Московский авиационный институт
(Национальный исследовательский университет)

Факультет: «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»
Дисциплина: «Искусственный интеллект»

Лабораторная работа № 2

Тема: «Машинное обучение»

Студент: Рыженко Иван Александрович

Группа: М80-408Б-18

Преподаватель: Ахмед Самир Халид

Дата:_____

Оценка:_____

Москва, 2021

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 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. Задачи со звездочкой бьются по вариантам. Мой номер по списку: 12.

Модели согласно моему варианту:

1) ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ
2) SVM
3) ДЕРЕВО РЕШЕНИЙ

2. Описание программы

В данной лабораторной работе с помощью языка Python я реализовал три алгоритма машинного обучения.

Логистическая регрессия. Для этой задачи я создал отдельный класс, имеющий методы инициализации модели, ее обучения и тестирования. Данные для обучения я выбрал из прошлой лабораторной работы: отзывы к фильмам и их характер (негативный или позитивный) и также искусственно сгенерированные точки на плоскости, подчиняющиеся линейному разделению, для проверки алгоритма. Модель реализована в соответствии с обычным методическим пособием. Оценка качества происходит с помощью метрики точности и матрицы неточности.

Метод опорных векторов. Аналогично предыдущему заданию, я создал отдельный класс модели и методы для работы с ним. Данные для обучения аналогичные, как и метрика оценки точности. Модель реализована в соответствии с алгоритмом линейной разделимости.

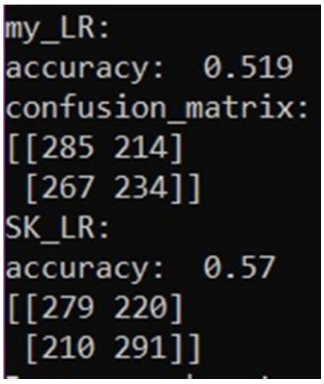
Дерево решений. Для этой задачи я создал множество функций (я не стал работать с классом, так как не совсем понял, как тогда правильно работать с рекурсией), для определения дерева (его обучения) и тестирования. Данные для анализа и метрика идентичны. Для построения

дерева я использовал алгоритм ID3.

3. Результаты работы программы

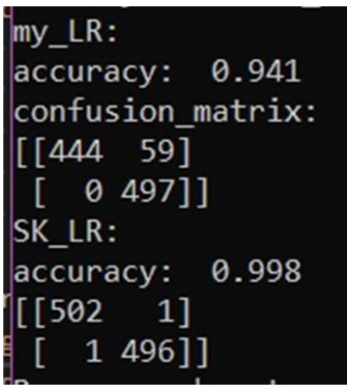
1) Логистическая регрессия.

А) Мои данные



Видно, что точность невысока. Мой алгоритм, как и алгоритм из scikit-learn, справился с задачей не очень хорошо. Это скорее всего происходит из-за того, что данные не подчиняются линейному разделению. Переобучение отсутствует, так как тестирование проводилась на обучаемых данных.

Б) Искусственно сгенерированные данные.



Теперь наблюдается очень неплохая точность. Вывод, мои данные плохо подходят для данного алгоритма. Переобучение отсутствует, так как данные разделяются обычной линией.

2) SVM

А) Мои данные

```
my_SVM:
accuracy:  0.526
confusion_matrix:
[[357 142]
 [332 169]]
SK_SVM:
accuracy:  0.558
[[261 238]
 [204 297]]
```

Ситуация оценивается схоже с предыдущей моделью.

Б) Искусственно сгенерированные данные.

```
my_SVM:
accuracy:  0.995
confusion_matrix:
[[497   5]
 [  0 498]]
SK_SVM:
accuracy:  0.99
[[499   3]
 [  7 491]]
```

Что и требовалось доказать.

3) Дерево решений

А) Мои данные

```
SK SOL_TREE:
accuracy:  0.49
confusion_matrix:
[[46 43]
 [59 52]]
SOL_TREE:
accuracy:  0.515
confusion_matrix:
[[52 37]
 [60 51]]
```

Вновь точность невысока. Как у моего алгоритма, так и у алгоритма scikit-learn.

Б) Искусственно сгенерированные данные.

```
SK SOL_TREE:
accuracy:  0.975
confusion_matrix:
[[96  4]
 [ 1 99]]
SOL_TREE:
accuracy:  0.785
confusion_matrix:
[[79 21]
 [22 78]]
```

Точность теперь получилась выше. То есть первоначальные данные снова оказались плохо операбельны.

Мой алгоритм работает довольно хуже, скорее всего, из-за более старого построения алгоритма в сравнении с его реализацией в scikit-learn.

4. Вывод

В данной лабораторной работе я вновь познакомился с различными алгоритмами машинного обучения. Произведя некоторый анализ, я понял, что данные, которыми я пытаюсь натренировать заданные модели слишком сложны (скорее всего, из-за слишком большой размерности вектора характеристик), и поэтому они требуют более особенных алгоритмов машинного обучения (например, свёрточные нейросети для анализа текста). Я доказал это, проверив работу мною реализованных моделей на искусственно воссозданных данных и сравнив результаты с уже готовой их реализацией в scikit-learn.

В заключение хочу сказать, что полученные знания и опыт в данной лабораторной работе мне, несомненно, пригодятся, как в будущих студенческих проектах, так и далеко за пределами института.