Отчёт об использованных методах

Pekhterev Denis

7 April 2019

1 Introduction

Итак, имеем определённый набор данных, в данной работе использовался набор данных - отзывов на категории товаров, представленных на сайте Amazon в период с мая 1996 по июль 2014 года. Были взяты наборы данных по следующим категориям товаров:

1) Отзывы на товары категории **"Рецензии на книги"**
2) Отзывы на товары категории **"Здоровье и личная гигиена"**
3) Отзывы на товары категории **"Кино и Телевидение"**
4) Отзывы на товары категории **"Электроника"**

Наборы данных были представлены в одинаковом формате json и состояли из следующих колонок:

- reviewerID - будет описание
- asin - будет описание
- reviewerName - будет описание
- helpful - будет описание
- reviewText - будет описание
- overall - будет описание
- summary - будет описание
- unixReviewTime - будет описание
- reviewTime - будет описание

Из этих столбцов для нашей задачи будут интересны только reviewText - сам отзыв, оставленный посетителем сайта и overall - оценка пользователя, которая ставится в соответствии с его отзывом.
Итак, обработав отзывы в формате json и переведя их в формат pandas - DataFrame, который является наиболее удобным для работы с наборами данных, мы получили 3 набора данных. Из-за того, что оперативная память на компьютере ограничена (8gb.), то рассмотрим только часть отзывов - 1

млн. записей.
Для начала, в данном упражнении, переведём оценки из формата 1-5 в бинарный (0/1), для этого воспользуемся следующей логикой:

- textпри overall $\in$(1-2) присвоим значение 0, что будет означать, что отзывы данной категории имеют отрицательную тональность.
- при overall $\in$(3) - не будем присваивать никакого значения, так как для первой задачи мы будем рассматривать задачу бинарной классификации.
- при overall $\in$ (4,5) - присвоим значение 1, так как оставляя отзывы с такими оценками, мы можем быть уверены, что пользователь был удовлетворён своей покупкой, а значит отзывимеет положительную тональность.

Имеем 4 набора данных, состоящих из отзыва пользователя и его оценки, в каждом по миллиону записей. Так как в задачах машинного обучения очень важно иметь сбалансированные выборки, то 1 миллион записей набирался по следующему алгоритму: выбирались 500000 положительных отзывов (оценка: 1) и 500000 отрицательных отзывов (оценка: 0).

Первым этапом при анализе тех или иных текстов методами машинного обучения, текстовые данные требуется перевести в числовой формат.
Для этого существует определённых набор методик, таких как:

- Метод "Мешок слов"(BoW, bag-of-words), является простым алгоритмом, который переводит набор данных в матричный формат, где каждая строка является отдельным предложением или отедельным отзывом, тогда как столбец отвечает за 1 слово в этом предложении/отзыве. Элементами же матрицы являются частоты с которыми данные слова встречаются в предложении/строки
- –я такой пока не описал: TF-IDF
- –и такой пока тоже

При построении числового представления отзывов (векторизации) я использовал самый проверенный метод !!!!! - метод "Мешок слов". Удалив из данных знаки препинания и другие лишние символы (отмечу, что они могут быть исследованы при более глубоком анализе), я воспользовался библиотекой sklearn, а точнее CountVectorizer (внутри задаю binary = True, так как сначала хочется изучить зависимость выходного результата от вхождения определённых слов в предложение), в котором зашита логика метода Мешок слов.
Формальная постановка рассматриваемой задачи: X - множество объектов, в нашем случае это одномерный вектор, который представляет из себя текстовые отзывы покупателей о приобритённом товаре. $x_1, ..., x_l \subset X$ - обучающая выборка. $D_j = 0, 1$ - бинарный признак, который отвечает за тональность отзыва: положительный/отрицательный. $f_j : X \to D_j, j = 1, .., n$

- признаки объектов (от англ. features) Тогда векторное описание объекта имеет следующий вид:

$$F = ||f_j(x_i)||_{l*n} = \begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \vdots & \ddots & \vdots \\ f_1(x_l) & \dots & f_n(x_l) \end{pmatrix}$$

Для решения данной задачи бинарной классификации, будем использовать логистическую регрессию. !тут вставлю картинку с тем, что из себя представляет! Так как наша целевая функция принимает только значения в диапазоне от 0 до 1, то линейную регрессию мы использовать не можем. Как раз для того, чтобы при разных входных данных, мы получали значения в заданном диапазоне, мы будем использовать функцию под названием сигмойда (или логистическую функцию). Она имеет вид $1/(1 + e^{-x})$. Получается, что сначала мы применим линейную регрессию, а выходной результат будем использовать в логистической функции.