РАБОТА С МЕТРИЧЕСКИМИ МЕТОДАМИ НА ПРИМЕРЕ МЕТРИКИ МИНКОВСКОГО

Задание основано на применении пакета Python Scikit-learn для решения задачи регрессии.

Цели практики:

- освоить работу с методом К - ближайших соседей
- научиться подбирать конкретную метрику Минковского для решения задачи.
- овладеть способом выбора лучшей метрики при решении задачи регрессии.

Начало работы

Нам понадобится решать задачу регрессии с помощью метода К - ближайших соседей. Для этой цели воспользуйтесь классом sklearn.neighbors.KNeighborsRegressor.

Метрика задается с помощью параметра metric, нас будет интересовать значение «minkowski».

Задание

Для исследования в данном задании применяется набор данных «Boston», в рамках которого необходимо предсказывать стоимость жилья на основе различных характеристик его расположения, таких как: загрязненность воздуха, близость к дорогам и др.

1. Загрузите выборку Boston с помощью функции sklearn.datasets.load_boston().

   Результатом вызова данной функции является объект, у которого признаки записаны в поле data, а целевой вектор - в поле target.

2. Приведите признаки в выборке к одному масштабу при помощи функции sklearn.preprocessing.scale.

3. Переберите разные варианты параметра метрики p по сетке от 1 до 10 с таким шагом, чтобы всего было протестировано 200 вариантов (используйте функцию numpy.linspace). Используйте KNeighborsRegressor с n_neighbors=5 и weights='distance' - данный параметр добавляет в алгоритм веса, зависящие от расстояния до ближайших соседей.

   В качестве метрики качества используйте среднеквадратичную ошибку (параметр scoring='mean_squared_error' у cross_val_score; при использовании библиотеки scikit-learn версии 18.0.1 и выше необходимо указывать scoring='neg_mean_squared_error'). Качество оценивайте, как и в предыдущем задании, с помощью кросс-валидации по 5 блокам с random_state = 42, не забудьте включить перемешивание выборки (shuffle=True).

4. Определите, при каком p качество на кросс-валидации оказалось оптимальным. Обратите внимание, что cross_val_score возвращает массив показателей качества по блокам; необходимо максимизировать среднее этих показателей, которое является ответом на задачу.

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, 0.4. При необходимости округляйте дробную часть до одного знака.

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

Метрические методы чувствительны к масштабу признаков. Если масштаб одного из признаков существенно превосходит масштабы остальных признаков, то их значения практически не будут влиять на ответы алгоритма. Поэтому важно производить масштабирование признаков. Обычно это делается путем вычитания среднего значения признака и деления на стандартное отклонение.

Главным параметром любого метрического алгоритма является функция расстояния (или метрика), используемая для измерения сходства между объектами. Можно использовать стандартный вариант (например, евклидову метрику), но гораздо более эффективным подходом является подбор метрики под конкретную задачу.

Одним из подходов является использование той же евклидовой метрики, но с весами, где каждой координате ставится в соответствие определенный коэффициент; чем он больше, тем выше вклад признака в итоговое расстояние. Веса настраиваются с целью оптимизации качества на отложенной выборке. Другой подход, о котором и пойдет речь в данном задании - выбор метрики из некоторого класса метрик. Мы возьмем за основу метрику Минковского:

$$\rho_p(x, z) = (\sum_{j=1}^{n} |x_j - z_j|^p)^{\frac{1}{p}}$$

Параметр метрики Минковского задается с помощью параметра p данного класса.

Дополнительные сведения по признакам приведены в статье: https://archive.ics.uci.edu/ml/datasets/Housing