Azure ML Studio
Метод k-ближайших соседей

Подготовка данных

Для реализации алгоритма k-ближайших соседей рекомендуем заранее подготовить данные, оставив только колонки (столбцы) с предикторами и откликом. Пусть полученный тренировочный набор данных состоит из колонок `X, Y` – предикторы и `Class` – отклика:

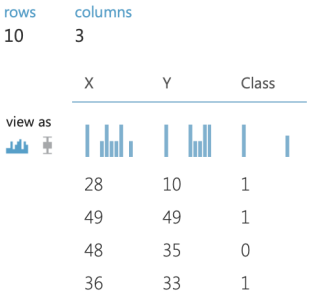

Рис. 1: Тренировочный набор данных.

Для дальнейшего удобства назовем столбцы привычным для нас образом: предикторам будут отвечать столбцы с именами $X_1, X_2, \ldots, X_p$, а отклику - столбец с именем Y. Для этого воспользуемся блоком `Edit Metadata`, на вход которого подаются данные, а в параметрах задаются новые названия колонок. Отметим, что названия указываются через запятую и соответствуют очередности колонок, выбранных в пункте `Selected columns` (на рисунке это колонки `X,Y,Class` переименованные в `X1,X2,Y`):

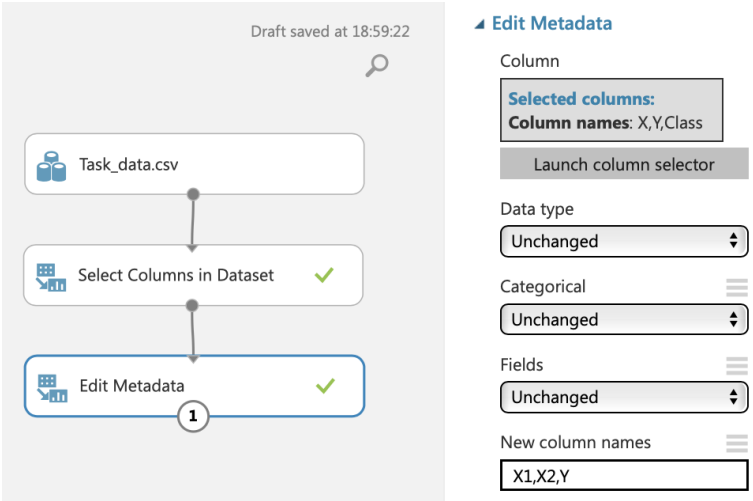

Рис. 2: Параметры блока `Edit Metadata`.

Классификация

Как известно, алгоритм k-ближайших соседей не требует обучения, и мы можем сразу перейти к классификации нового объекта.

Для начала научимся вычислять расстояние между новым объектом и объектами из тренировочного набора данных. Сделать это можно с помощью запросов на языке `SQLite` и блока `Apply SQL Transformation`.

Пусть требуется провести классификацию объекта x', имеющего предикторы $(x'_1, x'_2, ..., x'_p)$. Тогда для вычисления евклидова расстояния

$$d_E(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \ldots + (x_p - x'_p)^2},$$

от каждого объекта тренировочных данных до данного, можно использовать следующий запрос:

```
Select SQRT(power(x_1-x_1',2)+...+power(x_p-x_p',2)) from t1;
```

Возвращаясь к данным из примера, расстояние от объекта $(5, -4)$ до всех объектов из тестового набора данных может быть найдено с помощью запроса:

```
Select SQRT(power(X1-5,2)+power(X2+4,2)) from t1;
```

На выходе блока `Apply SQL Transformation` получаем результаты, представленные на рисунке:

Рис. 3: Евклидово расстояние до всех объектов.

Как видим, в названии столбца отображается текст запроса. Если столбец нужно переименовать, нужно переименовать, то запрос следует дополнить ключевым словом `AS` и новым названием:

```
Select SQRT(power(X1-5, 2)+power(X2+4, 2)) AS Dist from t1;
```

Кроме того, в запросе можно указать названия столбцов тренировочного набора данных, например чтобы отобразить классы объектов, или их названия и т.д. Для этого названия столбцов указываем через запятую:

```
Select SQRT(power(X1-5, 2)+power(X2+4, 2)) AS Dist, Y from t1;
```

Также важно отметить, что порядок строк не меняется и соответсвует тренировочному набору данных.

Если нужно выводить строки в определенном порядке, то это можно сделать при помощи ключевого слова `ORDER BY`. После этого слова указывают столбец или столбцы, которые являются критерием ранжирования. Например, отсортируем данные по значениям нового столбца `Dist`:

```
Select SQRT(power(X1-5, 2)+power(X2+4, 2)) AS Dist, Y from t1
ORDER BY Dist;
```

По умолчанию сортировка выполняется по возрастанию, для сортировки по убыванию используют ключевое слово `DESC`.

Остается ограничить число записей с помощью ключевого слова `LIMIT`. Например, если классификация нового объекта выполняется при $k = 3$, запрос может быть составлен следующим образом:

```
Select SQRT(power(X1-5, 2)+power(X2+4, 2)) AS Dist, Y from t1
ORDER BY Dist LIMIT 3;
```

Результата запроса приведен на рисунке:

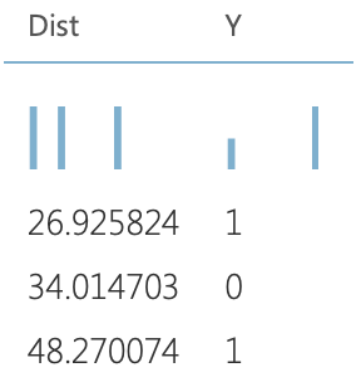

Рис. 4: Расстояния и классы для трех ближайших соседей.

Очевидно, что новый объект $(5, -4)$ будет отнесен к классу 1, так как ближайших представителей именно этого класса оказалось большинство.

Обратите внимание, что возможна ситуация, когда есть объекты с одинаковым расстоянием. В таком случае ключевое слово `LIMIT` следует использовать с осторожностью, так как значения одинаковых расстояний могут попасть на границу разделения данных. Рекомендуем предварительно сравнить число строк `rows` и число уникальных значений в столбце `Dist` – параметр `Unique Values`

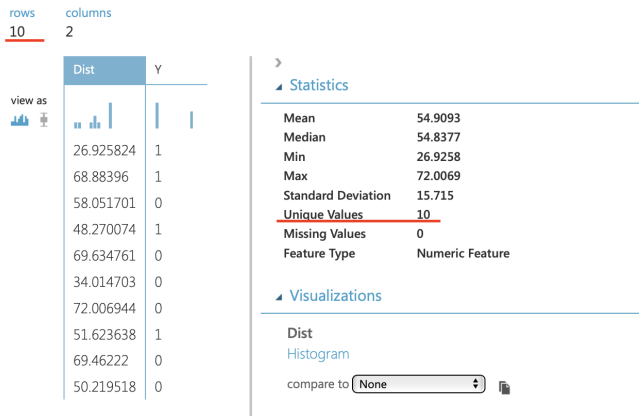

Рис. 5: Сравнение числа уникальных значений и строк.

Наконец, не всегда удается охватить взглядом всех соседей. В случае, когда их много, визуально оценить результаты не представляется возможным. Приведем еще один запрос, который группирует результаты последнего по классам (т.е. столбцу `Y`):

```
select Y, COUNT(*) as Votes from t1
GROUP BY Y ORDER BY Votes DESC;
```

Как видно из рисунка ниже, используется еще один блок `Apply SQL Transformation` (таким образом реализуется вложенность запросов).

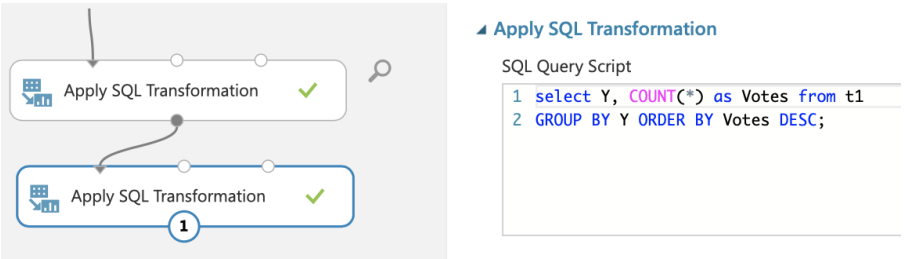

Рис. 6: Евклидово расстояние до всех объектов.

Результат голосования очевиден: в первой строке (с наибольшим количеством голосов) находится класс-победитель:

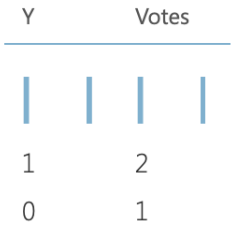

Рис. 7: Группировка результатов голосования.

Итак, объект $(5, -4)$ будет отнесен к классу `1` при $k = 3$.

Прочие метрики

В алгоритме k-ближайших соседей могут использоваться расстояния, отличные от евклидового. Приведем примеры как самих расстояний, так и запросов, их вычисляющих.

Пусть требуется провести классификацию объекта x', имеющего предикторы $(x'_1, x'_2, ..., x'_p)$. Тогда для вычисления манхэттенского расстояния или расстояния городских кварталов

$$d_1(x, x') = \sum_{i=1}^{p} |x_i - x'_i| .$$

используйте функцию `abs()`, которая вычисляет модуль, в следующем запросе:

```
Select abs(x_1-x_1')+...+abs(x_p-x_p') from t1;
```

Если необходимо вычислить расстояние Чебышёва

$$d_\infty(x, x') = \max_{1,...,p} |x_i - x'_i| .$$

используйте функцию `max()`, которая возвращает наибольший из элементов, в следующем запросе::

```
Select max(abs(x_1-x_1'),...,abs(x_p-x_p')) from t1;
```