Лабораторная работа №7

Работа с текстовыми данными.

Оглавление:

[ХОД РАБОТЫ: 1](#_Toc152018708)

[Моделирование тем и кластеризация документов 1](#_Toc152018709)

[Латентное размещение Дирихле 2](#_Toc152018710)

[ЗАДАНИЕ: 8](#_Toc152018711)

**Цель работы**: изучить методы работы с текстовыми данными, применить их для анализа текстовых наборов.

# ХОД РАБОТЫ:

Изучите материал Лекции 12 «Работа с текстовыми данными».

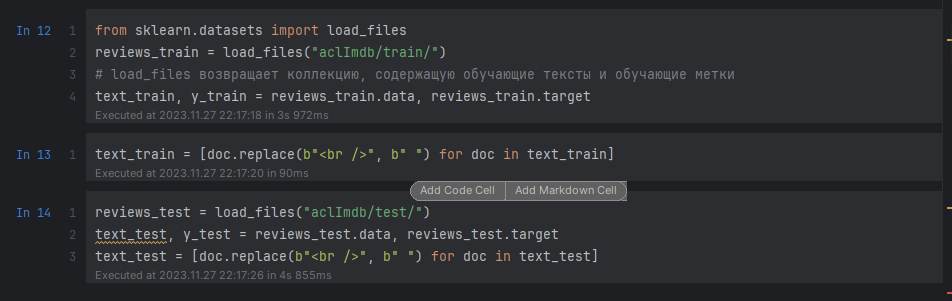
## Моделирование тем и кластеризация документов

Еще один метод, который часто применяется к текстовым данным – **моделирование тем (topic modeling).**

Моделирование тем – это зонтичный термин, описывающий процедуру присвоения каждому документу одной или нескольких тем, которая осуществляется, как правило, без учителя.

Хорошим примером моделирования тем является новостные данные, которые можно сгруппировать по таким темам, как «политика», «спорт», «финансы» и так далее. Если каждый документ может иметь только одну тему, то речь идет о задаче кластеризации документов. Если каждый документ может иметь несколько тем, эта задача относится к декомпозиционным методам. Каждая полученная компонента соответствует одной теме, а коэффициенты компонент, которые описывают документ, позволяют нам судить о том, насколько тесно данный документ связан с конкретной темой. Часто, когда люди говорят о моделирования тем, они имеют в виду конкретный декомпозиционный метод под названием **латентное размещение Дирихле** (Latent Dirichlet Allocation, LDA).[[1]](#footnote-1)

Пример данных с киноотзывами ([http://ai.stanford.edu/~amaas/data/sentiment/.](http://ai.stanford.edu/~amaas/data/sentiment/)):

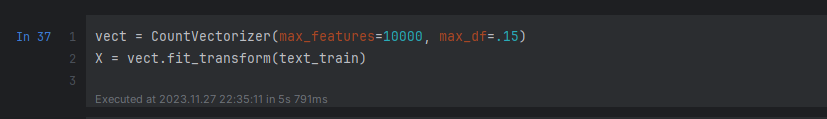


### Латентное размещение Дирихле

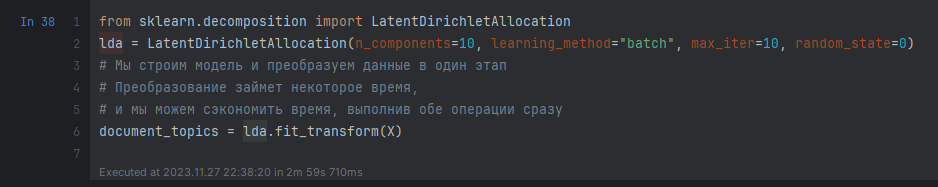
Говоря простым языком, модель **LDA** пытается найти группы слов (темы или топики), которые часто появляются вместе. LDA также подразумевает, что каждый документ можно интерпретировать как «смесь» из нескольких тем. Важно понимать, что для модели машинного обучения «тема» — это далеко не то же самое, что мы подразумеваем под «темой» в повседневной речи. В данном случае «тема» больше напоминает извлекаемые с помощью PCA или NMF компоненты, которые могут иметь или не иметь смысловое значение. Даже если «тема», полученная с помощью LDA, и имеет смысловое значение, все равно она не тождественна «теме» в ее традиционном понимании.

Вернемся, к примеру c новостными данными. Представьте, у нас есть набор статей о **спорте**, **политике** и **финансах**, написанных двумя конкретными авторами. В политической статье мы могли бы ожидать появление слов типа «губернатор», «голос», «партия» и т.д., тогда как в спортивном статье мы могли бы встретить слова типа «сборная», «очко» и «сезон». Слова в каждой из этих групп с большой вероятностью встречаются вместе, в то время как вероятность совместного появления слов «сборная» и «губернатор» будет существенно меньше. Однако это не единственные группы слов, которые, по нашему мнению, встречаются вместе. Возможно, что два репортера предпочитают употреблять различные фразы или различные варианты слов. Возможно, один любит использовать слово «разграничивать», а другому нравится слово «поляризовать». Тогда «темами» уже будут «слова, часто используемые репортером A» и «слова, часто используемые репортером B», хотя они не являются темами в обычном смысле этого слова.

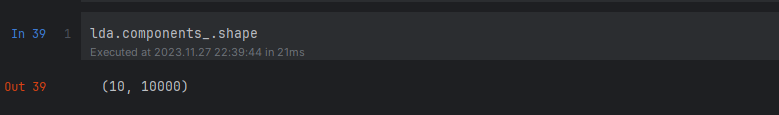
Применим LDA к данным IMDb киноотзывов, чтобы посмотреть, как этот метод работает на практике. Для моделей неконтролируемого обучения, применяющихся к текстовым документам, часто бывает полезно удалить наиболее часто употребляемые слова, поскольку в противном случае они в ходе анализа будут выбраны в качестве самых важных. Удалим слова, которые появляются по крайней мере в 15% документов, и ограничим модель «мешка слов» до 10000 слов, которые представляют собой наиболее часто встречающиеся слова, оставшиеся после удаления:



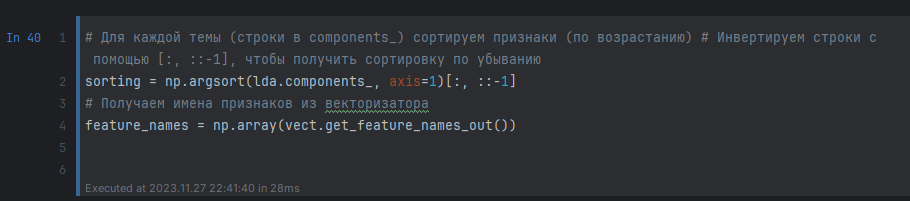
Построим модель, выделив 10 тем, что довольно мало для рассмотрения всех возможных вариантов.[[2]](#footnote-2) Аналогично компонентам в NMF темы не имеют какого-то внутреннего порядка и изменение количества извлекаемых тем изменит содержательную суть всех тем. Воспользуемся методом обучения "batch", который работает несколько медленнее по сравнению с методом "online", установленным по умолчанию, но дает, как правило, лучшие результаты. Кроме того, мы увеличим значение параметра max\_iter (до 25), что также позволить построить модель лучшего качества:

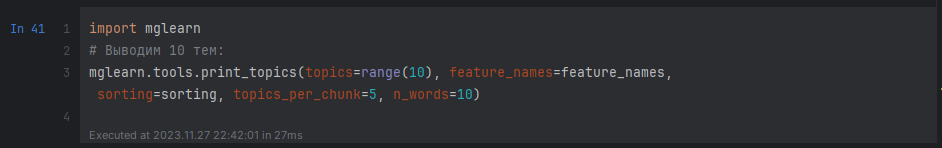


Как и декомпозиционные методы, LatentDirichletAllocation имеет атрибут components\_, который хранит информацию о том, насколько каждое слово важно для каждой выделенной темы. Атрибут components\_ имеет форму (n\_topics, n\_words):



Чтобы лучше понять содержательный смысл каждой темы, мы проанализируем наиболее важные слова для каждой из тем. Функция print\_topics позволяет представить эти признаки в удобном формате:

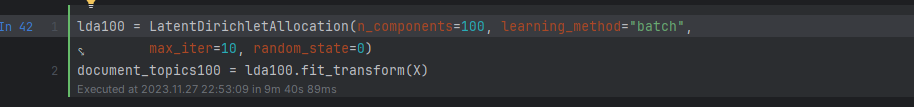




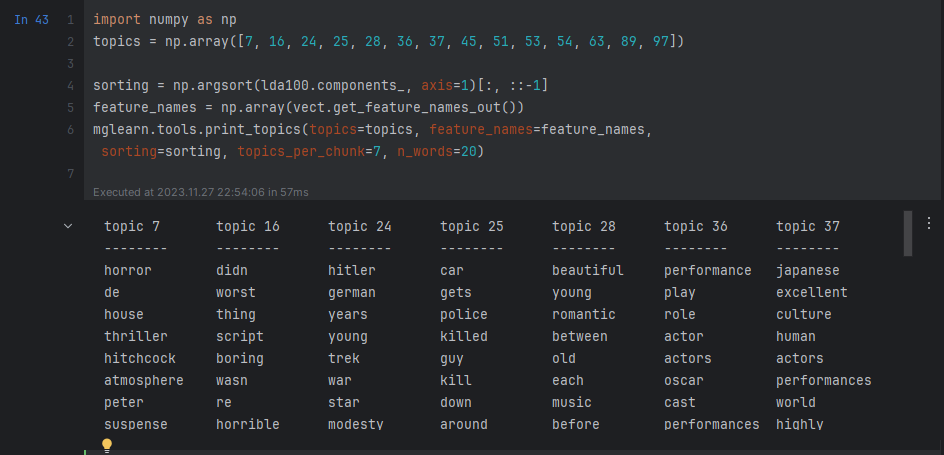


Если судить по словам, связанных с той или иной темой, похоже, что тема 1 соответствует историческим и военным фильмам, тема 2, возможно, связана с плохими комедиями, а тема 3, вероятно, соответствует ТВ-сериалам. Похоже, что тема 4 вобрала в себя некоторые очень распространенные слова, тогда как тема 6, по всей видимости, связана с детскими фильмами. Тема 8, по-видимому, содержит отзывы, связанные с кинонаградами. При n\_topics=10 темы должны быть очень широкими, чтобы вобрать в себя все многообразие киноотзывов.

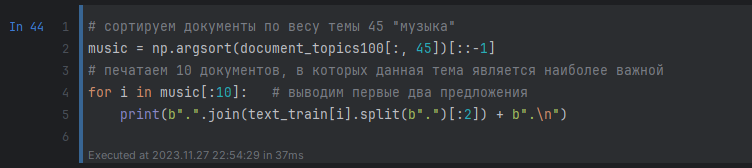
Теперь мы построим еще одну модель, на этот раз выделив 100 тем. Увеличение n\_topics в значительной мере усложняет анализ, но при этом повышает вероятность найти с помощью полученных тем интересные подмножества данных:

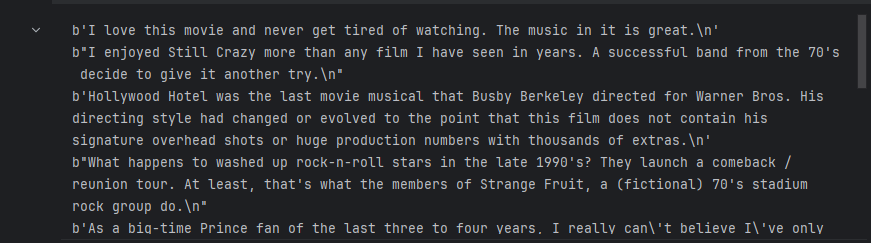


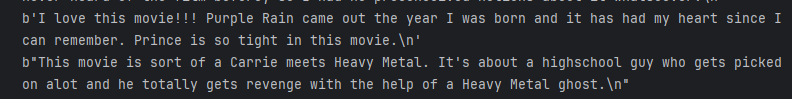
Вывод всех 100 тем было бы немного громоздким, поэтому мы выбрали лишь некоторые интересные и характерные темы:



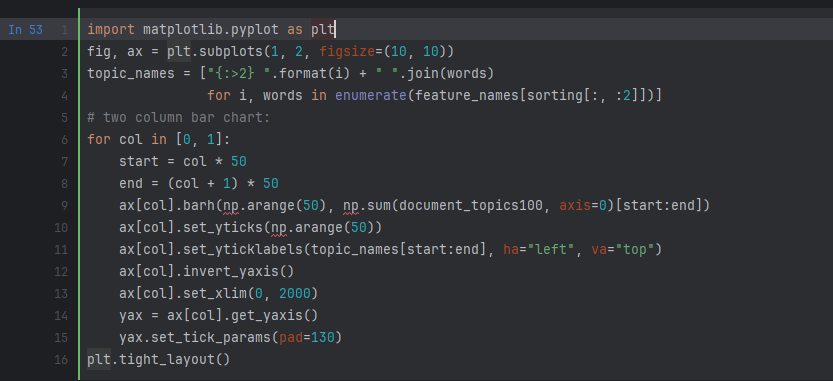
Похоже, что темы, извлеченные на этот раз, более конкретны, хотя многие из них трудно интерпретировать. Тема 7, по-видимому, соответствует фильмам ужасов и триллерам, темы 16 и 54 зафиксировали плохие отзывы, тогда как тема 63 преимущественным образом вобрала в себя положительные отзывы о комедиях. Если мы хотим сделать дополнительные выводы о выделенных темах, мы должны подтвердить свои предположения, выдвинутые нами, исходя из анализа наиболее важных слов по каждой теме. Для этого необходимо взглянуть на документы, которые были отнесены к этим темам. Например, тема 45, похоже, связана с музыкой. Давайте посмотрим, какие отзывы отнесены к этой теме:

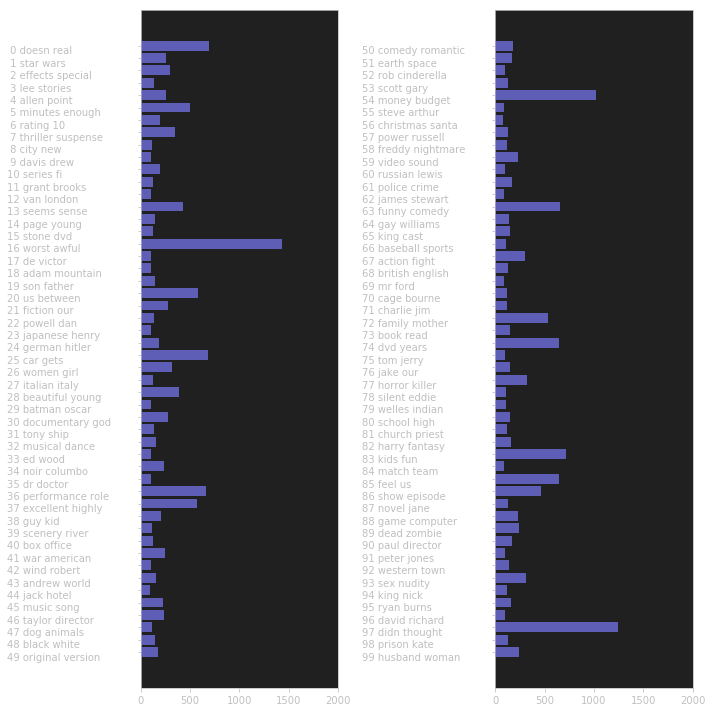






Как видно из вывода, данная тема охватывает широкий спектр музыкальных отзывов, посвященных мюзиклам, биографическим фильмах и трудно определимым жанрам, как в последнем отзыве. Еще один интересный способ исследовать темы – посмотреть, какой вес получает каждая тема в целом, просуммировав document\_topics по всем отзывам. Каждой теме мы дадим названия, используя два самых часто встречаемых слова. Рис. 1 показывает вычисленные веса тем:





*Рис. 1 Веса топиков, полученные с помощью LDA*

# ЗАДАНИЕ:

1. **Изучите материал лекции 12, проделайте пример, описанный в ходе работы.**
2. **Создайте свой «мешок слов», например, набор из 20-25 отзывов о чем-либо, напишите краткое пояснение к данным.**
3. **Создайте словарь стоп-слов, примените стоп-слова к своему «мешку слов»**
4. **Масштабируйте данные с помощью tf-idf**
5. **Исследуйте коэффициенты модели**
6. **Сделайте выводы по работе.**

1. Существует еще одна модель машинного обучения, которую тоже часто сокращенно называют LDA: линейный дискриминантный анализ (Linear Discriminant Analysis), являющийся линейной моделью классификации. Это приводит к некоторой путанице. Здесь под LDA подразумевается латентное размещение Дирихле (Latent Dirichlet Allocation).

   [↑](#footnote-ref-1)
2. На самом деле NMF и LDA решают во многом аналогичные задачи, и мы могли бы также использовать NMF для извлечения тем. [↑](#footnote-ref-2)