РАБОТА С ЛИНЕЙНОЙ КЛАССИФИКАЦИЕЙ

ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Задание основано на применении пакета Python Scikit-learn для изучения подхода градиентного спуска.

Цели практики:

- освоить основы работы с логистической регрессией
- овладеть реализацией градиентного спуска для настройки логистической регрессии
- научиться использованию регуляризации

Начало работы

В этом задании мы предлагаем вам самостоятельно реализовать градиентный спуск.

В качестве метрики качества будем использовать **AUC-ROC** (Area Under ROC-Curve). Она предназначена для алгоритмов бинарной классификации, выдающих оценку принадлежности объекта к одному из классов. По сути, значение этой метрики является агрегацией показателей качества всех алгоритмов, которые можно получить, выбирая какой-либо порог для оценки принадлежности.

В Scikit-Learn метрика **AUC** реализована функцией **sklearn.metrics.roc_auc_score**. В качестве первого аргумента ей передается вектор истинных ответов, в качестве второго — вектор с оценками принадлежности объектов к первому классу.

Задание

1. Загрузите данные из файла **data-logistic.csv.** Это двумерная выборка, целевая переменная на которой принимает значения **-1** или **1**.

2. Убедитесь, что выше выписаны правильные формулы для градиентного спуска. Обратите внимание, что мы используем полноценный градиентный спуск, а не его стохастический вариант! (https://scikit-learn.org/stable/modules/sgd.html)

3. Реализуйте градиентный спуск для обычной и **L2-регуляризованной** (с коэффициентом регуляризации 10) логистической регрессии. Используйте длину шага **k=0.1**. В качестве начального приближения используйте вектор **(0, 0).**

4. Запустите градиентный спуск и доведите до сходимости (евклидово расстояние между векторами весов на соседних итерациях должно быть **не больше 1e-5**). *Рекомендуется ограничить сверху число итераций десятью тысячами*.

5. Какое значение принимает AUC-ROC на обучении без регуляризации и при ее использовании?
Эти величины будут ответом на задание, необходимо привести *два числа через пробел*.
Обратите внимание, что на вход функции **roc_auc_scor**e нужно подавать оценки вероятностей, подсчитанные обученным алгоритмом.
Для этого воспользуйтесь сигмоидной функцией: $a(\chi) = 1/(1 + \exp(-\mathbf{w}_1 \chi_1 \wedge \mathbf{w}_2 \chi_2))$.

6. Поменяйте длину шага. При более длинных шагах будет ли сходиться алгоритм? Как меняется число итераций при уменьшении длины шага?

7. Попробуйте менять начальное приближение. Влияет ли оно на что-нибудь?

Ответ на каждое задание — текстовый файл, содержащий ответ в первой строчке (#.txt). Обратите внимание, что отправляемые файлы не должны содержать перевод строки в конце.

Утонения по выполнению задания:
Если ответом является нецелое число, то целую и дробную часть необходимо разграничивать точкой, например, 0.421. При необходимости округляйте дробную часть до трех знаков.

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

Логистическая регрессия представляет собой одну из форм линейных классификаторов, отличительной особенностью которой является способность оценивать вероятности классов. Это отличает её от большинства других линейных классификаторов, которые обычно могут предоставлять только идентификаторы классов.

В отличие от, например, линейной регрессии, логистическая регрессия использует сложный функционал качества, который не позволяет представить решение в явной форме.

Тем не менее, её параметры можно настраивать с использованием градиентного спуска.

При работе с выборкой, содержащей два признака, мы предполагаем, что ответы принадлежат множеству **{-1, 1}.** Для настройки логистической регрессии решается задача:

$$\frac{1}{\ell}\sum_{i=1}^{\ell}\log\left(1+\exp(-y_i(w_1x_{i1}+w_2x_{i2}))\right)+\frac{1}{2}C\|(w)\|^2\to\min_{w_1,w_2}$$

где χ_{i1} **и** χ_{i2} представляют значения первого и второго признаков соответственно на объекте χ_i. В данном контексте рассматриваются алгоритмы без свободного члена для упрощения процесса работы.

Градиентный шаг для весов включает одновременное обновление весов w_1 **и** w_2 с использованием следующих формул:

$$w_1 := w_1 + k\frac{1}{\ell}\sum_{i=1}^{\ell} y_i x_{i1}\left(1-\frac{1}{1+\exp(-y_i(w_1x_{i1}+w_2x_{i2}))}\right) - kCw_1$$

$$w_2 := w_2 + k\frac{1}{\ell}\sum_{i=1}^{\ell} y_i x_{i2}\left(1-\frac{1}{1+\exp(-y_i(w_1x_{i1}+w_2x_{i2}))}\right) - kCw_2$$

где k – представляет размер шага.

При использовании линейных методов можно столкнуться с проблемами переобучения и низким качеством из-за различных аномалий в данных, таких как *мультиколлинеарность и шум*.

Для предотвращения этих проблем рекомендуется использовать регуляризацию, которая снижает сложность модели и предотвращает переобучение. Сила регуляризации определяется коэффициентом **C** в указанных выше формулах.