Отчет по лабораторной работе № 2 по курсу «Машинное обучение»

Студент группы М8О-308б-18 МАИ Игитова Александра

1. Постановка задачи

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 вслучае необходимости.Сравнить полученные результаты с моделями реализованными в scikit-learn. Аналогичнопо строить метрики классификации. Показать, что полученные модели не переобучились. Также необходимо сделать выводы о применимости данных моделей к вашей задаче.

2. Реализация

В качестве набора данных была взята статистика о пульсарах с сервиса Kaggle. Данные разделяются на 2 класса: пульсары и не пульсары. Проверяем точность алгоритмов дерево решений, лоигистическа регрессия и случайный лес реализованных в библиотеке sklearn.

```
Random forest:
              precision    recall

  Non-pulsar       0.91      0.97
      Pulsar       0.96      0.90

    accuracy
   macro avg       0.94      0.93
weighted avg       0.94      0.93

Decision tree:
              precision    recall

  Non-pulsar       0.91      0.86
      Pulsar       0.85      0.90

    accuracy
   macro avg       0.88      0.88
weighted avg       0.88      0.88

Logictic regression:
              precision    recall

  Non-pulsar       0.93      0.94
      Pulsar       0.94      0.92

    accuracy
   macro avg       0.93      0.93
weighted avg       0.93      0.93
```

Алгоритмы на наших данных выдали около 90 процентов точности. Далее мы реализуем свои алгоритмы. Алгоритм дерева решений состоит в том что начальный набор данных разделяется на некоторое количество подмножеств. Каждое подмножество лежит в листе дерева. Внутренние же узлы представляют собой правила, по которым делится набор данных. А именно выбирается один из атрибутов и для него выбирается значение, по которому и будут делиться данные. Само же число выбирается таким образом, чтобы минимизировать индекс Джини. Этот индекс показывает насколько часто случайно выбранный элемент из набора неверно помечается, если он случайным образом помечается согласно распределению меток подмножества. Также чтобы дерево не перобучалось вводятся ограничения на глубину дерева и количество объектов в листьях. Затем реализуем алгоритм случайного леса. Он использует множество простых деревьев решений. Из нашей выборки получаем случайную новую подвыборку и строим дерево по ней. В результате получаем несколько таких деревьев и усредняем результат их предсказания.

```
trol53@trol53-Inspiron-15-3567:~/AI/MLlab2$ python3 random_forest.py
Trees: 1
accuracy :0.875000
precision :0.878114
recall :0.872847
Trees: 5
accuracy :0.912000
precision :0.893653
recall :0.927223
Trees: 10
accuracy :0.916000
precision :0.895435
recall :0.933507
```

Также была реализована логистическая регрессия. Применяем функцию сигмоиды к нашим данным и передаем результат в логистическую функцию потерь. Путем метода градиентного спуска пытаемся минимизировать ошибку.

```
fitted successfully to data
accuracy :0.835000
precision :0.946000
recall :0.774141
```

3. Выводы

В данной лабораторной работе были рассмотрены более сложные и более распространенные алгоритмы классификации. В деревьях решений в качестве минимизации был применен индекс Джини, хотя так же можно было применить формулу энтропии. Реализовывать эти алгоритмы было интереснее чем простейшие knn и Байес, ведь эти алгоритмы действительно применяются на практике для анализа данных довольно часто. Так же точность предсказания наших алгоритмов почти совпадает с таковой у алгоритмов библиотеки sklearn, что говорит о том что реализованы алгоритмы верно.