РАБОТА С МЕТРИЧЕСКИМИ МЕТОДАМИ НА ПРИМЕРЕ К-БЛИЖАЙШИХ СОСЕДЕЙ (KNN)

Задание основано на применении пакета Python Scikit-learn для метода kNN.

Цели практики:

- освоить работу с методом К - ближайших соседей
- уметь выбирать в нем параметр К
- овладеть подходами подготовки данных для применения данного метода.

Начало работы

В этом задании вам нужно подобрать оптимальное значение k для алгоритма kNN. Будем использовать набор данных о винах «Wine», которые представляют собой данные, полученные в результате большого количества анализов трех различных сортов вина, произведенных в одном регионе, где требуется предсказать сорт винограда, из которого изготовлено вино, используя результаты химических анализов.
В файле «wine.data» каждая строка представляет образец вина.
Всего 178 образцов распределенных на:

– 14 столбцов, из которых первый столбец является атрибутом флага класса. Имеются категориальные признаки - три категории, которые помечены, соответственно, как «1», «2» и «3».

– следующие 13 столбцов представляют собой примерные значения соответствующих атрибутов каждой выборки. Среди них первая категория включает 59 образцов, вторая категория - 71 образец, а третья категория - 48 образцов.

Задание

Загрузите датасет «wine.data» и выполните следующие шаги:

1. Извлеките из данных признаки и классы. Класс записан в первом столбце (три варианта), признаки – в столбцах со второго по последний. Для уточнения информации о сути признаков, можно ознакомиться с файлом «wine.names», в приложении к заданию.

2. Проведите оценку качества методом кросс-валидации по 5 блокам (5-fold). Для этого создайте генератор разбиений, который перемешивает выборку перед формированием блоков (shuffle=True). Для воспроизводимости результата, создавайте генератор KFold с фиксированным параметром random_state=42. В качестве меры качества используйте долю верных ответов (accuracy).

3. Найдите точность классификации на кросс-валидации для метода k ближайших соседей (sklearn.neighbors.KNeighborsClassifier), при k от 1 до 50. При каком k получилось оптимальное качество? Чему оно равно (число в интервале от 0 до 1)?

Данные результаты являются ответами на вопросы 1 и 2.

4. Произведите масштабирование признаков с помощью функции sklearn.preprocessing.scale. Снова найдите оптимальное k на кроссвалидации.

5. Ответьте и приведите ответ на вопросы 3 и 4:

какое значение k получилось оптимальным после приведения признаков к одному масштабу;

как изменилось значение качества.

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, 0.5. При необходимости округляйте дробную часть до

двух знаков.

Вам понадобится производить кросс-валидацию по блокам.
Это делается в два этапа:

1. Создается генератор разбиений sklearn.model_selection.KFold, который задает набор разбиений на обучение и валидацию. Число блоков в кросс-валидации определяется параметром n_folds. Обратите внимание, что порядок следования объектов в выборке может быть неслучайным, это может привести к смещенности кроссвалидационной оценки. Чтобы устранить такой эффект, объекты выборки случайно перемешивают перед разбиением на блоки. Для перемешивания достаточно передать генератору KFold параметр shuffle=True. 2.
2. Вычислить ошибку на всех разбиениях можно при помощи функции sklearn.model_selection.cross_val_score. В качестве параметра estimators передается классификатор, в качестве параметра cv — генератор разбиений с предыдущего шага. С помощью параметра scoring можно задавать меру качества, по умолчанию в задачах классификации используется доля верных ответов (accuracy). Результатом является массив, значения которого нужно усреднить. Приведение признаков к одному масштабу можно делать с помощью функции sklearn.preprocessing.scale, которой на вход необходимо подать матрицу признаков и получить масштабированную матрицу, в которой каждый столбец имеет нулевое среднее значение и единичное стандартное отклонение.

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

Метрические методы основаны на гипотезе компактности, суть которой состоит в том, что объекты с похожими признаковыми описаниями имеют похожие значения целевой переменной. Если эта гипотеза верна, то строить прогноз для нового объекта можно на основе близких к нему объектов из обучающей выборки — например, путем усреднения их ответов (для регрессии) или путем выбора наиболее популярного среди них класса (для классификации). Методы такого типа и называются метрическими.

Они имеют несколько особенностей:

- процедура обучения, по сути, отсутствует — достаточно лишь запомнить все объекты обучающей выборки;
- можно использовать метрику, учитывающую особенности конкретного набора данных — например, наличие категориальных (номинальных) признаков;
- при правильном выборе метрики и достаточном размере обучающей выборки метрические алгоритмы показывают качество, близкое к оптимальному;

Метрические методы чувствительны к масштабу признаков — так, 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.