Отчет по лабораторной работе №2
по курсу «Искусственный интеллект»

Выполнил студент группы М8О-308б-16 Никитин Андрей

Тема:
Алгоритмы машинного обучения

Задача:
Требуется реализовать класс на выбранном языке программирования, который реализует один из алгоритмов машинного обучения. Обязательным является наличия в классе двух методов fit, predict. Необходимо проверить работу вашего алгоритма на ваших данных (на таблице и на текстовых данных), произведя необходимую подготовку данных. Также необходимо реализовать алгоритм полиномиальной регрессии, для предсказания значений в таблице. Сравнить результаты с стандартной реализацией sklearn, определить в чем сходство и различие ваших алгоритмов. Замерить время работы алгоритмов.

Вариант: 3) SVM (Метод опорных векторов)

Оборудование студента:
Ноутбук Lenovo ThinkPad 13, процессор Intel® Core™ i5-7200U CPU 1.70 GHz, память 8ГБ, 64-разрядная система.

Программное обеспечение:
ОС Linux Mint 19, Mozilla Firefox 66.0.2

Ход работы:

Основная идея метода заключается в построении гиперплоскости в пространстве признаков. Предполагается, что для более уверенной классификации, необходимо, чтобы данная гиперплоскость была максимально отдалена от всех объектов выборки. Это возможно, при условии, что ближайшие к гиперплоскости объекты разных классов равноудалены от нее. Такую гиперплоскость называют оптимальной.

Уравнение гиперплоскости можно задать с помощью вектора нормали к данной плоскости. Можно показать, что данная задача сводится к решению задачи минимизации нормы этого вектора при заданных ограничения. Если выборка линейно неразделима, то вводятся дополнительные ограничения, которые смягчают требования к разделению объектов (soft-margin svm).

Решая данную задачу методом седловой точки, можно свести данную задачу к эквивалентной задаче, которая выглядит следующим образом:

$$\begin{cases} -\mathbf{L}(\lambda) = -\sum_{i=1}^{n} \lambda_\mathbf{i} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_\mathbf{i} \lambda_\mathbf{j} c_i c_j (\mathbf{x_i} \cdot \mathbf{x_j}) \to \min_\lambda \\ 0 \le \lambda_\mathbf{i} \le \mathbf{C}, \quad 1 \le i \le n \\ \sum_{i=1}^{n} \lambda_\mathbf{i} c_i = 0 \end{cases}$$

Есть множество алгоритмов решающих данную задачу. Один из довольно эффективных sequential minimal optimization. Его суть заключается в выборе двух множителей и оптимизация функции для них. При этом эта данная подзадача решается аналитически. Решив данную задачу для лямбда можно найти вектор весов, который выражается через множители Лагранжа при приравнивании производной функции Лагранжа к нулю.

Описание SMO:
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-98-14.pdf

Исходный код SVM:

https://github.com/AndrewNikitin/ML_Labs/blob/master/%D0%9B%D0%A0%202/SupportVectorMachine.py

Вторая часть задания заключается в построении полиномиальной регрессии. Эта задача легко сводится к линейной регрессии добавлением новых признаков, который соответствуют элементам полинома. Данная задача решается методом наименьших квадратов.

Исходный код PolynomialRegression:

https://github.com/AndrewNikitin/ML_Labs/blob/master/%D0%9B%D0%A0%202/PolynomialRegression.py

Тестирование алгоритмов:

https://github.com/AndrewNikitin/ML_Labs/blob/master/%D0%9B%D0%A0%202/Test.ipynb