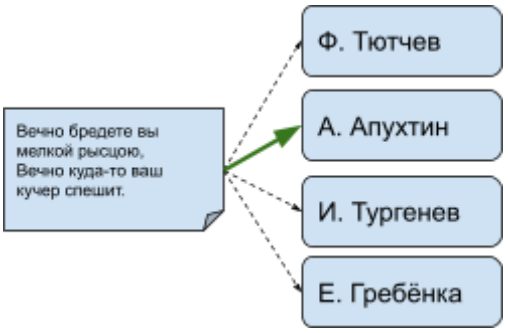
Здесь важно отметить, что у нас нет никакой информации о текстах документа, кроме той, которую можно извлечь из них самих: например, определенные слова или их частотность.

Задача классификации относится к задачам обучения с учителем. Это значит, что для построения алгоритма имеется подвыборка текстов с уже проставленными классами. Её используют для обучения классификатора и определения его параметров, при которых классификатор даст наилучший результат. Систему разделяют на обучающую и тестовую выборки: на обучающей выборке она вырабатывает правила, по которым разделяет документы на классы, а на тестовой выборке проверяется качество разделения.

ОДНОЗНАЧНАЯ И МНОГОЗНАЧНАЯ КЛАССИФИКАЦИИ

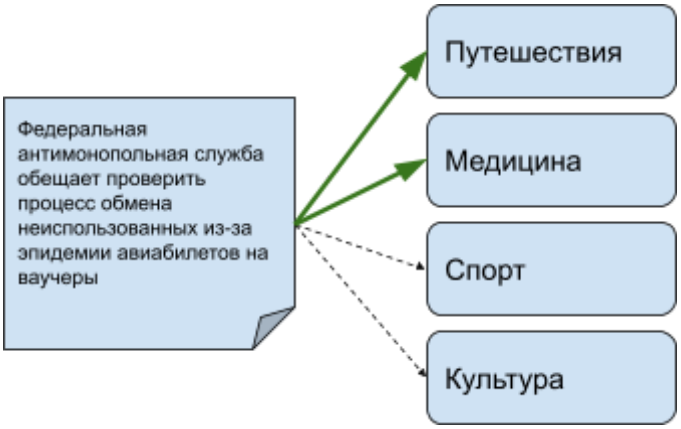
Классификация может быть однозначной или многозначной. Однозначная классификация - такая, при которой одному документу может соответствовать только одна категория.

Однозначная классификация может быть многоклассовой или бинарной. Например, определение авторства текста, если одному тексту может соответствовать только один автор, но при этом нужно выбрать из списка возможных авторов.



Ещё один пример многоклассовой классификации - определение категории новостного текста для распределения текстов по разделам на сайте. При бинарной классификации есть только две непересекающиеся категории: например, обнаружение спама или разделение отзывов на положительные или отрицательные.

Многозначная классификация - классификация, при которой у текста может быть сразу несколько меток. Например, определение темы текста и отображение контента в зависимости от интересов пользователя: новость об изменении стоимости авиабилетов может отобразиться и тем пользователям, которые интересуются отдыхом за рубежом, и тем, кто следит за экономикой.



МЕТОДЫ КЛАССИФИКАЦИИ

ОСНОВНЫЕ ЭТАПЫ РЕШЕНИЯ ЗАДАЧИ КЛАССИФИКАЦИИ

Основные этапы решения задачи классификации: предобработка и индексация документов, уменьшение размерности пространства, построение и обучение классификатора и оценка качества.

О предобработке текста мы говорили раньше: она включает в себя токенизацию, удаление стоп-слов (слишком частотные слова, частицы, предлоги, союзы), приведение слов к нормальной форме. Это позволяет частично сократить размерность пространства.

ИНДЕКСАЦИЯ ДОКУМЕНТОВ

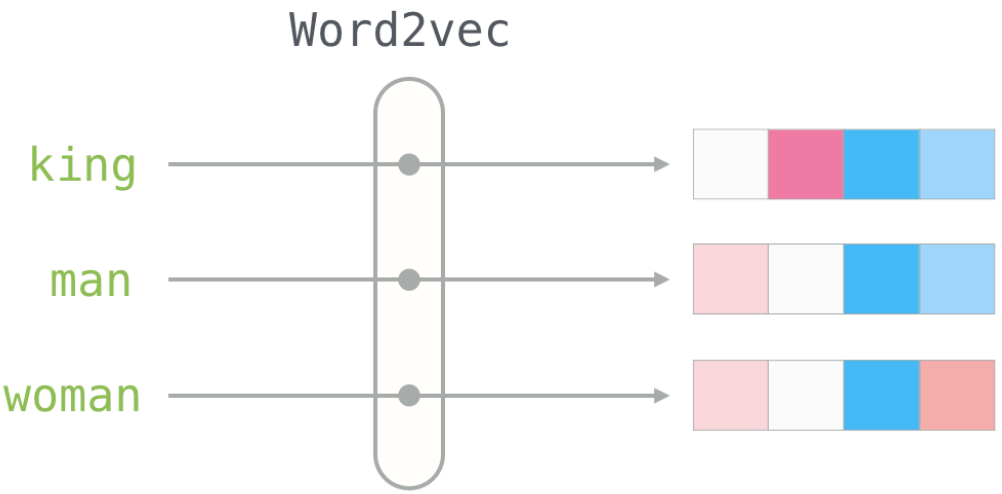
Индексацией документов иногда называют построение такой числовой модели, с помощью которой текст переводится в удобное для предобработки представление.

Один из таких методов **мы уже знаем - это bag of words** или мешок слов. Вектор размером со словарь, ненулевые значения -- частоты термов.

Это - один из самых простых способов представить текст в виде чисел, но в таком случае полностью теряется информация о порядке слов.

Второй способ индексации текстов - **н-граммы**. В таком случае мы подсчитываем не только отдельно стоящие слова или символы, но и их пары, тройки и так далее. Это позволяет, например, учитывать порядок слов или словочетания, отличая “понравится” от “не понравился”, чего не происходит, если мы представляем текст просто как мешок слов, но появляется другая проблема - если учитывать все н-грамы, признаков может быть слишком много. В таком случае рекомендуется удалить слишком частотные или, наоборот, слишком редкие н-грамы и очень внимательно отнестись к выбору N.

Есть также и более современные способы представления текстов, использующие нейронные сети и предсказательные методы дистрибутивной семантики. Например, можно упомянуть семейство методов word2vec.



Каждое слово представляется в виде вектора, который, так сказать, кодирует его смысл на основе информации о его контекстных словах. Путём объединения разными способами таких векторов для текста можно получить его представление, у которого нет проблемы с представлением синонимов и которое неявно содержит в себе информацию о языке, полученную на другой и, возможно, куда б**о**льшей коллекции текстов. Обычно это приносит прирост в качестве результатов.

Дистрибутивная семантика стала особенно важна в последнее время в связи с успехами нейронных сетей, но для нас пока это совсем другая история.

АЛГОРИТМЫ КЛАССИФИКАЦИИ

Существуют разнообразные методы классификации: линейные (например, логистическая регрессия), вероятностные (например, наивный Байес), метрические (метод ближайших соседей), логические (например, деревья принятия решений) и методы, основанные на нейронных сетях. Мы не будем рассматривать все из них, но подробнее остановимся на нескольких - логистической регрессии, методе опорных векторов и наивном Байесовском классификаторе.

ЛИНЕЙНЫЕ АЛГОРИТМЫ

Рассмотрим линейные алгоритмы. Общая идея таких алгоритмов заключается в том, что объекты обучающей выборки представляют собой точки многомерного пространства. Наша цель -- построить такую поверхность, которая отделила бы точки одного класса от точек другого класса. Линейный алгоритм ищет такую линейную

разделяющую гиперплоскость. В двумерном случае гиперплоскостью является прямая линия.

Линейная плоскость задается уравнением: скалярное произведение w на x минус b равно нулю, где икс -- это признаки, а дубльвэ и бэ -- настраиваемые параметры. Далее нужно понять, где находится точка многомерного пространства, соответствующая объекту, относительно линейной плоскости. Для этого нужно посмотреть на знак выражений, которое располагается слева от равенства. Таким образом происходит классификация. Поэтому цель линейных методов сводится к поиску коэффициентов W и константы b, определяющих гиперплоскость.

К однозначным плюсам линейных алгоритмов относятся высокая скорость обучения, интерпретируемость коэффициентов и высокая скорость. Но у них есть и существенный недостаток: если выборка линейна неразделима и зависимость ответов от признаков сложная, точность алгоритма будет невысокой.

ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Одним из линейных алгоритмов является логистическая регрессия, частный случай обобщенной линейной регрессии. Этот метод хорош тем, что прогнозирует не просто ответ “да” или “нет”, но и вероятность отнесения текста к определенному классу. В логистической регрессии веса настраиваются при помощи градиентного спуска. Таким образом минимизируется число ошибок на обучающей выборке. Может получиться так, что число ошибок на обучающей выборке будет небольшим, но при этом алгоритм будет показывать низкую точность на тестовой выборке. Это значит, что алгоритм переобучился - оказался не способен показать такой же качественный результат на новых данных. Чтобы избежать переобучения, к минимизируемой функции добавляется слагаемое, которое зависит только от вектора W. Это слагаемое называется регуляризатором и позволяет бороться с переобучением.

Плюсом логистической регрессии является то, что на выходе мы получаем оценку вероятности отнесения документа к определенному классу. Кроме того, этот алгоритм имеет относительно простую программную реализацию. Недостатком логистической регрессии является сложная интерпретируемость алгоритма и неустойчивость по отношению к выбросам в исходных данных.

МЕТОД ОПОРНЫХ ВЕКТОРОВ

Следующий линейный алгоритм, который мы рассмотрим - это метод опорных векторов, Support Vector Machine. Метод опорных векторов - такой подход к классификации, при котором ищется гиперплоскость, которая наилучшим образом разделяет классы в обучающих данных. Алгоритм устроен так: он ищет точки на графике, которые расположены ближе всего непосредственно к линии разделения. Эти точки называются опорными векторами. Затем, алгоритм вычисляет расстояние между опорными векторами и разделяющей плоскостью. Это расстояние которое называется зазором. Основная цель алгоритма — максимизировать расстояние зазора. Лучшей гиперплоскостью считается такая гиперплоскость, для которой этот зазор является максимально большим.

У метода опорных векторов есть как плюсы, так и минусы. Этот метод хорош тем, что позволяет работать с небольшим набором данных, и дает при этом довольно точный результат. Но у него есть и недостатки. В первую очередь это сложная интерпретируемость параметров алгоритма. Кроме этого, алгоритм неустойчив по отношению к выбросам в исходных данных.

Некогда метод опорных векторов был очень популярен и существует и существует большое количество его расширений, в которых авторы борются с перечисленными недостатками. Отдельного упоминания заслуживает Kernel Trick - использование ядерных функций для метода опорных векторов.

НАИВНЫЙ БАЙЕС

Последний метод, который мы рассмотрим - метод наивный Байес. Если нам известен текст, то задачу можно поставить как определение условной вероятности принадлежности текста к определенной категории при условии, что в нём присутствуют выбранные нами признаки. Например, с какой вероятностью письмо попадет в спам, если в нём присутствует слово “выигрыш”. Такие условные вероятности сложно считать напрямую, поэтому используется формула Байеса, которую вы видите на слайде. Поскольку знаменатель не зависит от класса Y, по которому происходит оптимизация, его можно опустить. Формулу расчета вероятности принадлежности текста к классу можно свести к более простой форме, используя гипотезу независимости. Это значит, что мы предполагаем, что все признаки (в нашем случае - слова в тексте) не зависят друг от друга. Информация о том, что в тексте

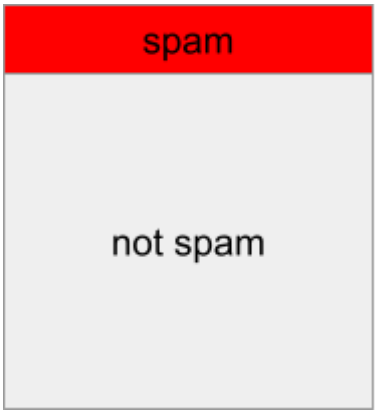
встречается определенный токен, не влияет на вероятность встретить другой. Например, мы считаем, что, если мы встретили в письме слово “выигрыш”, это не влияет на вероятность встретить в том же письме слово “деньги”. В результате мы получаем следующую формулу, которую вы видите на экране. Величины, которые в ней участвуют, настраиваются, основываясь на обучающей выборке - размеченных текстах. Вероятность класса Y задается относительной частотой в обучающей выборке, а вероятность И-того токена можно моделировать по-разному. В самом простом случае вероятность можно задать равной относительной частоте и-того токена в классе Y.

Плюс этого алгоритма в том, что параметры алгоритма могут быть вычислены по небольшому количеству обучающих данных. У алгоритма есть достоинства: высокая скорость работы, он не чувствителен к размерам обучающей выборки, устойчив к переобучению. К недостаткам же относятся в первую очередь невысокая точность классификации и невозможность учитывать зависимость результатов от сочетания различных признаков.

МЕТРИКИ

Рассмотрим на примере бинарной классификации, какие есть подходы к измерению качества классификации. Самый простой способ - подсчитать точность или **accuracy**. В этом случае мы смотрим соотношение корректно определенных классов к общему объектов. Эта метрика очень понятная, но у неё есть свои недостатки.

Первая проблема возникает, когда датасет очень несбалансирован. Например, только 10% писем в вашей выборке являются спамом. Тогда 90-процентную accuracy даёт константа: предсказание, что любое письмо не спам. Сама по себе accuracy как единственное число для оценки качества в этом случае нам почти ничего не скажет, кроме того, справляемся мы лучше константы или нет.



f(x) := ‘not spam’

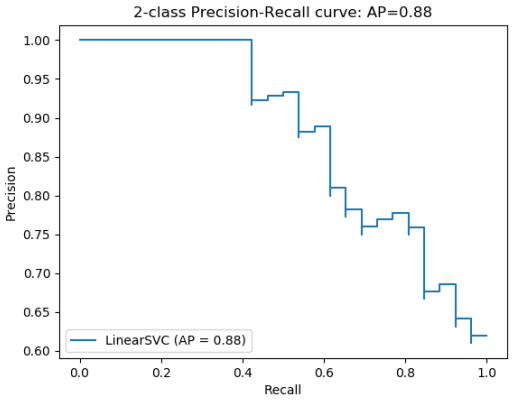
Accuracy = 0.9

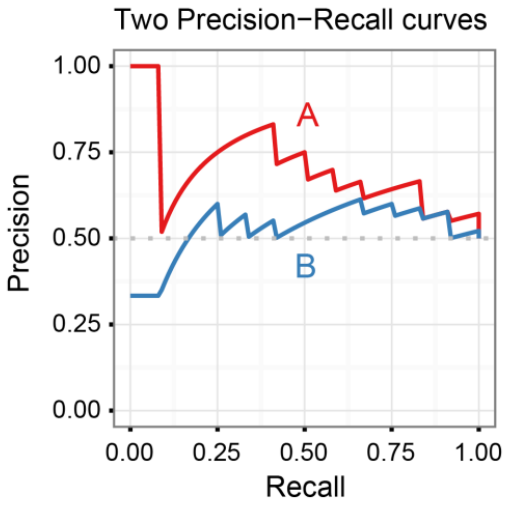
Вторая проблема - невозможность учитывать различные ошибки. Например, вам гораздо важнее определять спам, и стоимость ложного срабатывания, когда нужное письмо попало в спам, очень высока, а обратная ситуация, когда спам не определился, и попал во входящие, неприятна, но не так критична. При подсчете точности невозможно разделить такие случаи.

Поэтому рассматриваются две другие метрики, точность или прецизионность (precision) и полнота (**recall**). Мы различаем четыре типа предсказаний. Первый, true positives (ИП) - нужный нам класс правильно определился. Например, мы хотели найти спам, и нашли его. Второй, false negative (ЛО) - спам не определился и попал в папку “Входящие”. Третий случай, false positive (ЛП) - наоборот, обычное письмо попало в спам. True negatives (ИО) - верное предсказание, письмо, которое не является спамом, не было определено в категорию “спам”.

Precision - процент правильно определенных писем из всех, которые система пометила как “Спам”. Эта метрика хорошо отображает ложные срабатывания алгоритма. Recall же, наоборот, показывает, сколько писем со спамом из всех, существующих в выборке, модели удалось найти. Такие метрики уже гораздо лучше отображают реальность, чем простое измерение точности предсказания.

Чтобы привести обе метрики к одному числу, существует стандартный метод, который называется F1-мера. F1-мера - гармоническое среднее точности и полноты. Если точность или полнота стремятся к нулю, то она также стремится к нулю.





Выбор модели часто определяется тем, что важнее - точность или полнота. Модель, которая предсказывает “да” с небольшим уровнем уверенности, будет иметь высокую полноту и низкую точность, в то время как модель, которая будет давать такое предсказание с высоким уровнем уверенности, будет иметь низкую полноту и высокую точность.

	precision	recall	f1-score	support
class0	0.888	0.877	0.882	308
class1	0.958	0.535	0.687	43
class2	0.712	0.806	0.756	175
accuracy			0.825	526
macro avg	0.853	0.739	0.775	526
weighted avg	0.835	0.825	0.824	526

Для многоклассовой классификации можно вычислять все рассмотренные нами оценки качества для каждого класса по отдельности. То есть мы как бы для каждого класса все остальные сваливаем в один класс и рассматриваем предсказания как задачу бинарной классификации. Так мы сможем посмотреть, насколько хорошо в плане точности и полноты для каждого класса мы справляемся с задачей.

Есть и другие способы оценки классификации, которые также могут пригодиться в обработке естественного языка. Более полный обзор методов классификации, в том числе текстов, можно найти в любом серьёзном учебнике либо курсе по машинному обучению.