



Workshop 18

settembre

Il dataset Iris

Vogliamo costruire un'intelligenza artificiale da allenare su uno storico di classificazioni già effettuate in passato. Lo storico è nella tabella *dbo.IrisTraining*. Non considerare la colonna *bias*.

L'IA dovrà predire con la massima accuratezza possibile le specie degli iris raccolti in futuro.

```
acknowledgements
```

```
Fisher,R. A.. (1988). Iris. UCI Machine Learning Repository. https://doi.org/10.24432/C56C76.
```

```
sepal_length;sepal_width;petal_length;petal_width;class
```

```
4;3.5;1.4;0.2;Iris setosa
```

```
4.2;3.0;1.4;0.2;Iris setosa
```

```
4.7;3.2;1.3;0.2;Iris setosa
```

```
4.6;3.1;1.5;0.2;Iris setosa
```

```
7.0;3.2;4.7;1.4;Iris versicolor
```

```
6.4;3.2;4.5;1.5;Iris versicolor
```

```
5.7;2.8;4.1;1.3;Iris versicolor
```

```
6.3;3.3;6.0;2.5;Iris virginica
```

```
5.8;2.7;5.1;1.9;Iris virginica
```

```
7.1;3.0;5.9;2.1;Iris virginica
```

Premessa: formula matematica della distanza

Sia $v = (v_1, v_2, v_3, v_4)$ un vettore numerico (come il contenuto delle quattro colonne di una particolare riga della tabella `dbo.IrisTraining`)

Sia $w = (w_1, w_2, w_3, w_4)$ un secondo vettore numerico

La distanza tra v e w è data dalla formula

radice quadrata di

$$(v_1 - w_1)^2 + (v_2 - w_2)^2 + (v_3 - w_3)^2 + (v_4 - w_4)^2$$

Algoritmo 1-Neighbour – parte 1

Prendiamo il nuovo iris nella tabella *dbo.NuovoIris* (di cui facciamo finta di non conoscere la classe).

Nuovo iris

sepal_length	sepal_width	petal_length	petal_width	class
4,2	3,1	1,6	0,4	?

Algoritmo 1-Neighbour – parte 2

Calcoliamo la distanza da tutti gli iris presenti nello storico e già classificati.

sepal_length	sepal_width	petal_length	petal_width	class	distanze
4	3,5	1,4	0,2	Iris setosa	0,529150262
4,2	3	1,4	0,2	Iris setosa	0,3
4,7	3,2	1,3	0,2	Iris setosa	0,6244998
7	3,2	4,7	1,4	Iris versicolor	4,296510212
6,4	3,2	4,5	1,5	Iris versicolor	3,80394532
6,9	3,1	4,9	1,5	Iris versicolor	4,403407771
6,3	3,3	6	2,5	Iris virginica	5,312249994
5,8	2,7	5,1	1,9	Iris virginica	4,149698784
7,1	3	5,9	2,1	Iris virginica	5,458937626

Algoritmo 1-Neighbour – parte 1

Consideriamo l'iris dello storico con distanza minore. Quella sarà la nostra previsione.

sepal_length	sepal_width	petal_length	petal_width	class	distanze
4	3,5	1,4	0,2	Iris setosa	0,529150262
4,2	3	1,4	0,2	Iris setosa	0,3
4,7	3,2	1,3	0,2	Iris setosa	0,6244998
7	3,2	4,7	1,4	Iris versicolor	4,296510212
6,4	3,2	4,5	1,5	Iris versicolor	3,80394532
6,9	3,1	4,9	1,5	Iris versicolor	4,403407771
6,3	3,3	6	2,5	Iris virginica	5,312249994
5,8	2,7	5,1	1,9	Iris virginica	4,149698784
7,1	3	5,9	2,1	Iris virginica	5,458937626

Algoritmo 5-Neighbours – parte 1

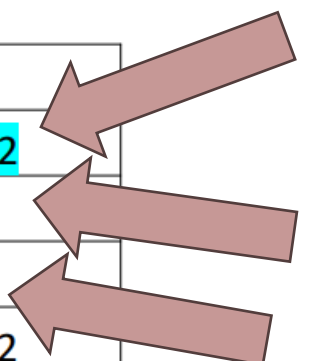
Consideriamo l'iris dello storico con distanza minore. Quella sarà la nostra previsione.

sepal_length	sepal_width	petal_length	petal_width	class	distanze
4	3,5	1,4	0,2	Iris setosa	0,529150262
4,2	3	1,4	0,2	Iris setosa	0,3
4,7	3,2	1,3	0,2	Iris setosa	0,6244998
7	3,2	4,7	1,4	Iris versicolor	4,296510212
6,4	3,2	4,5	1,5	Iris versicolor	3,80394532
6,9	3,1	4,9	1,5	Iris versicolor	4,403407771
6,3	3,3	6	2,5	Iris virginica	5,312249994
5,8	2,7	5,1	1,9	Iris virginica	4,149698784
7,1	3	5,9	2,1	Iris virginica	5,458937626

Algoritmo 5-Neighbours – parte 2

Tra i cinque iris più vicini abbiamo 3 Iris setosa, 1 Iris virginica e 1 Iris versicolor. L'algoritmo predice dunque Iris setosa

sepal_length	sepal_width	petal_length	petal_width	class	distanze
4	3,5	1,4	0,2	Iris setosa	0,529150262
4,2	3	1,4	0,2	Iris setosa	0,3
4,7	3,2	1,3	0,2	Iris setosa	0,6244998
7	3,2	4,7	1,4	Iris versicolor	4,296510212
6,4	3,2	4,5	1,5	Iris versicolor	3,80394532
6,9	3,1	4,9	1,5	Iris versicolor	4,403407771
6,3	3,3	6	2,5	Iris virginica	5,312249994
5,8	2,7	5,1	1,9	Iris virginica	4,149698784
7,1	3	5,9	2,1	Iris virginica	5,458937626



Generalizziamo a un dataset di Test

Vogliamo calcolare ora la predizione per tutte le 30 righe della tabella *dbo.IrisTest*

Generalizza prima la versione 1-neighbour e poi la versione 5-neighbours.

Quale tecnica vuoi usare per lavorare su tutte le 30 righe?

Un'idea potrebbe essere usare un cursore, ma non è la più efficiente!

Traccia soluzione

1) Effettuare una CROSS JOIN tra le due tabelle in modo da creare tutte le combinazioni possibili

2) Inserire nella SELECT il calcolo della distanza. Utilizzare le funzione SQRT e POWER.

Consolidiamo questo step in una CTE o in una Tabella temporanea?

3) Utilizziamo la Window Function RANK() per calcolare per ogni riga di test le 5 righe di training più vicine. Per far ciò aggiungiamo prima la RANK nella SELECT di una CTE e poi filtramo su questa nuova colonna (≤ 5)

Traccia soluzione

- 4) Raggruppiamo i dati per Id di test e predizione. Contiamo dunque quante righe sono presenti per ogni combinazione
- 5) Utilizziamo nuovamente la window function RANK() per calcolare per ogni ID di Test la predizione più ricorrente
- 6) Verifichiamo quante predizioni sono effettivamente uguali alla colonna class originale.