Лабораторная работа № 2 по курсу ИИ

Выполнил студент группы М8О-308Б-17 МАИ *Гринин Вячеслав Витальвич.*

Условие

Необходимо реализовать алгоритмы машинного обучения. Применить данные алгоритмы на наборы данных, подготовленных в первой лабораторной работе. Провести анализ полученных моделей, вычислить метрики классификатора. Произвести тюнинг параметров в случае необходимости. Сравнить полученные результаты с моделями реализованными в scikit-learn. Аналогично построить метрики классификации. Показать, что полученные модели не переобучились. Также необходимо сделать выводы о применимости данных моделей к вашей задаче.

Алгоритмы

1. Логистическая регрессия
2. KNN
3. Дерево решений
4. Случайный лес

Метод решения

Датасеты и постановка задачи

В процессе предыдущей лабораторной было создано два датасета, а также поставлены задачи машинного обучения:

- RGB - содержится информация о цветах. На её основе надо обучить сеть определять яркость - тусклые и яркие цвета.
- Статистика видео на YouTube - содержится **много** различной информации по видео, а также является ли видео популярным или нет. С его помощью надо научить нейросеть определять, является ли ролик популярным или нет.

Результаты

Собственноручные реализации из "грязи и палок" оказывались всегда медленее реализаций из модуля `sklearn`. Не знаю, что там с этими алгоримами делали, но явно что-то страшное. Для сравнения реализация KNN из этого модуля полностью отрабатывала за 3 секунды, в то время как местная реализация работала 3 минуты. И именно из-за этой причины мне часто казалось, что что-то идёт не так. Как итог я тратил в

пустую время, пока искал возможные ошибки (что было катастрофически пробелмно, потому что я не знал, как в `python` производить отладку аля `C++`).

Бонусом идёт то, что точность также была ниже. Однако в этом случае разница не такая катастрофическая. Теперь к самим алгоритмам.

1. Логистическая регрессия - практически быстрее всех. Однако на втором датасете проявила худшую точность. Причиной тому может являться то, что признаки не получилось как-то линейно разделить.

2. KNN - отработала несколько дольше, но взамен выдала великолепную точность. Особенно проблема по скорости чувствуется в местной реализации.

3. Дерево решений - самая быстрая и самая точная рука на диком Западе. Данный алгоритм отлично подходит для данных датасетов.

4. Случайный лес - выдал несколько похуже результаты, чем обычное дерево решений, хоть и идейно случайный лес использует просто несколько деревьев решений.

Выводы

Пришёл к выводу, что писать на Питоне без какой-либо особой подготовки можно, но очень проблематично, потому что большую часть времени тратишь на чтение различной документации.

Что касается самих алгоритмов, каждый из них имеет свои границы применимости, где они будут максимально эффективны. Например, условием хорошей работы логистической регрессии является возможность линейно разделить обучающую выборку.

Отдельно стоит отметить дерево решений. Он является достаточно универсальным, однако в то же время он имеет неплохие шансы к переобучению. Связано это с тем, что в процессе обучения алгоритм подгоняет свои узлы под выборку, включая шум. Из-за этого он будет достаточно плохо воспринимать новую информацию. Это исправимо, если ограничить его глубину. Но тогда появляется некоторая погрешность, что тоже не очень хорошо. Решением этой проблемы является случайный лес - группировка нескольких деревьев решений в один лес.