Задание 1. Определение языка (0.5)

Даны три файла с разными языками в папке с заданием.

1. Откройте файлы.

2. Объедините все записи из файлов в один датасет.

3. Реализуйте свой классификатор или используйте любые вам известные стандартные библиотеки для определения языка каждой записи (sklearn/nltk и другие библиотеки для машинного обучения не использовать).

4. Добейтесь точности классификации 0.85 на KFold с k=10. 5.

* Определите, что это за языки. Опишите, как вы определили язык.

6. Задание присылайте в ipython notebook.

Полезные библиотеки codecs, unicodedata, base64

Задание 2. Определение языка (0.5)

Для передачи разного рода информации внутри текстовых данных (в частности, с помощью электронной почты), а именно: текст на языках, для которых используются кодировки, отличные от ASCII, и нетекстовые данные, такие, как картинки, музыка, фильмы и программы существуют разные алгоритмы кодирования информации.

Вам необходимо реализовать кодирование текстовых последовательностей приведенных в задании 1 следующим образом.

Кодирование (0.2)

Возьмем текст русский текст «АБВГД». В двоичной форме в кодировке Windows-1251 мы получим 5 байтов:

11000000, 11000001, 11000010, 11000011, 11000100, (00000000) — лишний нулевой байт нужен, чтобы общее число бит делилось на 6.

Разделим эти биты на группы по 6:

110000, 001100, 000111, 000010, 110000, 111100, 010000, 000000.

Берем массив символов «ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/» и получившиеся числа переводим в эти символы, используя их, как

индексы массива, получаем «wMHCw8Q». Остается только добавить в конце один символ "=", как указание на один лишний нулевой байт, который мы добавляли на первом шаге и получить окончательный результат:

«АБВГД»: base64 = «wMHCw8Q=».

В общем случае для того, чтобы преобразовать данные, первый байт помещается в самые старшие восемь бит 24-битного буфера, следующий — в средние восемь и третий — в младшие значащие восемь бит. Если кодируется менее чем три байта, то соответствующие биты буфера устанавливаются в ноль. Далее каждые шесть бит буфера, начиная с самых старших, используются как индексы строки́ «ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/» и её символы, на которые указывают индексы, помещаются в выходную строку. Если кодируются только один или два байта, в результате получаются только первые два или три символа строки, а выходная строка дополняется двумя или одним символами «=». Это предотвращает добавление дополнительных битов к восстановленным данным. Процесс повторяется над оставшимися входными данными.

Декодирование (0.2)

С декодированием практически также легко. По сути это обратная операция кодированию. Последовательность символов, полученных при конвертации байт, мы разбиваем на ровные группы по 4. Затем каждый символ в соответствии с алфавитом кодирования мы получаем цифровой порядковый индекс (номер), каждое подобное значения мы конвертируем в двоичную систему (6 бит) и получаем 24 бита, которые делим на уже три части и это будут наши первоначальные байты информации. Повторить до конечного результата.

Протестируйте результат (0.1)

Протестируйте реализацию на 5-10 строках из каждого файла первого задания. Сравните результат с библиотекой base64.