**Описание классификатора**

1. Выделены следующие классы:

1. Бывший СССР
2. Интернет и СМИ
3. Культура
4. Мир
5. Наука и техника
6. Путешествия
7. Россия
8. Силовые структуры
9. Среда обитания
10. Экономика.

2. Для обучения под каждый класс выделено 1000 статей, записанных в файле *'dataset.csv'*.

3. Каждый текст проходит предобработку: удаление всех символов кроме русских букв, приведение c к нижнему регистру, лемматизацию (*pymorphy2.MorphAnalyzer*()), удаление стоп-слов (*'stop-ru.txt'*) и стеммизацию (*nltk.stem.SnowballStemmer*()). Обработанные данные записываются в файл *'cleaned\_dataset.csv'*.

4. Модуль *CountVectorizer* позволяет сконвертировать набор текстов в матрицу токенов. *CountVectorizer* по умолчанию сам производит токенизацию и выкидывает слова с длиной меньшей чем два. Модуль используется для построения вектора с количеством слов для каждого предложения (bag of words).

5. Модуль *TfidfTransformer* позволяет вычислить значения IDF (обратная частота документа, вес, показывающий, насколько часто используется слово). Цель использования TF-IDF вместо необработанных частот появления токена в тексте состоит в том, чтобы уменьшить влияние токенов, которые очень часто встречаются в корпусе и, следовательно, менее информативны.

6. Для получения предсказаний с большей точностью, было применено три способа обучения классификатора:

1. *MultinomialNB* – наивный байесовский метод. Точность: 0,8005.
2. *SGDClassifier* – стохастический градиентный спуск. Точность: 0,8315.
3. *LogisticRegression* – логистическая регрессия. Точность: 0,864.

Выбрана последняя модель.

7. Чтобы с цепочкой *'vectorizer => transformer => classifier'* было проще работать, в sklearn есть класс Pipeline, который функционирует как составной классификатор.

8. Модуль *pickle* записывает сериализованный объект обученного классификатора в файл *'lr.pkl'*.

9. Так же реализован метод *predict\_topic* для получения предсказания тематики текста. Метод использует записанный в файл классфикатор.