Немного о векторизации

CountVectorizer строит представление текста как вектора из счётчиков токенов, по сути Bag-of-Words с массой дополнительных настроек.

TfIdfVectorizer работает аналогично, но при этом каждому токену из словаря сопоставляет не счётчик, а вес tf-idf, поэтому он использует статистику, собранную по всему обучающему множеству. В том числе именно поэтому важно CountVectorizer и TfIdfVectorizer случайно не "обучить" на всём наборе данных, то есть следует применять fit_transform к обучающей выборке, а к тестовой -- уже только transform.

Важнейшие гиперпараметры здесь – это:

- ngram_range -- интервал для задания N в нграммах; так, если он равен кортежу (2, 3), то в словарь мешка слов будут добавляться пары и тройки подряд идущих токенов; по умолчанию значение (1, 1), то есть стандартный мешок слов, где в словаре -- отдельные токены;
- min_df -- число (если int) либо доля (если float) документов, в которых должен встретиться кандидат для занесения в словарь;
- max_df -- то же, только максимум -- используется, чтобы отбросить слишком частотные токены;
- также можно задать **max_features** -- в словаре останется не более заданного числа токенов (топ-max_features по частоте)
- можно подключить также свои стоп-слова, свою токенизацию и так далее.

Векторизаторы на основе счётчиков позволяют использовать различные стратегии фильтрации и взвешивания, настоятельно рекомендуется ознакомиться с документацией, особенно с параметрами **norm, use_idf, smooth_idf, sublinear_tf.**

Они дополнительно хранят в специальных структурах данных различную информацию о наборе данных (например, словарь и частоты). Поэтому иногда они могут занимать много места в памяти, долго загружаться с диска и требовать много времени даже на этапе построения вектора. Как "приближённое" решение этих проблем был предложен HashingVectorizer, способ построения векторов без вызова команды fit_transform. Он использует так называемый hashing trick: по строковому представлению токена вычисляется значение хэш-функции, и в соответствующую по порядку ячейку вектора записывается единица. Этот векторизатор не хранит словарь, и для некоторых токенов хэши могут совпадать, поэтому важно сознательно выбирать размерность путём задания n_features. Подробности о плюсах, минусах и особенностях устройства HV можно прочитать в документации scikit-learn.