Задание:

Нужно реализовать классификатор текстов на основе модели KNN (К ближайших соседей).

У вас есть датасет из 45 статей, по 15 статей на каждую рубрику. Каждую статью представляете в виде вектора чисел — читайте про TF-IDF. Обучаете KNN модель на полученных векторах чисел. Кидаете в модель статью (не из обучающей выборки) — получаете рубрику (класс), к которой она относится.

Прикрепил презентацию с лекции, которая может быть полезной при выполнении задания.

Перед тем, как расчитывать TF-IDF нужно из статей:

- удалить стоп слова (<https://snipp.ru/seo/stop-ru-words>)

- удалить именованные сущности и цифры (библиотека natasha)

- от каждого слова взять только стемму/лемму (библиотека pymorphy)

Отдельными письмами отправил 3 датасета со статьями.

Библиотеки:

- <https://scikit-learn.org/stable/install.html>

- <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

- <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html>

Установка зависимостей:

pip install scikit-learn

Нужно выполнить техническую реализацию и прокомментировать каждую строку вашего алгоритма.

Задача не сложная:

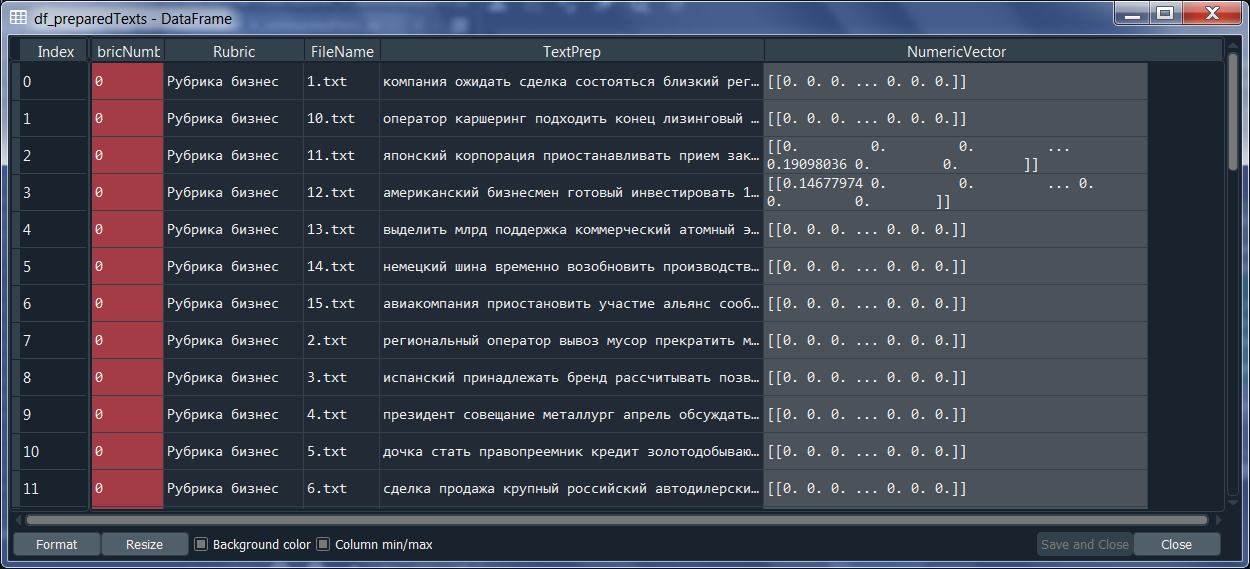
- KNN — одна из самых простых моделей, разобраться можно + много ресурсов с объяснениями.

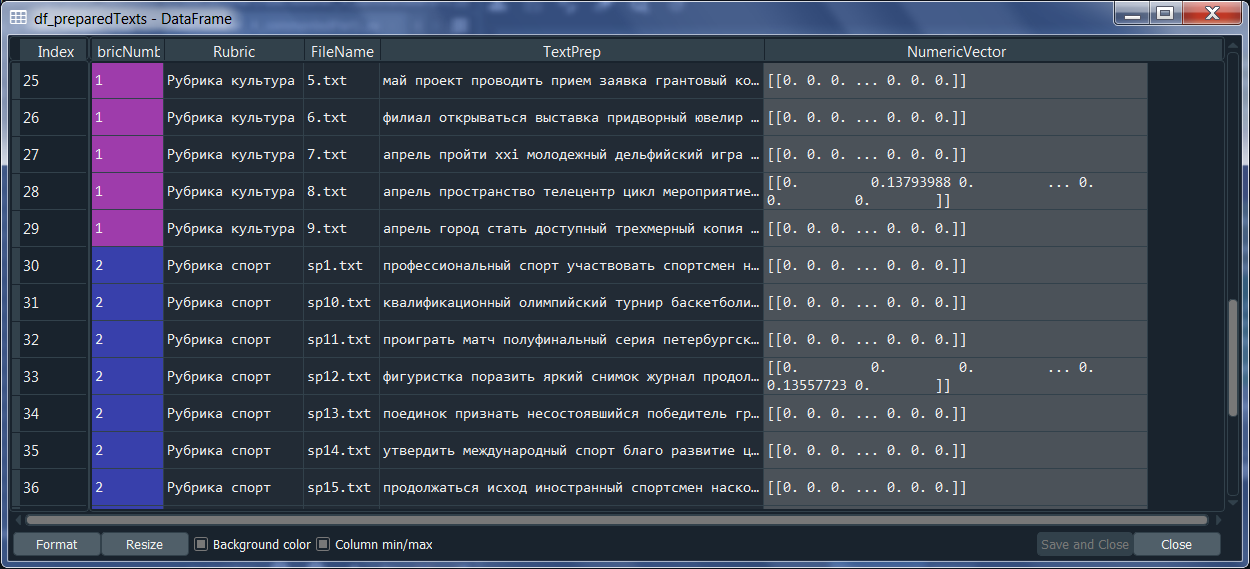
- TF-IDF — довольно простой подход, простая математика (сложить — поделить — перемножить)

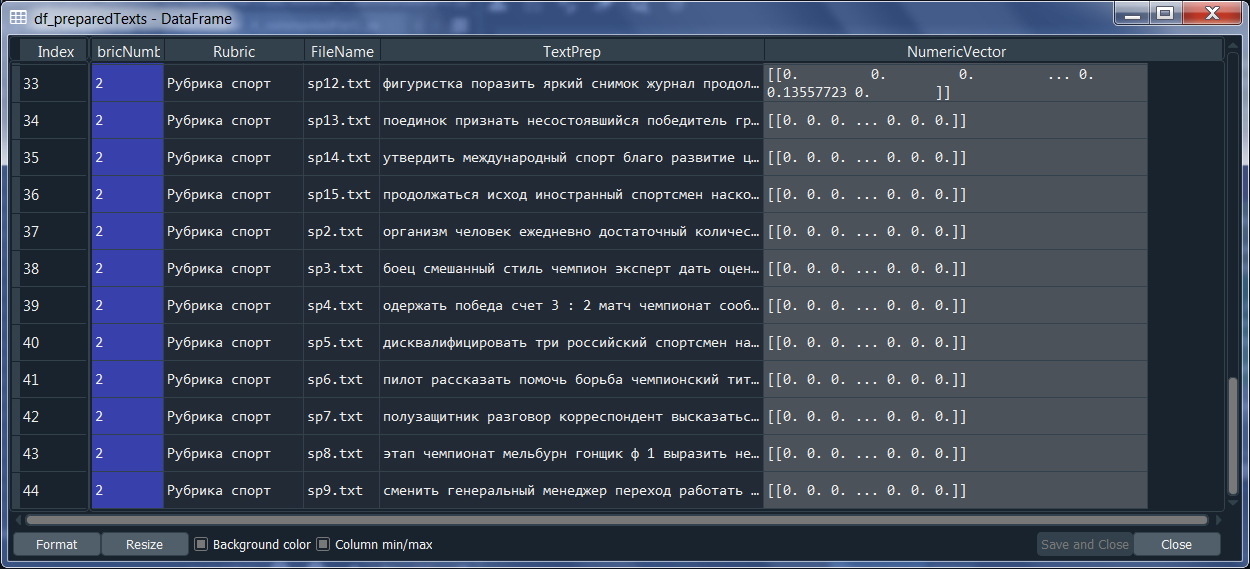
KNN работает с числами, с векторами чисел. Статьи это текст, нужно текст представить в виде чисел. Есть много подходов к решению этой задачи, один из самых простых TF-IDF.

**Решение**

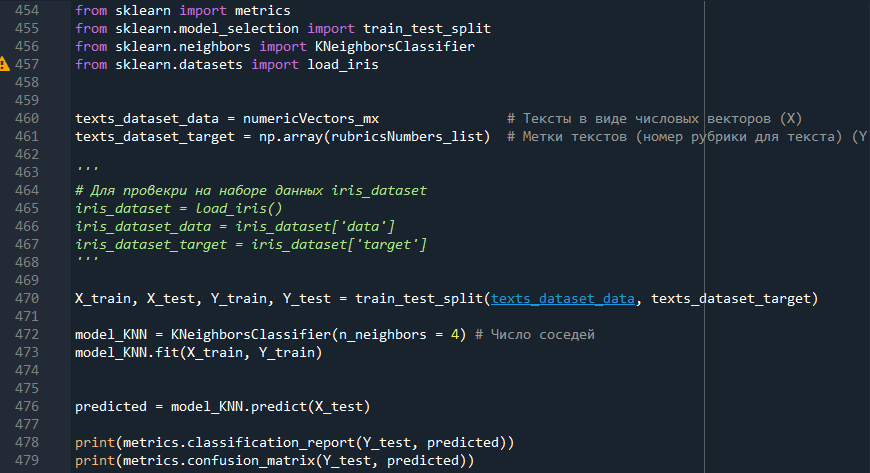
Основная сводная таблица df\_preparedTexts:

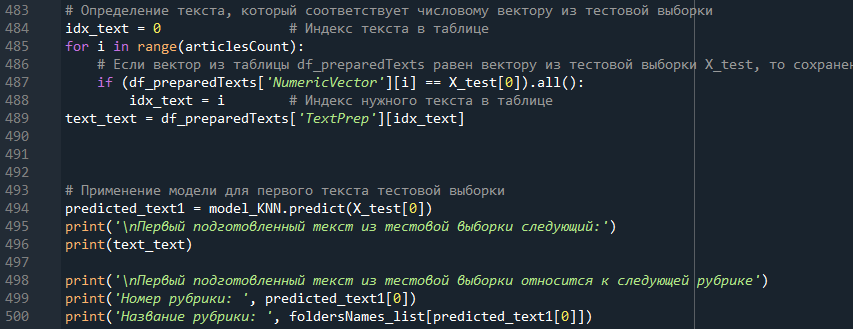




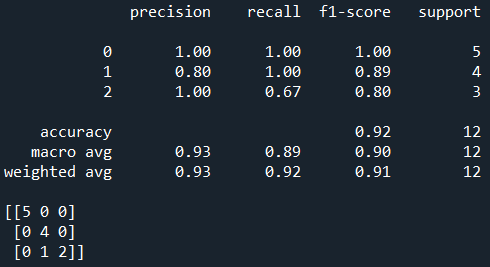


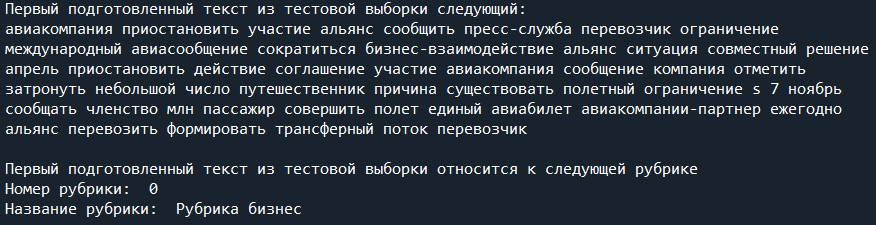
Код в конце программы:



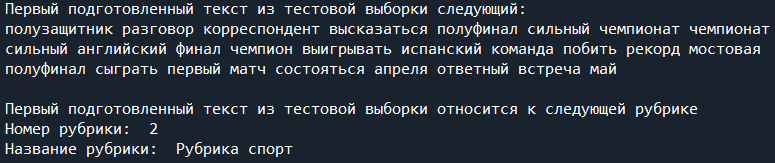


Результат:





Второй запуск программы:



Третий запуск программы:

