Lazy classifier

УМАД

ВШЭ, ФКН

Карасев Олег

Москва, 2021

1. Введение

Задачей данной работы являлось имплементировать алгоритм ленивой классификации, основанный на Формальном Анализе Понятий.

Первая часть моей работы состоит из алгоритма классификации и предобработки данных. Вторая часть – презентация результатов и их анализ.

1. Алгоритм классификации и предобработки данных

­Бэйзлайн данного проекта строится на основе ленивой классификации.

Алгоритм заключается в следующем:

Мы делим входной массив данных на элементы. Необходимо подсчитать, какое количество пересечений признаков входного элемента есть у положительного класса и отрицательного. На основе этих данных мы принимаем решение классификации. Далее мы поэлементно проверяем пересечение признаков входного элемента с элементами массива класса 1. В случае пересечения признаков мы проверяем, входит ли пересечение в какой-либо из элементов массива класса 2. В случае, если не входит – мы добавляем к саппорту класса 1. Если не входит, признак отметается. Для саппорта класса 2 мы повторяем алгоритм. Решение о назначении класса принимается исходя из разности саппорта первого класса и второго (принимается класс, чей саппорт больше).

Усовершенствованный алгоритм классификации заключается в добавлении порогового значения для ложных гипотез. В случае, если пересечение признаков с классом 1 включено в один из элементов класса 2 мы не отметаем признак. Однако, мы подсчитываем такие случае и когда количество фальсификаций превышает пороговое значение – отметаем признак.

Для оценки результатов алгоритмов были использованы четыре метрики и кросвалидация.

1. Презентация результатов и анализ.

Реализация алгоритмов производилась на двух датасетах. Первый датасет – tic-tac-toe.

Метрики для оценки использовались следующие:

1. Accuracy
2. Roc\_Auc
3. Precission
4. Recall

В первой реализации алгоритма на датасете крестики-нолики я получил следующие значения метрик (соответственно):

1) 0.9491329479768786

2) 0.9252195445837688

3) 0.9288327916775688

4) 1.0

Во реализации с пороговым значением 2:

1) 0.976878612716763

2) 0.9663685787710948

3) 0.9665356561514469

4) 1.0

Для сравнения была имплементирована логистическая регрессия.

Полученные метрики:

1. 0.7687861271676301
2. 0.7137464387464387
3. 0.753731343283582
4. 0.9351851851851852

Для проверки алгоритма был использован сторонний [датасет](https://archive.ics.uci.edu/ml/datasets/Teaching+Assistant+Evaluation). Задачей является оценить, к какому классу оценки относится учитель (высокий или низкий класс оценивания). Датасет содержит категориальные, бинарные и действительные признаки. Для бинаризации признаков были написаны функции бинаризации.

В датасете содержится 151 признак и 5 атрибутов.

В датасете содержатся следующие признаки:

1. En\_speaker – говорит ли преподаватель на английском языке или нет,
2. Instructor – категориальный признак,
3. Course – категориальный признак,
4. Semester – бинарный признак, преподает ли преподаватель круглый год или только летом,
5. Target – оценка преподавателя, таргет.

В первой реализации алгоритма на датасете оценки преподавателей я получил следующие значения метрик (соответственно):

1. 0.6242857142857142
2. 0.6698232323232323
3. 0.6518648018648019
4. 0.6888888888888889

В реализации алгоритма с пороговым значением равным 2 на датасете оценки преподавателей я получил следующие значения метрик (соответственно):

1. 0.7519047619047619
2. 0.8016594516594516
3. 0.8405194805194806
4. 0.7277777777777777

Для сравнения была имплементирована логистическая регрессия.

Полученные метрики:

1. 0.7142857142857143
2. 0.6944444444444444
3. 0.7142857142857143
4. 0.5555555555555556