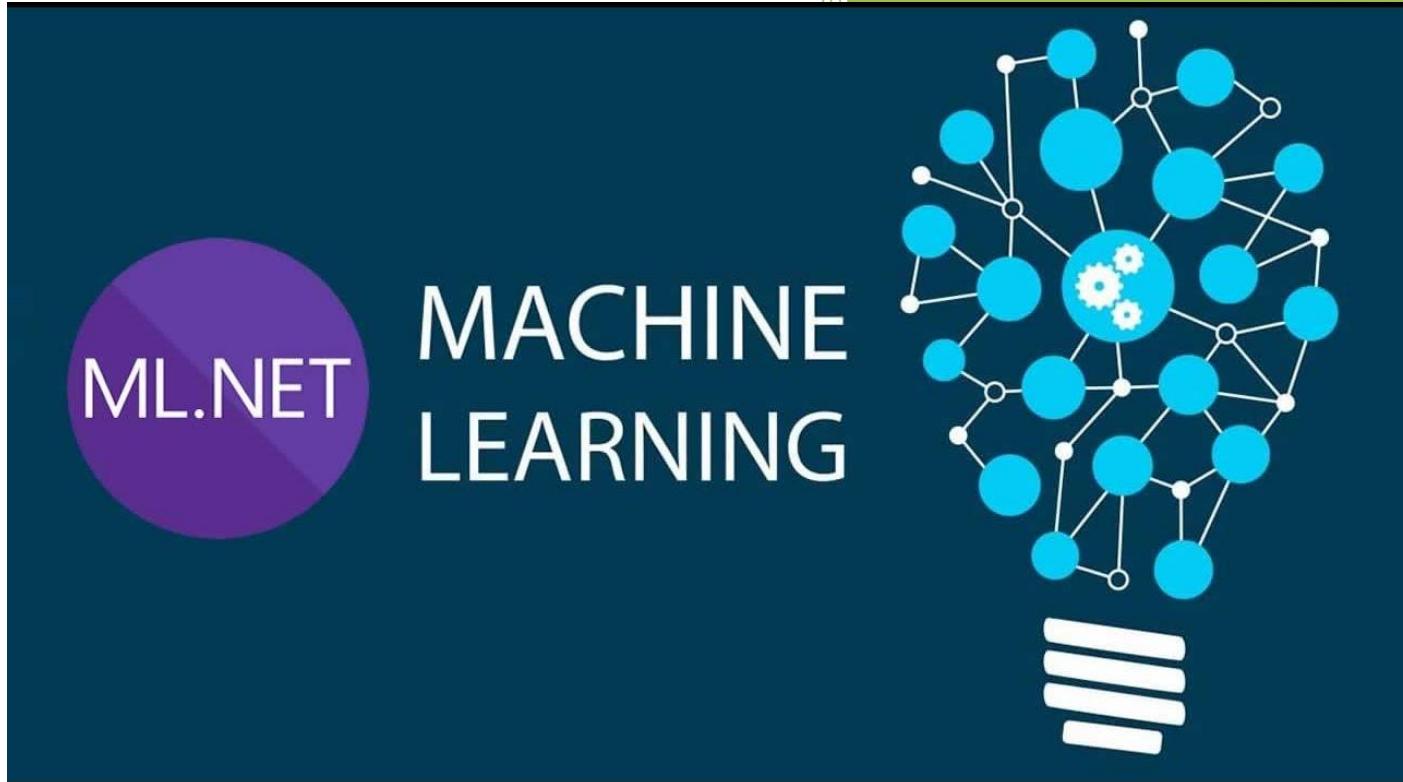


LonGreat ML.NET

教學範例

Community V1.0.0



目 錄

內容

分析情感(二元分類).....	4
建立專案.....	4
準備資料.....	4
載入資料.....	5
資料轉換.....	8
新增資料轉換.....	8
結果檢視.....	8
特徵與標籤.....	9
訓練模型.....	9
訓練器.....	9
執行.....	10
評估.....	11
預測.....	12
紐約市的計程車費用預測價格(迴歸模型).....	13
建立專案.....	13
準備並瞭解資料.....	13
載入資料.....	14
資料轉換.....	17
新增資料轉換.....	17
結果檢視.....	19
特徵與標籤.....	20
訓練模型.....	21
訓練器.....	21
執行.....	21
評估.....	22
鳶尾花花卉分類.....	24
瞭解問題.....	24
選取適當的機器學習工作.....	24
準備資料.....	24
鳶尾花(多元分類).....	24
建立專案.....	24
載入資料.....	25

資料轉換.....	28
特徵與標籤.....	29
訓練模型.....	29
評估.....	30
鳶尾花(群集).....	32
建立專案.....	32
載入資料.....	32
特徵與標籤.....	35
訓練模型.....	35
評估.....	36
推薦的電影(矩陣分解模型).....	38
選取適當的機器學習工作.....	38
建立專案.....	38
載入您的資料.....	39
資料轉換.....	43
新增資料轉換.....	43
結果檢視.....	43
特徵與標籤.....	44
訓練模型.....	45
訓練器.....	45
執行.....	45
其他建議演算法.....	47
評估.....	48
花瓣識(影像分類).....	49
影像分類傳輸學習範例概觀.....	49
瞭解問題.....	49
ML.NET 影像分類 API.....	49
什麼是傳輸學習？.....	49
訓練程式.....	50
瓶頸階段.....	50
訓練階段.....	50
瞭解預先定型的模型.....	51
準備並瞭解資料.....	51
載入資料.....	52
資料轉換.....	55
隨機資料列.....	55
結果檢視.....	56
特徵與標籤.....	57
訓練模型.....	57

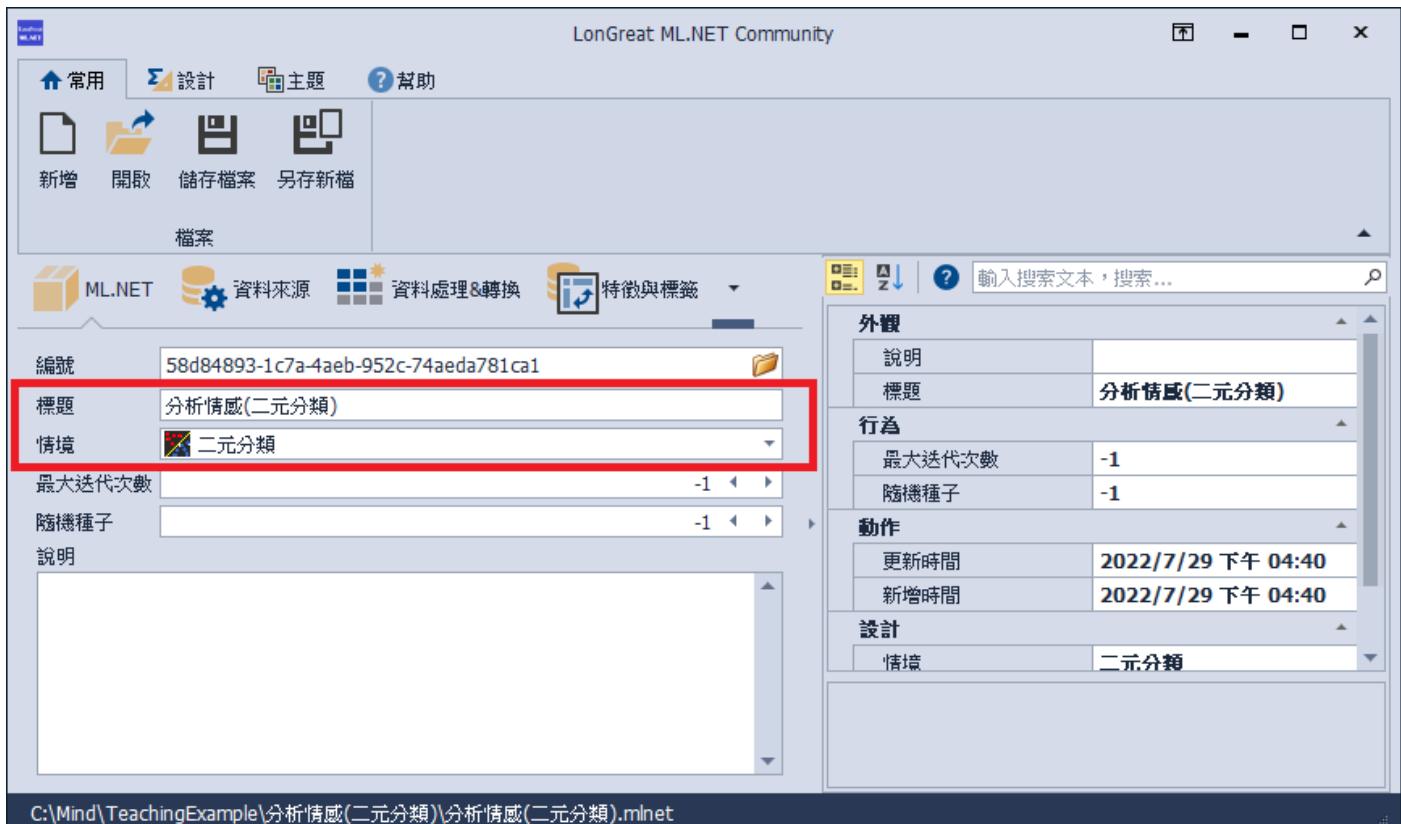
LonGreat ML.NET Community

訓練器.....	57
執行.....	58
評估.....	59
預測.....	60
相關連結.....	60

分析情感(二元分類)

建立專案

1. 輸入標題 => 分析情感(二元分類)
2. 選擇情境 => 二元分類



準備資料

輸入資料集類別 `SentimentData` 具有使用者評論 (`SentimentText`) 的 `string`，以及代表情感的 `bool` (`Sentiment`) 值 1 (正面) 或 0 (負面)。其描述每個欄位的資料檔案順序。此外，`Sentiment` 屬性具有 [ColumnName](#) 屬性來將它指定為 `Label` 欄位。下列範例檔案沒有標頭資料列，且看起來像這樣：

<code>SentimentText</code>	<code>Label</code>
女服務生的服務速度有點慢。	0
不夠酥脆。	0

哇。。。喜歡這個位置。	1
服務很迅速。	1

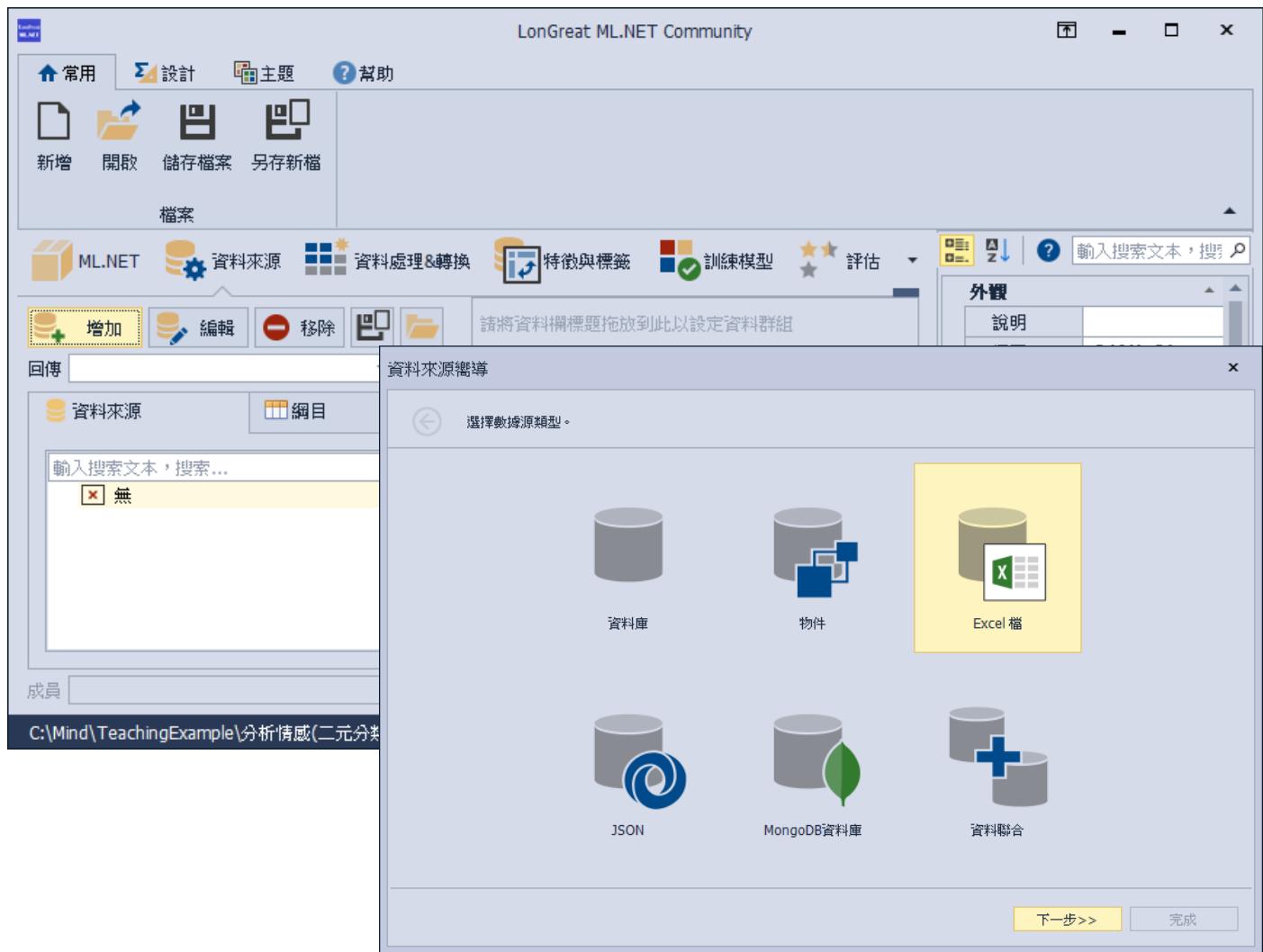
SentimentPrediction 是在模型定型後所使用的預測類別。它繼承自 SentimentData，以便輸入 SentimentText 可以和輸出預測一起顯示。Prediction 布林值是在提供新輸入 SentimentText 時，模型預測的值。

輸出類別 SentimentPrediction 包含模型計算的兩個其他屬性：Score – 模型計算的原始分數，和 Probability – 針對文字具有正面情感之可能性所校正的分數。

本教學課程中最重要的屬性是 Prediction。

載入資料

資料來源->增加->Excel 檔



選取 yelp_labelled.xlsx



This screenshot shows the second and third steps of the 'Data Source Wizard'.
Step 2 (Left): Title bar '資料來源嚮導' (Data Source Wizard). Subtitle '指定導入設置' (Specify import settings). It lists several import options:

- 使用 first 行的值為欄位名稱 (Use the value of the first row as column names)
- Skip empty rows
- Skip hidden rows
- Skip hidden columns

Step 3 (Right): Title bar '資料來源嚮導' (Data Source Wizard). Subtitle '選擇所需的工作表、表或定義的名稱，指的指定範圍內' (Select the required worksheet, table, or defined name within the specified range). A list box shows 'Sheet' selected. At the bottom right are '下一步>>' (Next >) and '完成' (Finish) buttons.

按完成



檢視資料

Sentiment Text	Label
Wow... Loved this place.	1
Crust is not good.	0
Not tasty and the texture was just nasty.	0
Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.	1
The selection on the menu was great and so were the prices.	1
Now I am getting angry and I want my damn pho.	0
Honesty it didn't taste THAT fresh.)	0
The potatoes were like rubber and you could tell they had been made up ahead of time being kept und...	0
The fries were great too.	1
A great touch.	1
Service was very prompt.	1
Would not go back.	0
The cashier had no care what so ever on what I had to say it still ended up being wayyy overpriced.	0
I tried the Cape Cod ravioli, chicken,with cranberry...mmmm!	1
I was disgusted because I was pretty sure that was human hair.	0
I was shocked because no signs indicate cash only.	0
Highly recommended.	1
Waitress was a little slow in service.	0
This place is not worth your time, let alone Vegas.	0
did not like at all.	0
The Burritos Blah!	0

765

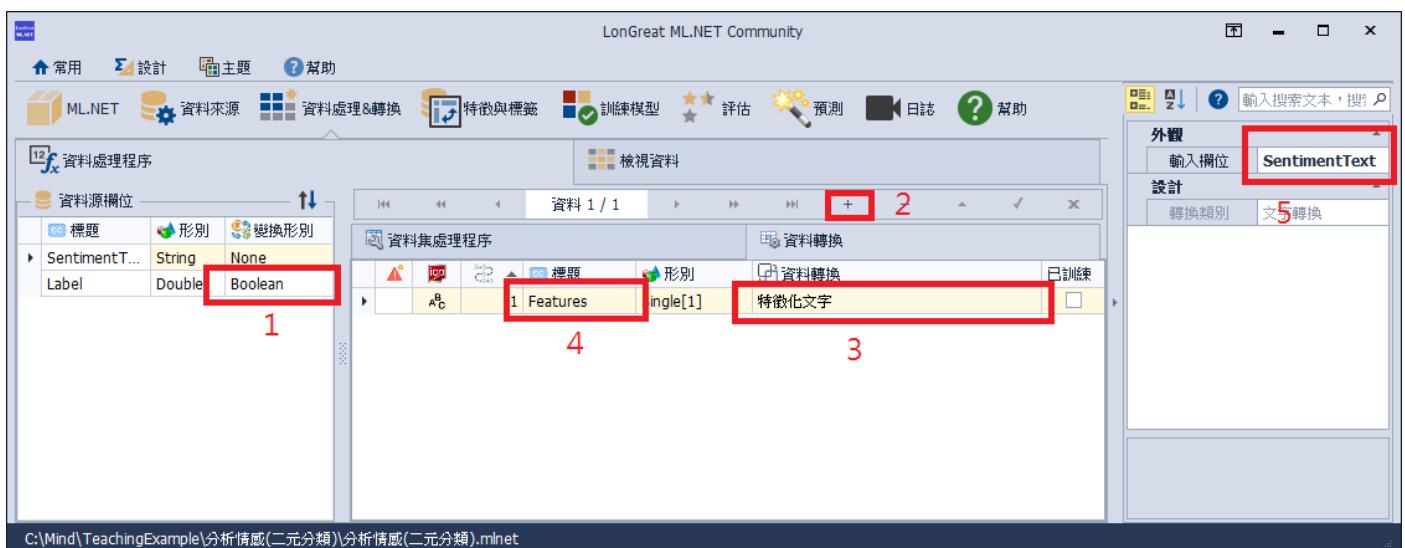
C:\Mind\TeachingExample\分析情感(二元分類)\分析情感(二元分類).mlnet

資料轉換

新增資料轉換

將欄位 SentimentText 特徵化，使用特徵化文字(FeaturizeText)函式

1. Label 變換形別=>Boolean
2. +新增資料轉換
3. 選擇資料轉換函式=>特徵化文字(FeaturizeText)
4. 定義新欄位標題名稱=>Featurizes
5. 選擇輸入欄位=>SentimentText



結果檢視

1. 至檢視資料頁籤
2. 選擇顯示筆數
3. 按充滿資料

SentimentText	Label	Features
女服務生的服務速度有點慢。	0	[0.76, 0.65, 0.44, ...]
不夠酥脆。	0	[0.98, 0.43, 0.54, ...]
哇。。。喜歡這個位置。	1	[0.35, 0.73, 0.46, ...]
服務很迅速。	1	[0.39, 0, 0.75, ...]

LonGreat ML.NET Community

D:\TeachingExample\分析情感(二元分類).mlnet

特徵與標籤

拖拉 Features=>特徵

拖拉 Label => 標籤

C:\Mind\TeachingExample\分析情感(二元分類)\分析情感(二元分類).mlnet

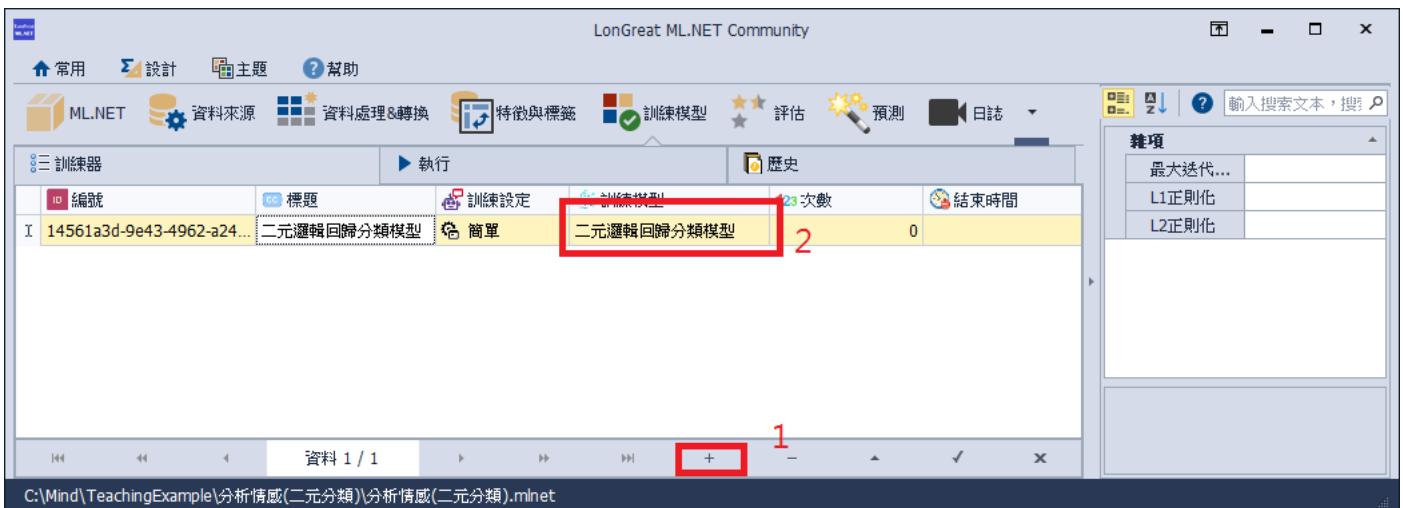
訓練模型

訓練器

按+新增

選擇二元邏輯回歸分類模型 (SdcaLogisticRegressionBinary)

LonGreat ML.NET Community

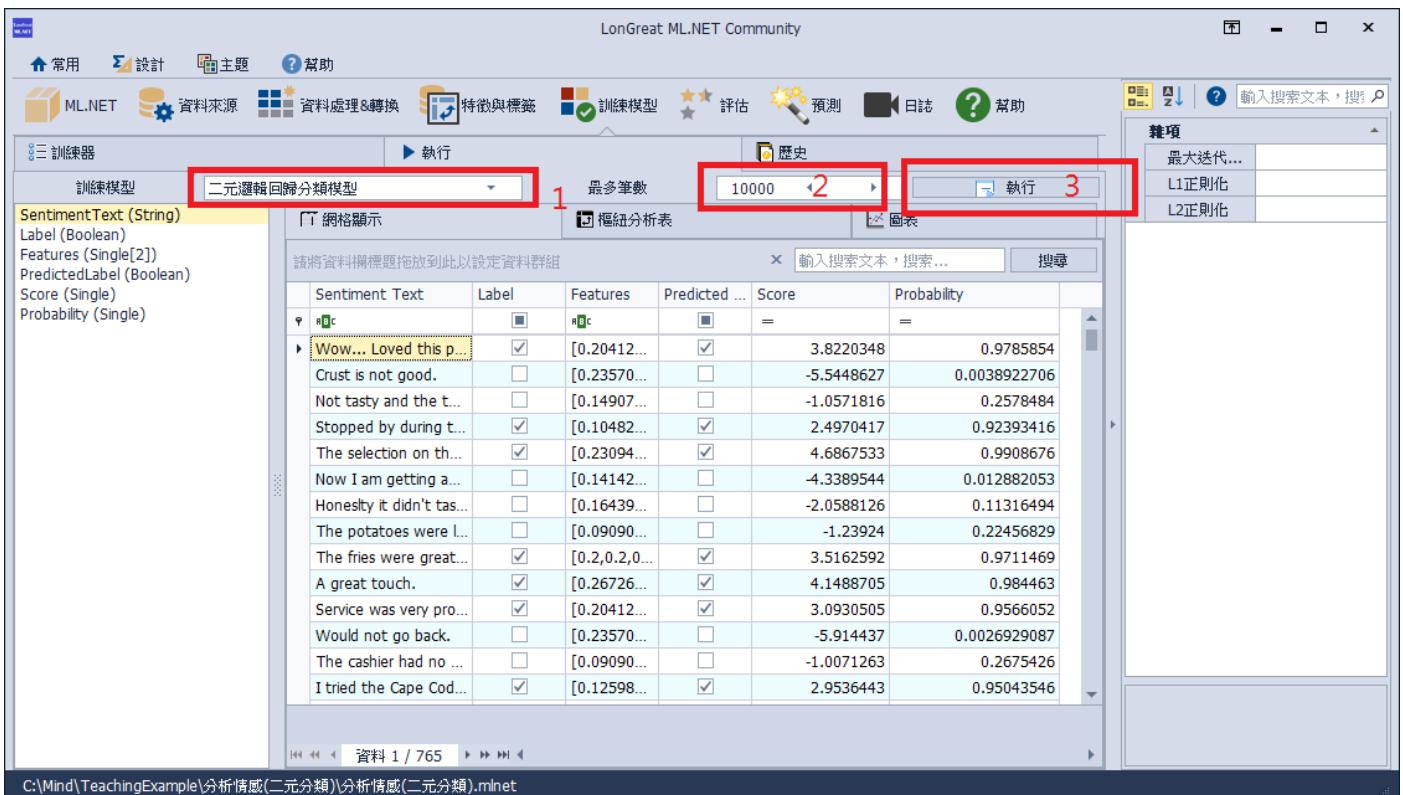


執行

1. 選擇訓練模型
2. 要顯示最多筆數
3. 執行

結果多三個欄位

- 預測標籤 PredictedLabel
- 或然率 Probability
- 分數 Score



評估

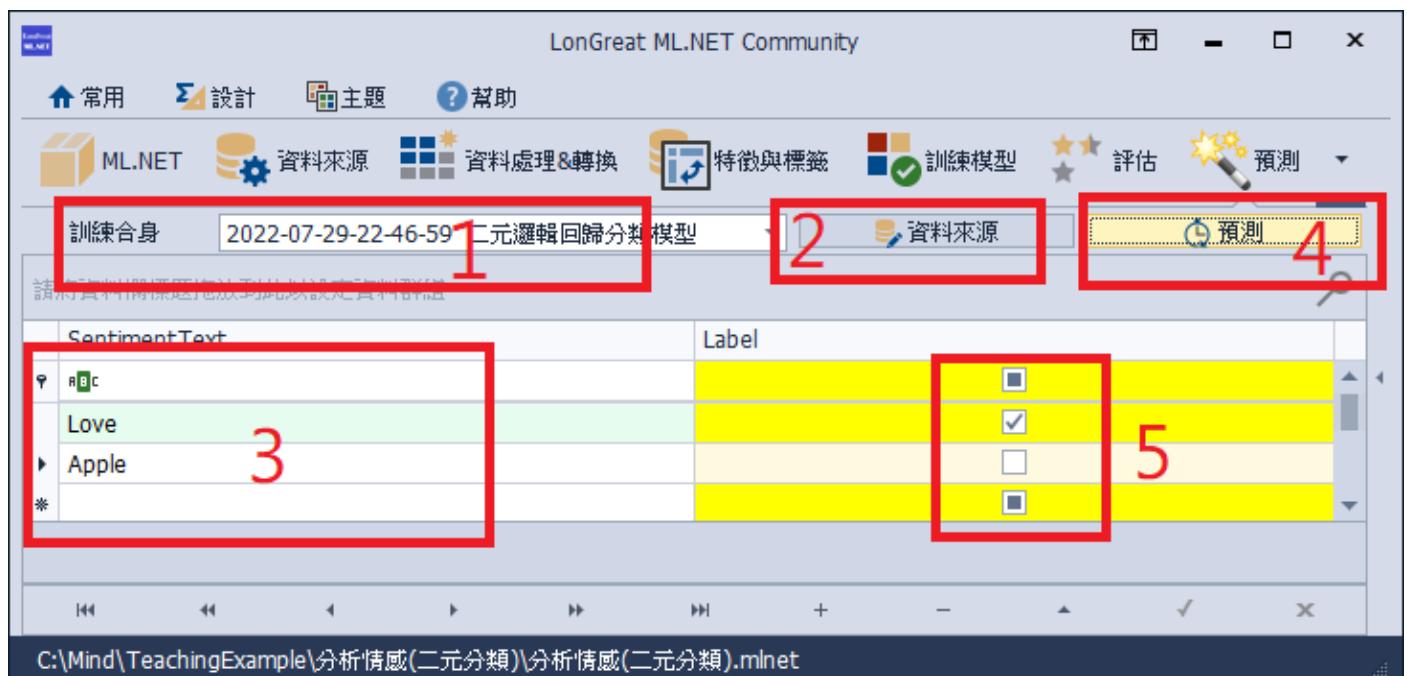
1. 選取以訓練完合身的模型
2. 選取設定資料須與原來相同 SentimentText 與 Label
3. 點擊預測按鈕=>產生出批次預測結果
4. 點擊評估按鈕=>評估結果



Sentiment Text	Label	Features	Predicted Label	Score	Probability
Wow... Loved this place.	1	[0.20412...	1	3.8220348	0.9785854
Crust is not good.	0	[0.23570...	0	-5.5448627	0.003892...
Not tasty and the texture was just nasty.	0	[0.14907...	0	-1.0571816	0.2578484
Stopped by during the late May bank holiday off Rick...	1	[0.10482...	1	2.4970417	0.92393416
The selection on the menu was great and so were t...	1	[0.23094...	1	4.6867533	0.9908676
Now I am getting angry and I want my damn pho.	0	[0.14142...	0	-4.3389544	0.012882...
Honesty it didn't taste THAT fresh.)	0	[0.16439...	0	-2.0588126	0.11316494
The potatoes were like rubber and you could tell th...	0	[0.09090...	0	-1.23924	0.22456829
The fries were great too.	1	[0.2,0,2,0...	1	3.5162592	0.9711469
A great touch.	1	[0.26726...	1	4.1488705	0.984463
Service was very prompt.	1	[0.20412...	1	3.0930505	0.9566052
Would not go back.	0	[0.23570...	0	-5.914437	0.002692...
The cashier had no care what so ever on what I had...	0	[0.09090...	0	-1.0071263	0.2675426
I tried the Cape Cod ravoli, chicken, with cranberry.....	1	[0.12598...	1	2.9536443	0.95043546

預測

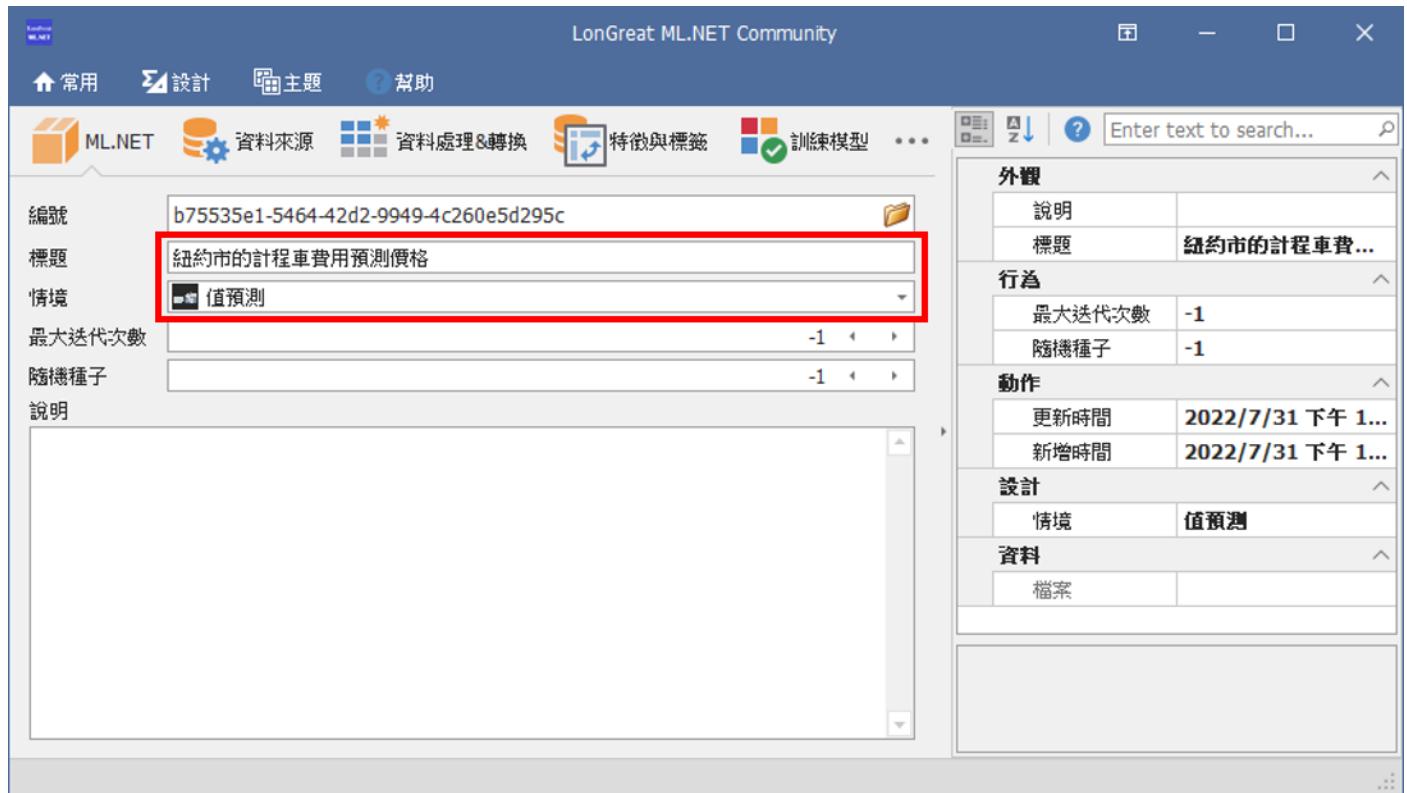
1. 選取以訓練完合身的模型
2. 點擊資料來源按鈕=>產生資料表
3. 輸入資料表 SentimentText 欄位
4. 點擊預測按鈕
5. 產生出 Label 預測結果



紐約市的計程車費用預測價格(迴歸模型)

建立專案

1. 輸入標題 => 紐約市的計程車費用預測價格
2. 選擇情境 => 值預測



準備並瞭解資料

1. 下載 [taxi-fare-train.csv](#) 和 [taxi-fare-test.csv](#) 資料集，並將它們儲存至您在上一個步驟所建立的 *Data* 資料夾。我們可以使用這些資料集將機器學習模型定型，然後評估模型的準確程度。這些資料集原先來自 [NYC TLC Taxi Trip 資料集](#)。
2. 開啟 [taxi-fare-train.csv](#) 資料集，然後查看第一個資料列中的資料行標頭。請查看每個資料行。瞭解資料，並決定哪些資料行是 **features**，以及哪一個資料行是 **label**。

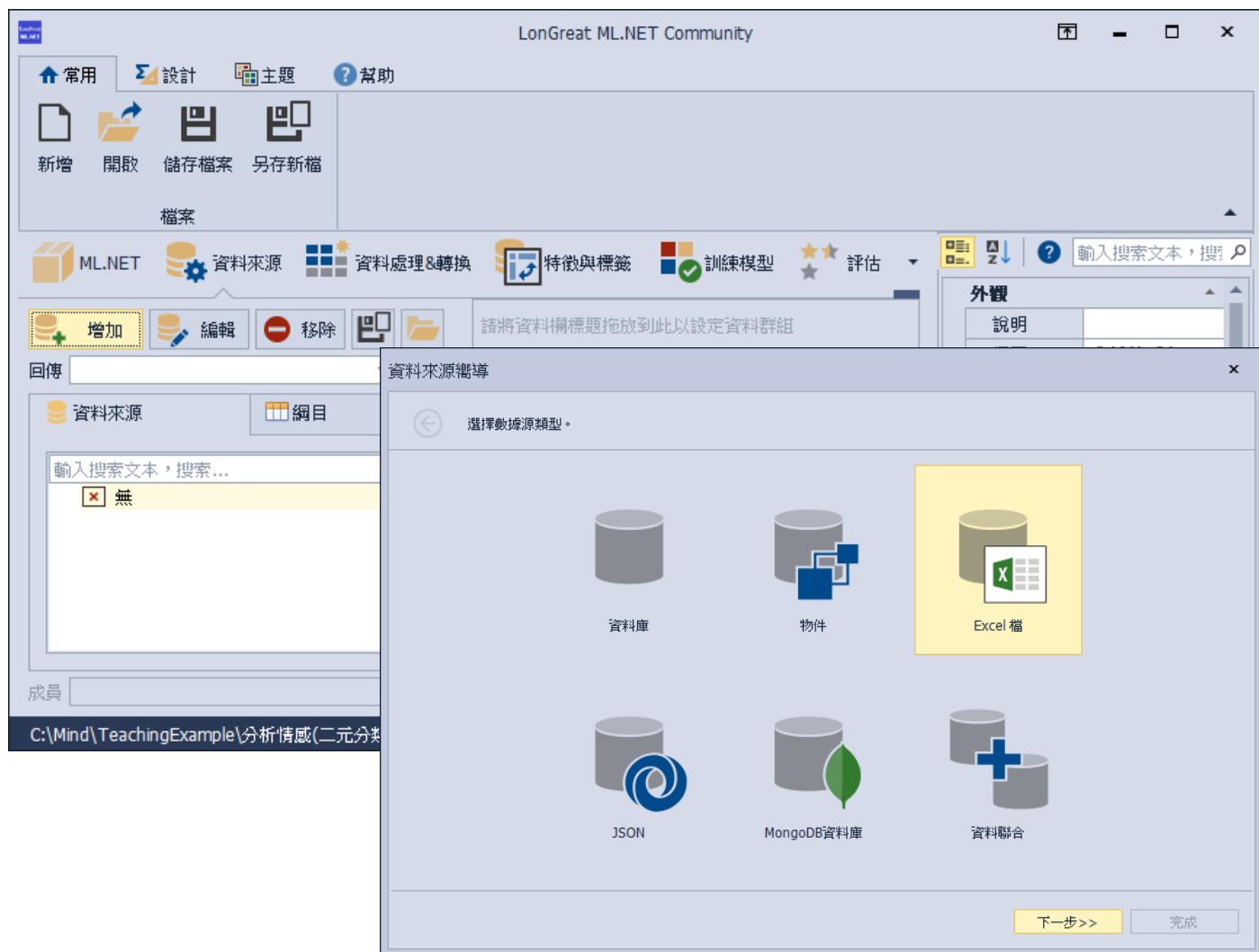
label 是您希望進行預測的資料行。識別的 **Features** 是您提供模型來預測 **Label** 的輸入。

提供的資料集包含下列資料行：

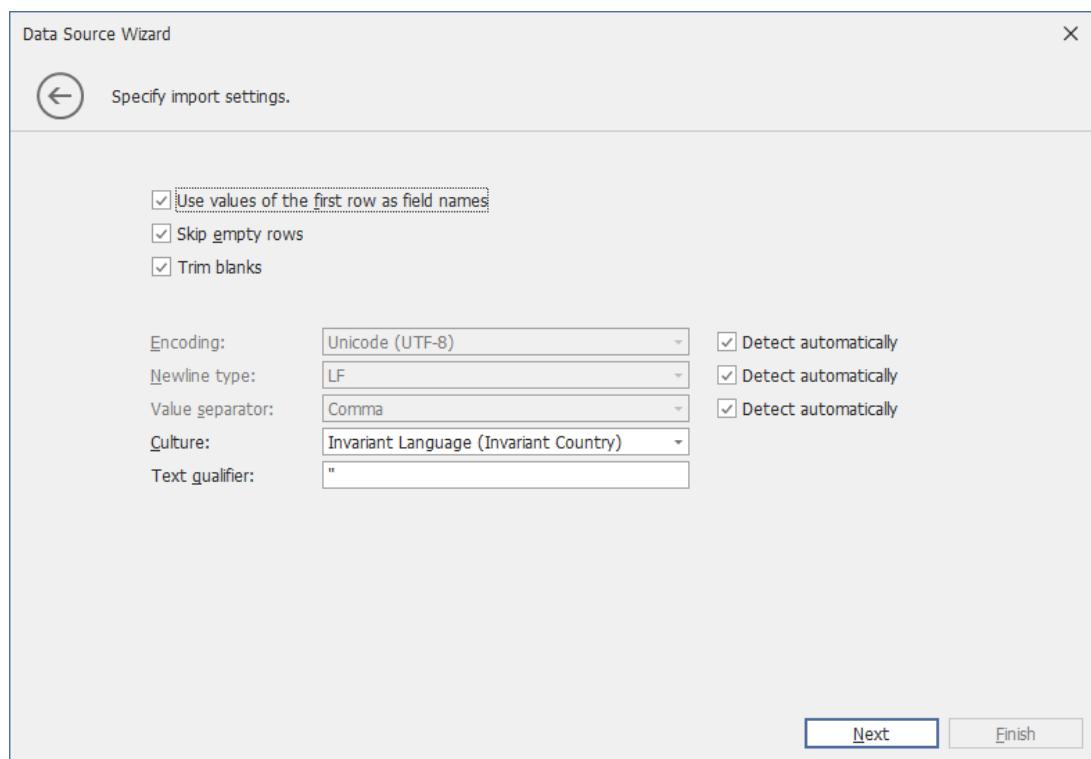
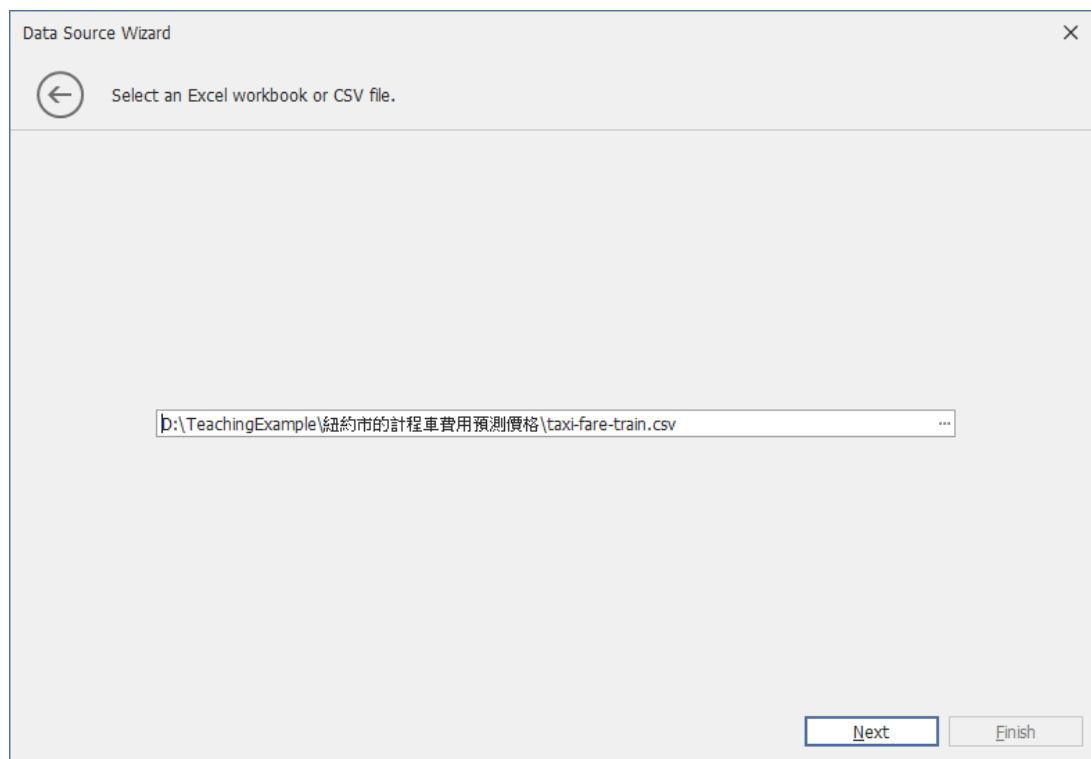
- vendor_id：計程車廠商的識別碼是一項特徵。
- rate_code：計程車行程的費率類型是一項特徵。
- passenger_count：行程的乘客數目是一項特徵。
- trip_time_in_secs：行程所花費的時間長度。您想要在行程結束之前預測行程的車資。屆時，您不知道行程會花多少時間。因此，行程時間不是一項特徵，您將從模型中排除這個資料行。
- trip_distance：行程的距離是一項特徵。
- payment_type：付款方式（現金或信用卡）是一項特徵。
- fare_amount：計程車車資總計是標籤。

載入資料

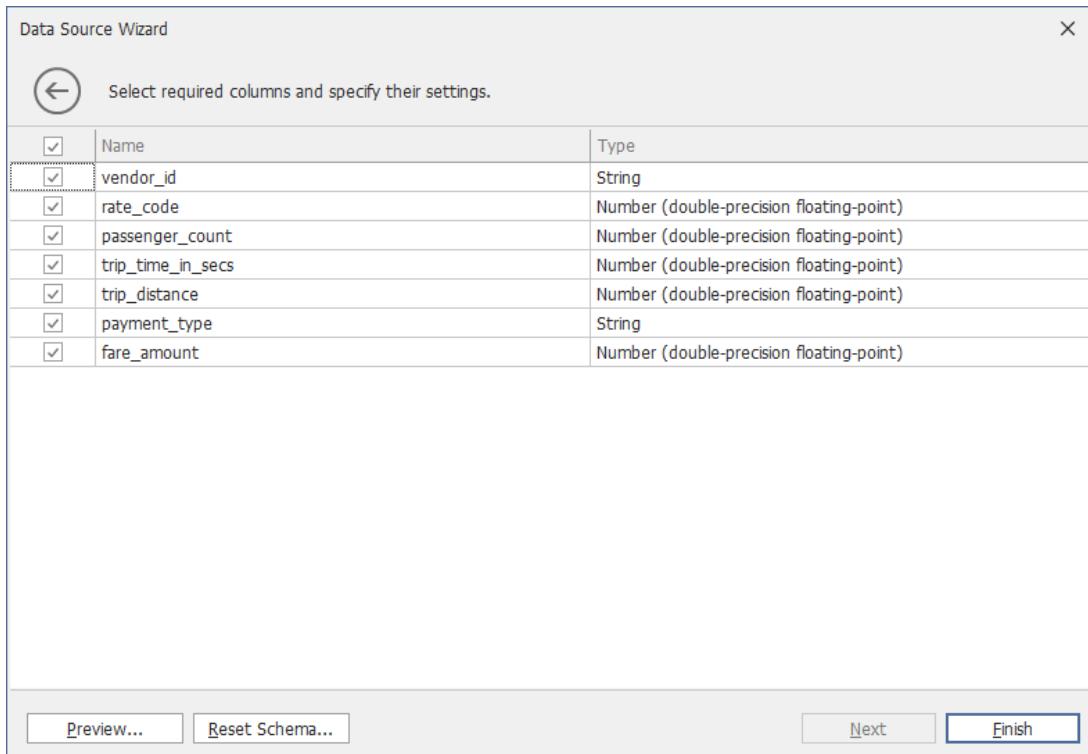
資料來源->增加->Excel 檔



選取 taxi-fare-train.csv



按完成



檢視資料

The screenshot shows the ML.NET Studio interface with the following components:

- Toolbar:** Includes '常用' (Common), '設計' (Design), '主題' (Theme), '幫助' (Help), 'ML.NET' icon, '資料來源' (DataSource), '資料處理&轉換' (Data Processing & Transformation), '特徵與標籤' (Features & Labels), '訓練模型' (Train Model), '評估' (Evaluate), '預測' (Predict), '日誌' (Log), and a question mark icon.
- DataSource1 Properties:** A properties panel on the right side with sections for '外觀' (Appearance), '行為' (Behavior), '動作' (Actions), '設計' (Design), and '資料' (Data). It shows details like '說明' (Description: 紐約市的計程...), '標題' (Title: -1), '最大值...' (Max Value: -1), '隨機種子' (Random Seed: -1), '更新時間' (Update Time: 2022/7/31 ...), '新增時間' (Create Time: 2022/7/31 ...), and '檔案' (File).
- Data Grid:** A large table displaying data from 'excelDataSource1'. The columns are: vendor_id, rate_code, passenger_c..., trip_time_in..., trip_distance, payment_type, fare_amount. The data consists of approximately 100,000 rows, mostly labeled 'CMT'.
- DataSource1 Properties:** A properties panel on the left showing the connection details for 'excelDataSource1'.
- Search Bar:** A search bar at the top right of the interface.

資料轉換

新增資料轉換

1. 最常編碼向量 => 將 vendor_id 轉換為編碼向量

The screenshot shows the ML.NET Data Processing Studio interface. On the left, the 'Source Fields' pane lists fields: vendor_id, rate_code, passenger_c..., trip_time_in..., trip_distance, payment_type, fare_amount. The 'Transformations' pane shows a list of steps:

步驟	標題	形別	資料轉換	已訓練
1	VendorIdEncoded	Single[1]	最常編碼向量	<input type="checkbox"/>
2	RateCodeEncoded	Single[1]	最常編碼向量	<input type="checkbox"/>
3	PaymentTypeEncoded	Single[1]	最常編碼向量	<input type="checkbox"/>
4	PassengerCountMV	PassengerCountMV	均值方差標準化	<input type="checkbox"/>
5	TripTimeMV	TripTimeMV	均值方差標準化	<input type="checkbox"/>
6	TripDistanceMV	TripDistanceMV	均值方差標準化	<input type="checkbox"/>
7	Label	Label	複製資料欄	<input type="checkbox"/>

The 'Output Column' section on the right shows: 輸入欄位 vendor_id, 最大鍵數 1000000, 輸出種類 Indicator, 鍵序數 ByOccurrence.

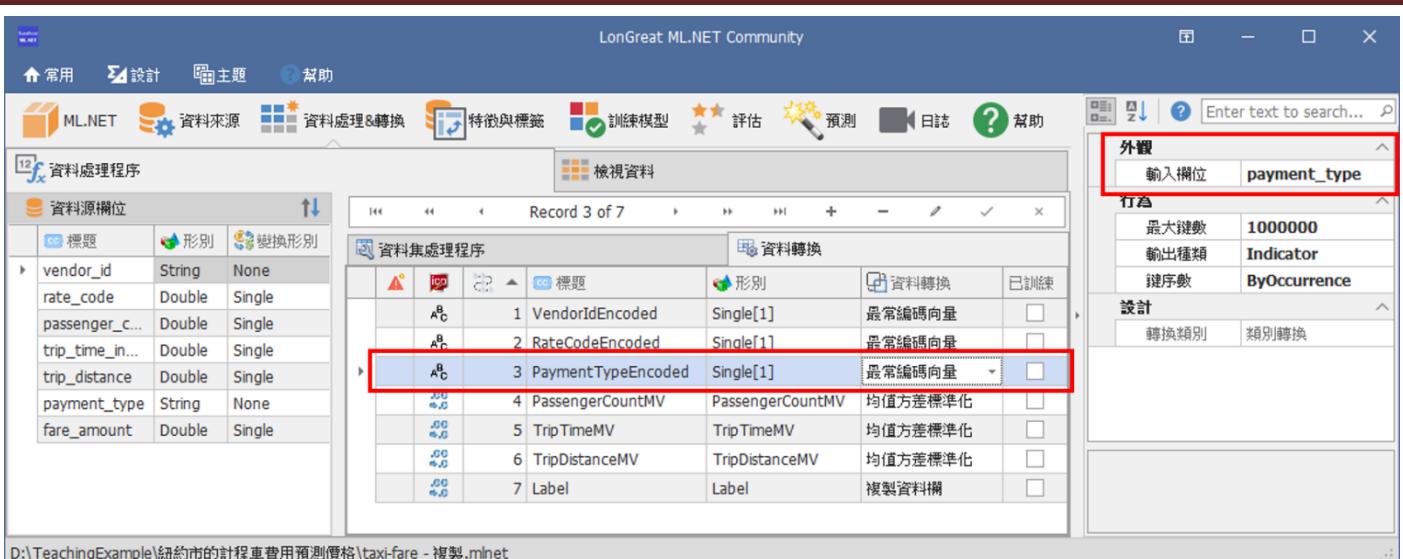
2. 最常編碼向量 => 將 rate_code 轉換為編碼向量

The screenshot shows the ML.NET Data Processing Studio interface. On the left, the 'Source Fields' pane lists fields: vendor_id, rate_code, passenger_c..., trip_time_in..., trip_distance, payment_type, fare_amount. The 'Transformations' pane shows a list of steps:

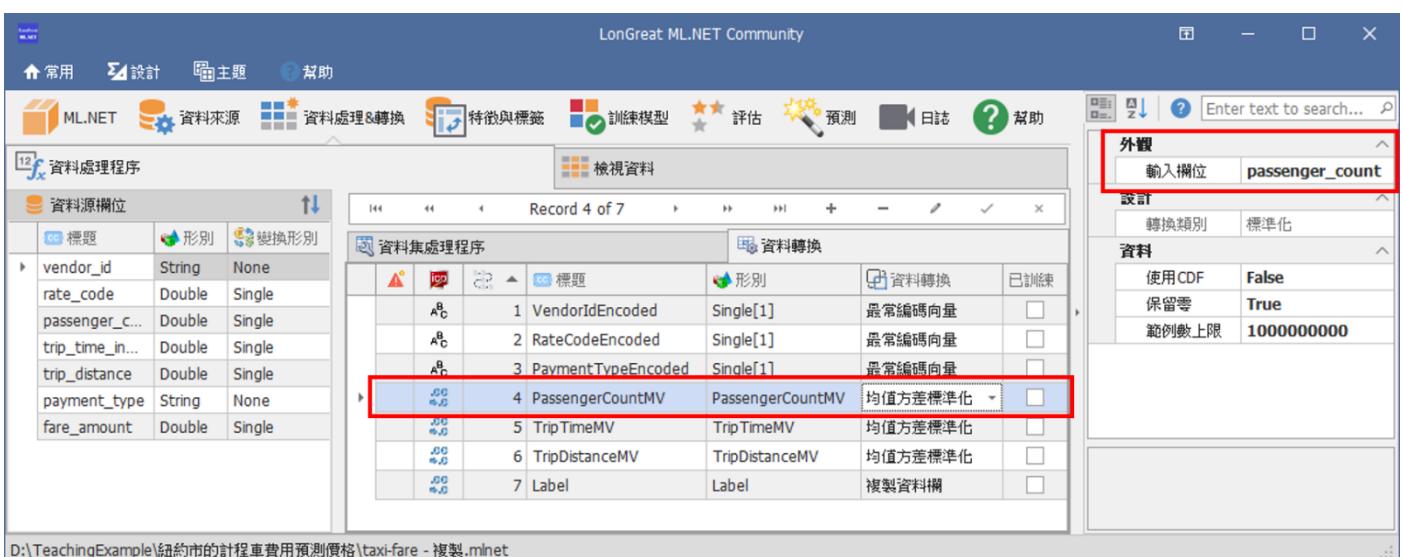
步驟	標題	形別	資料轉換	已訓練
1	VendorIdEncoded	Single[1]	最常編碼向量	<input type="checkbox"/>
2	RateCodeEncoded	Single[1]	最常編碼向量	<input type="checkbox"/>
3	PaymentTypeEncoded	Single[1]	最常編碼向量	<input type="checkbox"/>
4	PassengerCountMV	PassengerCountMV	均值方差標準化	<input type="checkbox"/>
5	TripTimeMV	TripTimeMV	均值方差標準化	<input type="checkbox"/>
6	TripDistanceMV	TripDistanceMV	均值方差標準化	<input type="checkbox"/>
7	Label	Label	複製資料欄	<input type="checkbox"/>

The 'Output Column' section on the right shows: 輸入欄位 rate_code, 最大鍵數 1000000, 輸出種類 Indicator, 鍵序數 ByOccurrence.

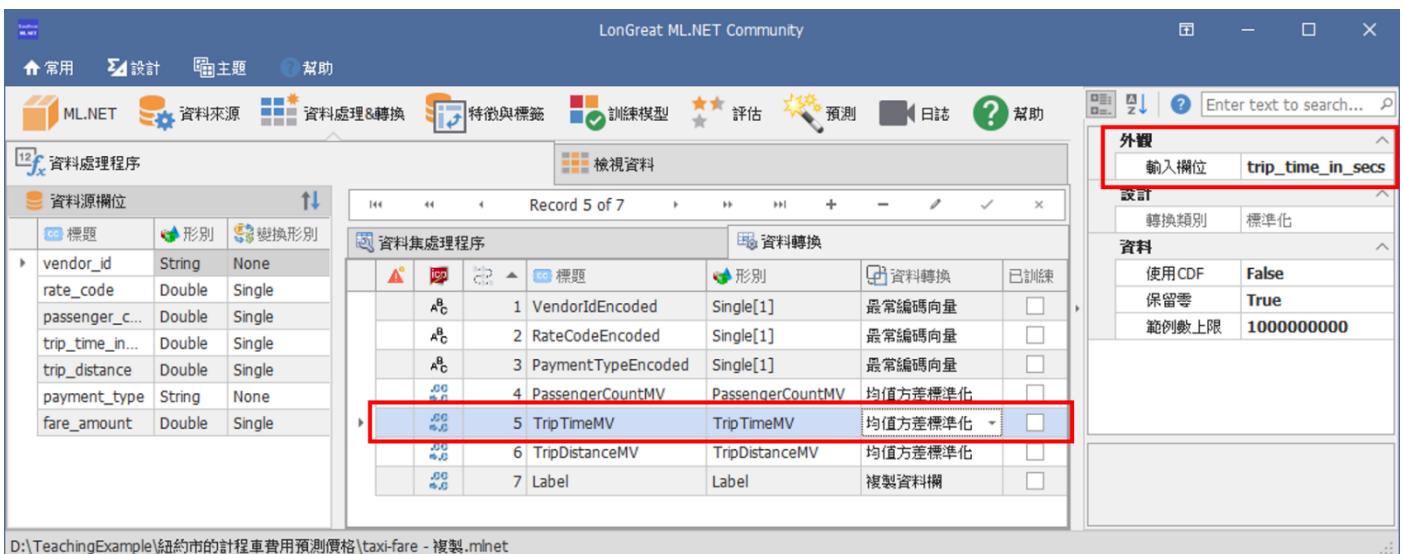
3. 最常編碼向量 => 將 payment_type 轉換為編碼向量



4. 均值方差標準化 => 將 passenger_count 轉換為均值方差標準化

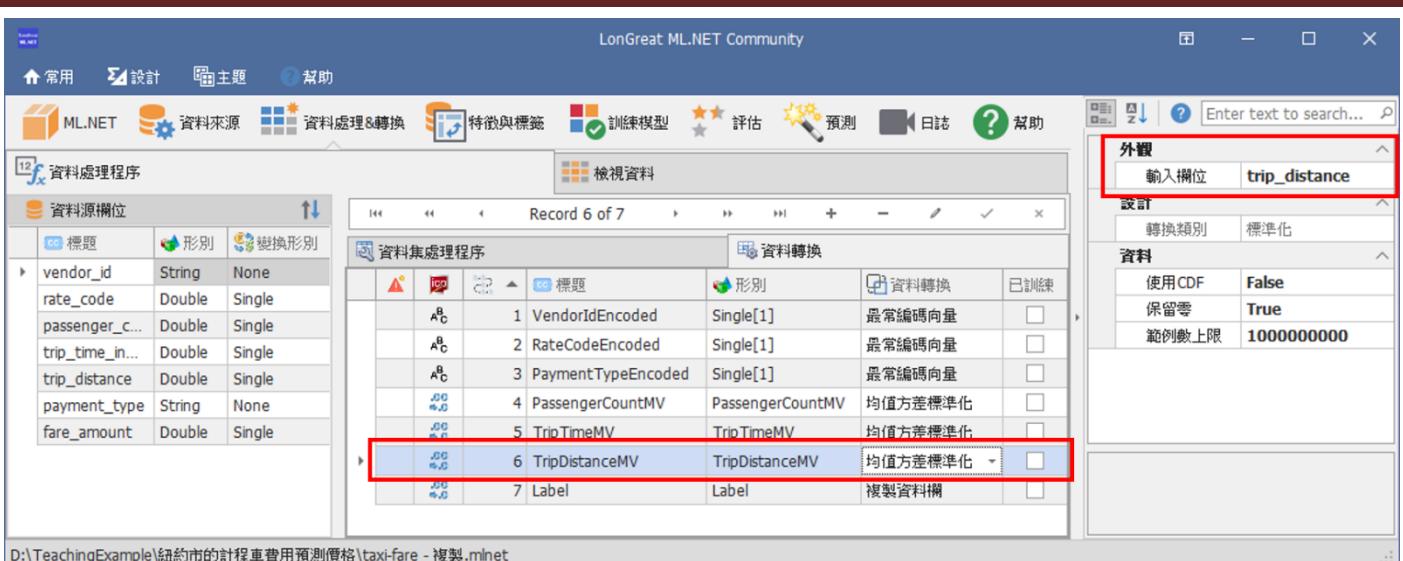


5. 均值方差標準化 => 將 trip_time_in_secs 轉換為均值方差標準化

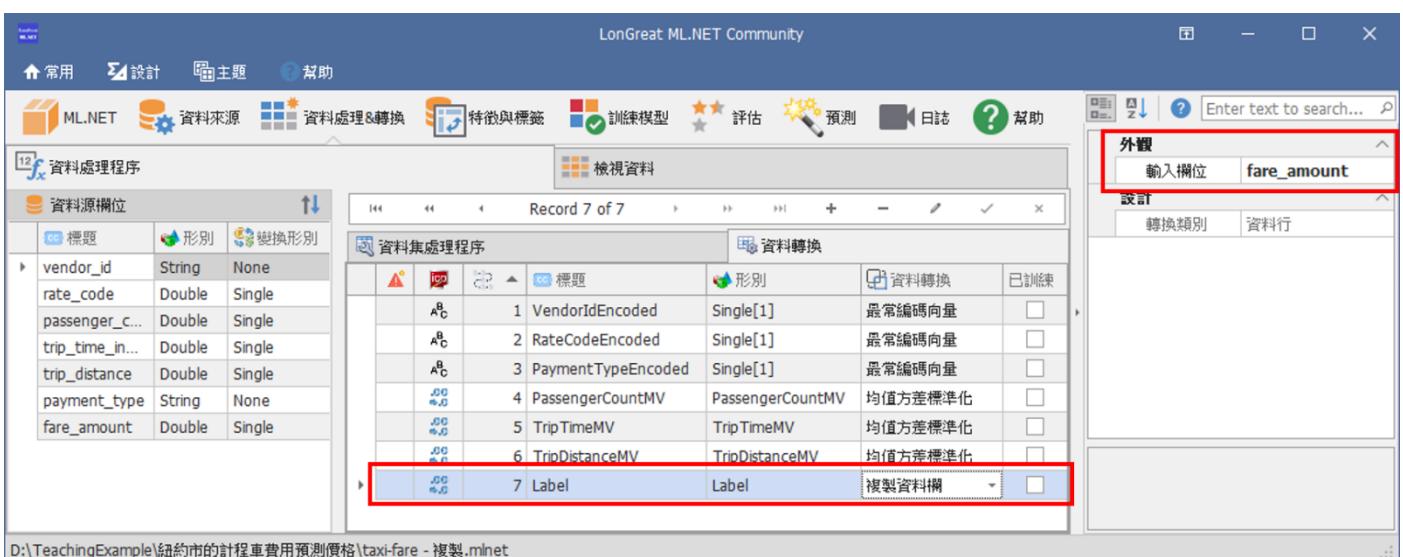


6. 均值方差標準化 => 將 trip_distance 轉換為均值方差標準化

LonGreat ML.NET Community



7. 複製資料欄 => 將 fare_amount 複製



結果檢視

- I. 至檢視資料頁籤
- II. 選擇顯示筆數
- III. 按充滿資料

LonGreat ML.NET Community

The screenshot shows the ML.NET Studio interface with a data grid containing 10,000 rows of taxi fare data. The columns include vendor_id, rate_code, passenger_count, trip_time_in_secs, trip_distance, payment_type, fare_amount, VendorIdEncoded, RateCodeEncoded, PaymentTypeEncoded, PassengerCountMV, TripTimeMV, TripDistanceMV, and Label. The '檢視資料' (View Data) button is highlighted with a red box.

特徵與標籤

拖拉 VendorIdEncoded=>特徵

拖拉 RateCodeEncoded=>特徵

拖拉 PaymentTypeEncoded=>特徵

拖拉 PassengerCountMV=>特徵

拖拉 TripTimeMV=>特徵

拖拉 TripDistanceMV=>特徵

拖拉 Label =>標籤

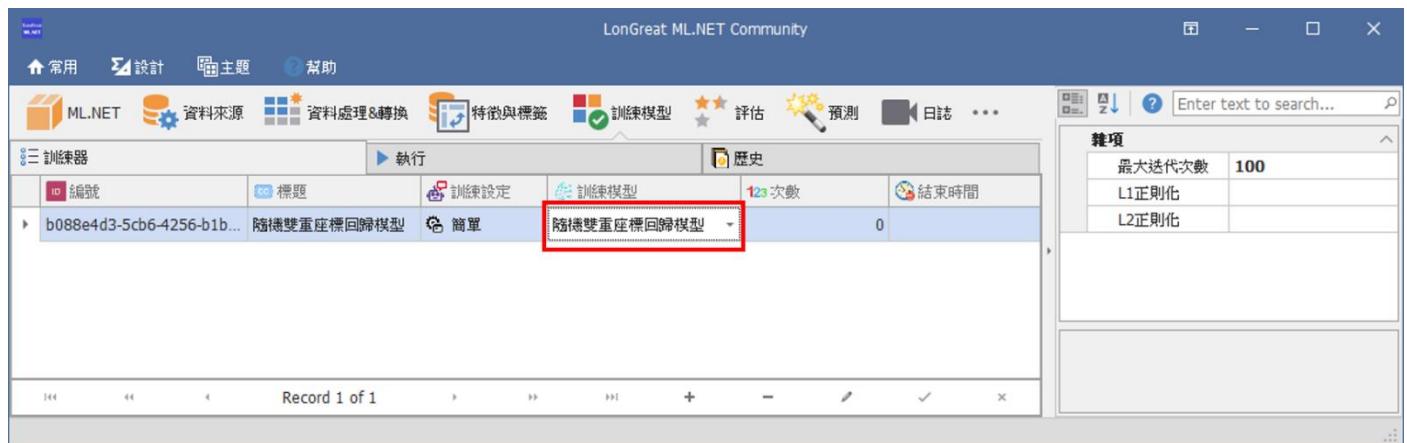
The screenshot shows the ML.NET Studio feature selection interface. It displays four categories: 忽略 (無), 特徵 (一或多), 標籤 (只有一), and 標重 (無或一). The '特徵 (一或多)' category contains VendorIdEncoded, RateCodeEncoded, PaymentTypeEncoded, PassengerCountMV, TripTimeMV, and TripDistanceMV. The '標籤 (只有一)' category contains Label. The '忽略 (無)' and '標重 (無或一)' categories are empty. A sidebar on the right shows the catalog and prediction settings.

訓練模型

訓練器

按+新增

選擇隨機雙重座標回歸模型(SdcaRegression)

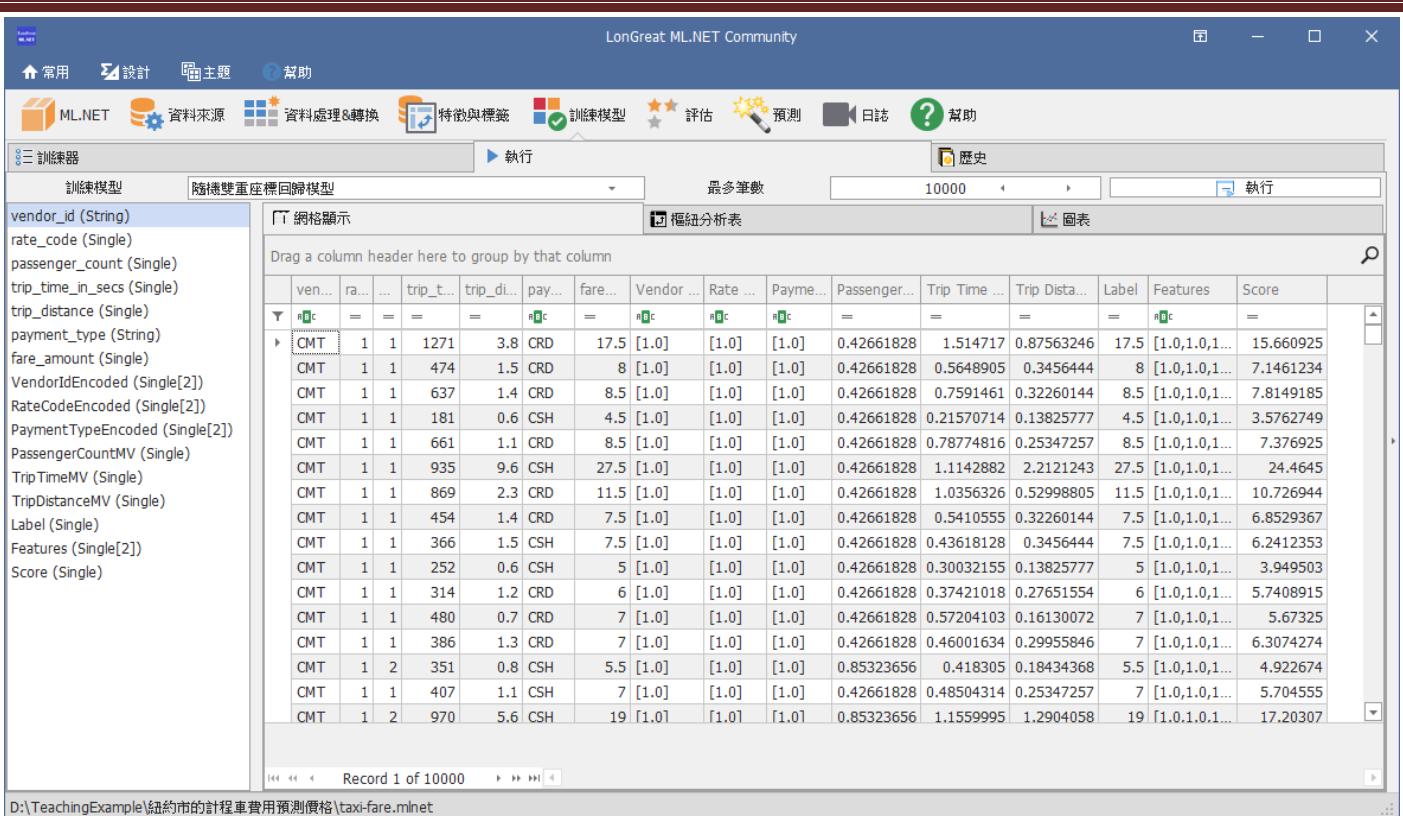


執行

1. 選擇訓練模型
2. 要顯示最多筆數
3. 執行

結果多二個欄位

- 特徵 PredictedLabel
- 分數 Score



評估

- A. 選取以訓練完合身的模型
- B. 選取設定資料須與原來相同
- C. 點擊預測按鈕=>產生出批次預測結果
- D. 點擊評估按鈕=>評估結果

資料	
平均絕對誤差(MAE)	0.956015948805809
判定係數	0.8939507105933373
均方誤差(MSE)	9.539699943138512
根均方誤差(RMSE)	3.0886404684162434
損失函數	9.539699909326917

平均絕對誤差(MAE)

絕對損失定義為 $L_1 = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i|$, 其中 m 是測試集中的實例數， \hat{y}_i 是每個實例的預測標籤，和 y_i 是每個實例的正確標籤。 L_1 損失是一個非負的遞減度量。較小的值表明該指標的模型更好。

LonGreat ML.NET Community

The screenshot shows the ML.NET Studio interface with the following details:

- Top Bar:** Includes "常用" (Common), "設計" (Design), "題目主題" (Topic Theme), and "幫助" (Help).
- Toolbar:** Features icons for "ML.NET", "資料來源" (Data Sources), "資料處理&轉換" (Data Processing & Transformation), "特徵與標籤" (Features & Labels), "訓練模型" (Train Model), "評估" (Evaluate), "預測" (Predict), "日誌" (Log), and "幫助" (Help).
- Left Sidebar:** Shows the "評估" (Evaluate) section with a table of feature names and their types.
- Central Grid View:** Displays a grid of 100,000 records from a taxi fare prediction model. The columns include vendor_id, rate_code, passenger_count, trip_time_in_secs, trip_distance, payment_type, fare_amount, VendorIdEncoded, RateCodeEncoded, PaymentTypeEncoded, PassengerCountMV, TripTimeMV, TripDistanceMV, Label, Features, and Score.
- Right Sidebar:**
 - 資料 (Data):** Shows summary statistics: 平均... 0.956015..., 累定... 0.893950..., 均方... 9.539699..., 根均... 3.088640..., 損失... 9.539699... .
 - 平均絕對誤差 (MAE):** Formula: $\frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i|$, where y_i is the test example's label, \hat{y}_i is the predicted value, and m is the number of examples.
- Bottom Status Bar:** Shows "Record 1 of 100000" and the file path "D:\TeachingExample\紐約市的計程車費用預測價格\taxi-fare.mlnet".

鳶尾花花卉分類

瞭解問題

這個問題是關於將一組鳶尾花按照花卉特徵分成不同的群組。這些特徵是萼片的長度和寬度以及花瓣的長度和寬度。本教學課程中假設不知道每個花卉的類型。您想要從特徵瞭解資料集的結構，還要預測資料執行個體如何符合此結構。

選取適當的機器學習工作

因為您不知道每個花卉屬於哪個群組，所以您選擇非監督式機器學習工作。若要按照類似元素歸於同一群組的方式來分割群組中的資料集，請使用群集機器學習工作。

準備資料

1. *Iris.xlsx* 資料集，如需有關鳶尾花資料集的詳細資訊，請參閱[鳶尾花資料集](#)維基百科頁面和[鳶尾花資料集](#)頁面（也就是資料集的來源）。

Iris.xlsx 檔案包含五個資料行，分別表示：

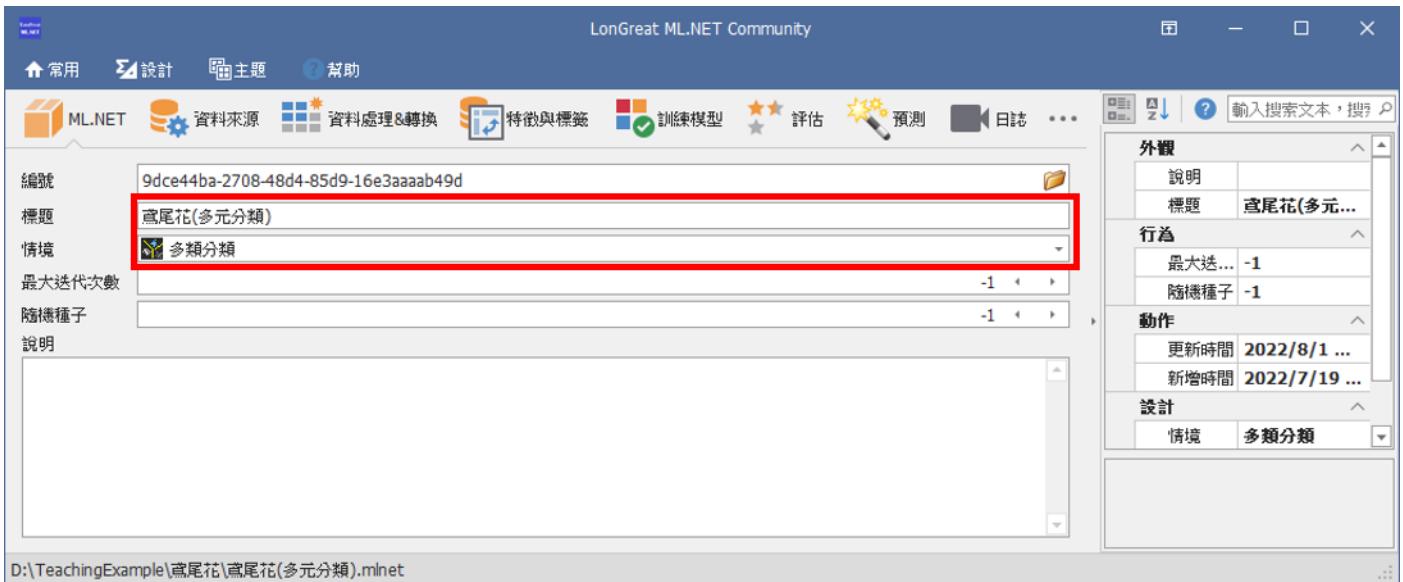
- 萼片長度 SepalLength (以公分為單位)
- 萼片寬度 SepalWidth (以公分為單位)
- 花瓣長度 PetalLength (以公分為單位)
- 花瓣寬度 PetalWidth (以公分為單位)
- 鳶尾花的類型 Class

鳶尾花(多元分類)

建立專案

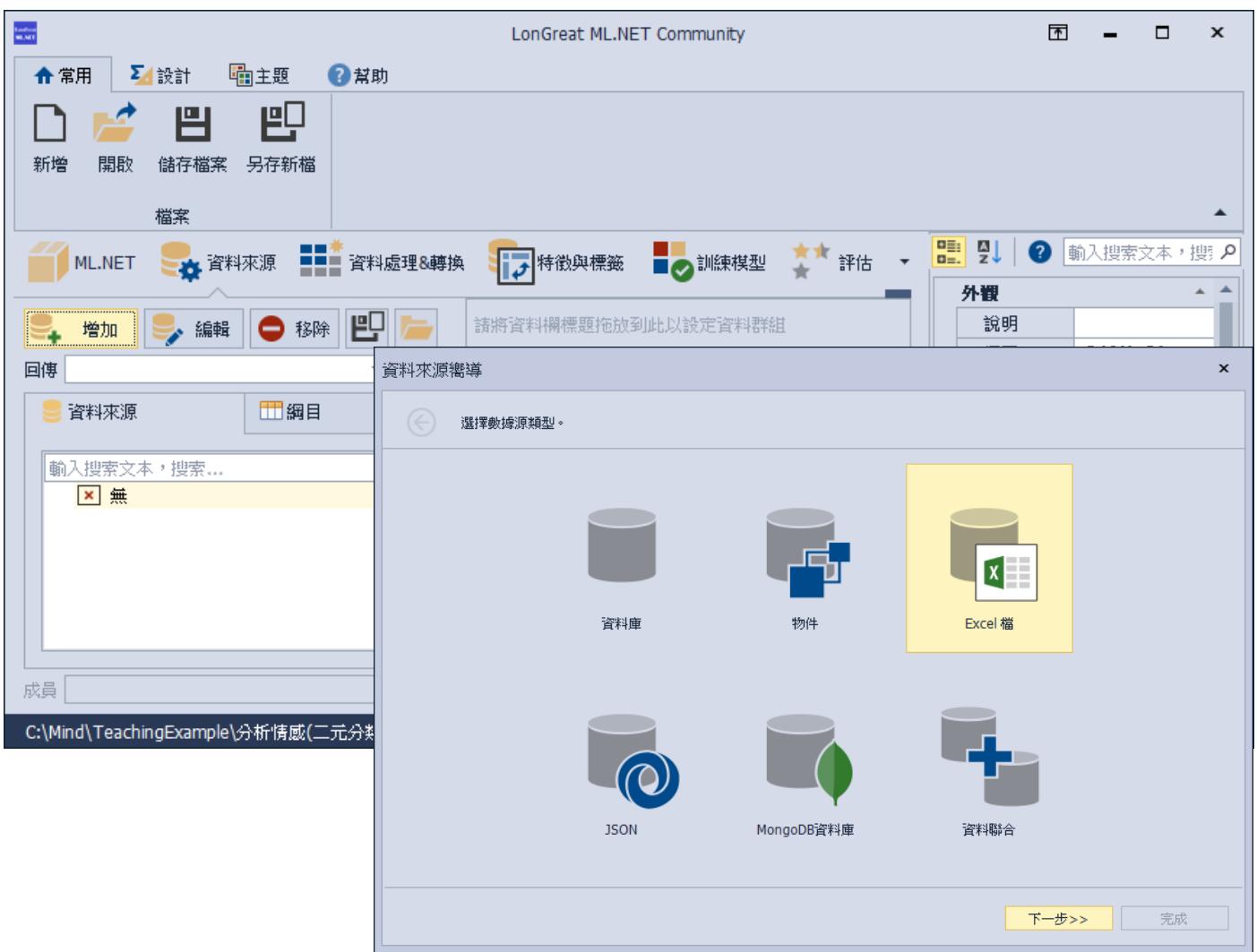
1. 輸入標題 => 鳶尾花(多元分類)
2. 選擇情境 => 多類分類

LonGreat ML.NET Community

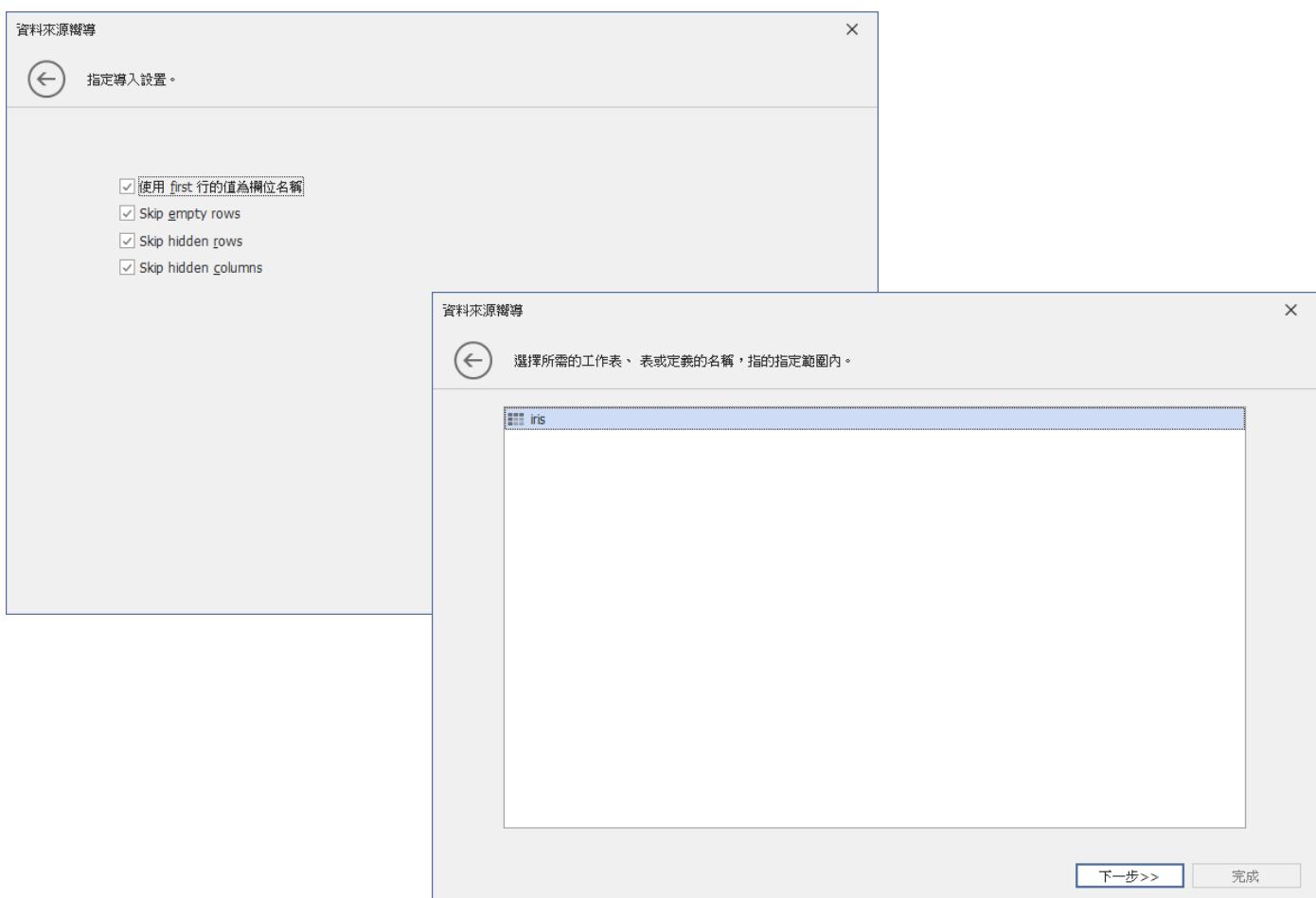
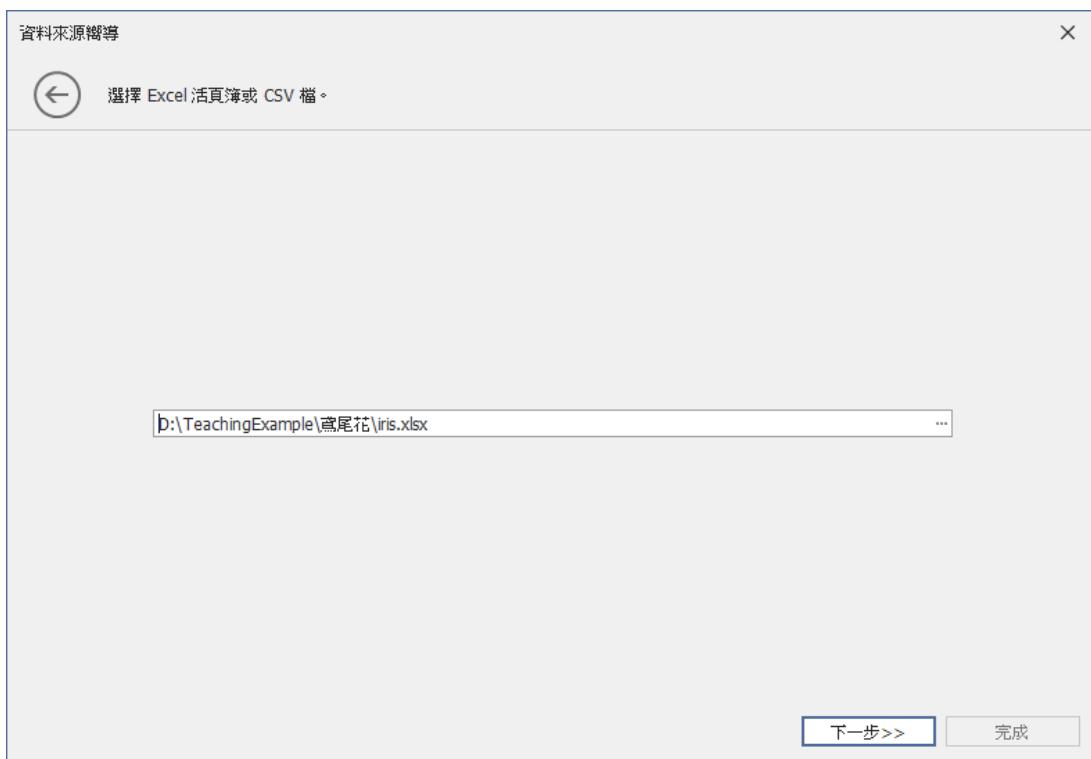


載入資料

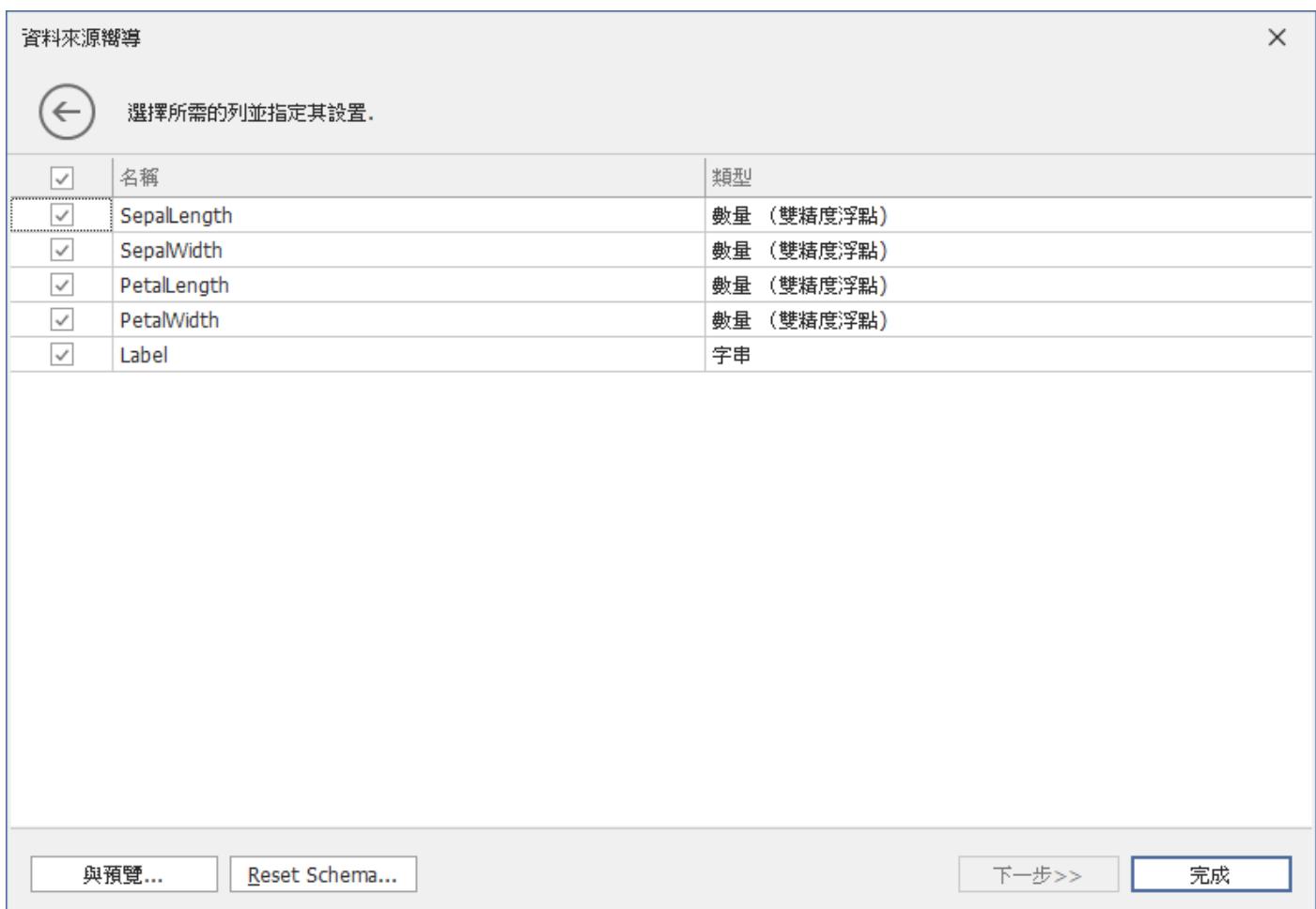
資料來源->增加->Excel 檔



選取 iris.xlsx



按完成



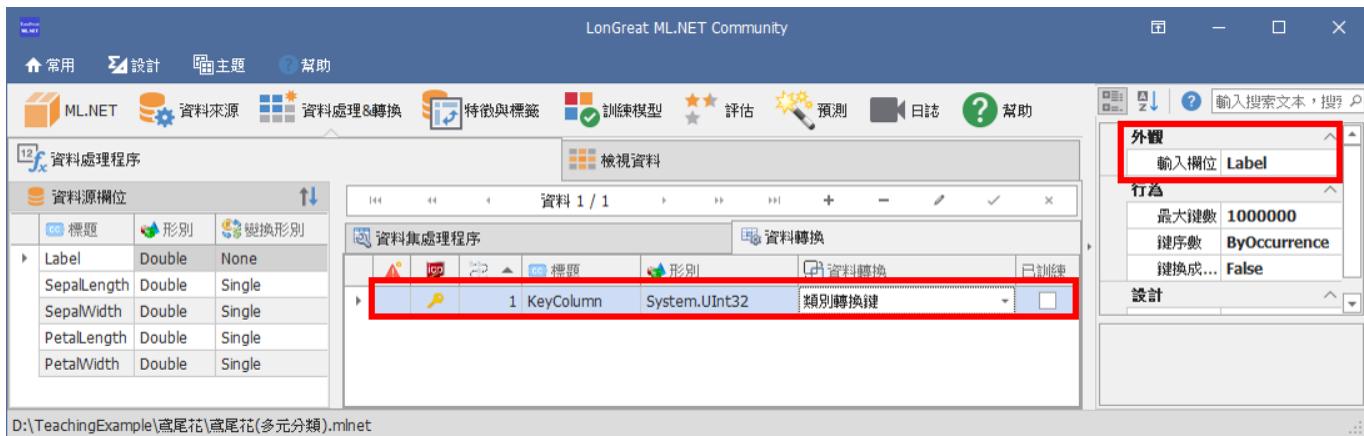
檢視資料

	Sepal Length	Sepal Width	Petal Length	Petal Width	Label
=	5.1	3.5	1.4	0.2	Iris-setosa
	4.9	3	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
	5	3.6	1.4	0.2	Iris-setosa
	5.4	3.9	1.7	0.4	Iris-setosa
	4.6	3.4	1.4	0.3	Iris-setosa
	5	3.4	1.5	0.2	Iris-setosa
	4.4	2.9	1.4	0.2	Iris-setosa
	4.9	3.1	1.5	0.1	Iris-setosa
	5.4	3.7	1.5	0.2	Iris-setosa
	4.8	3.4	1.6	0.2	Iris-setosa
	4.8	3	1.4	0.1	Iris-setosa
	4.3	3	1.1	0.1	Iris-setosa
	5.8	4	1.2	0.2	Iris-setosa
	5.7	4.4	1.5	0.4	Iris-setosa
	5.4	2.0	1.3	0.1	Iris-setosa

資料轉換

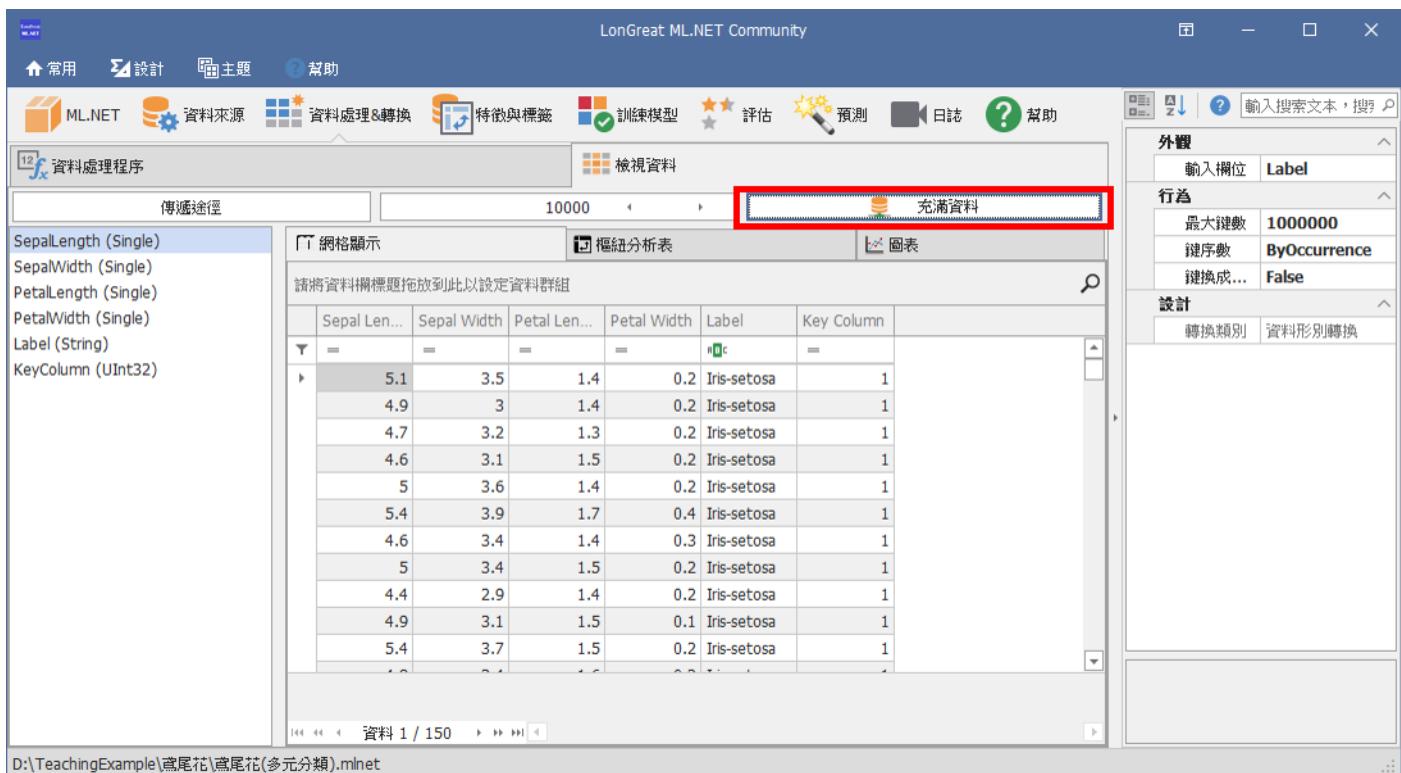
新增資料轉換

1. 類別轉換鍵 => 將 Label 轉換為鍵值



結果檢視

- 至檢視資料頁籤
- 選擇顯示筆數
- 按充滿資料



特徵與標籤

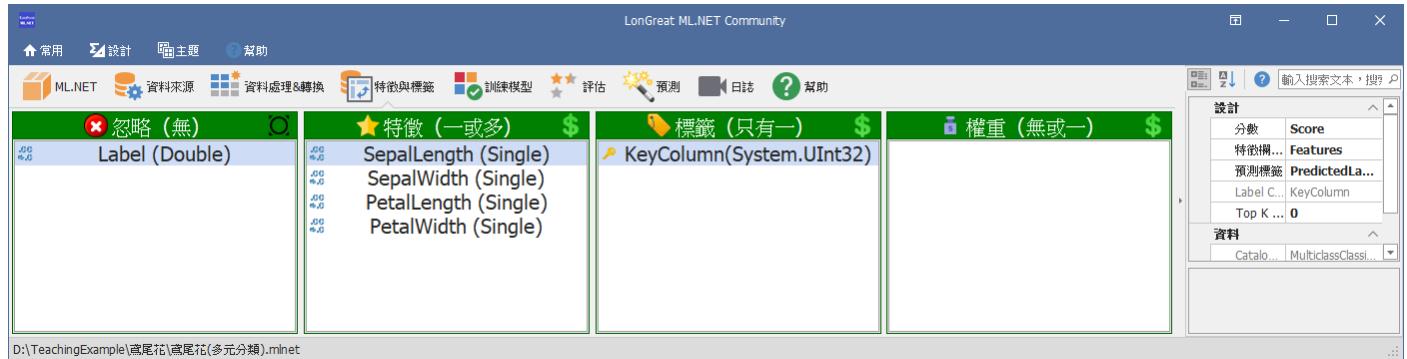
拖拉 SepalLength=>特徵

拖拉 SepalWidth=>特徵

拖拉 PetalLength=>特徵

拖拉 PetalWidth=>特徵

拖拉 KeyColumn=>標籤



訓練模型

訓練器

按+新增

選擇多元 L-BFGS 最大熵模型 (LbfgsMaximumEntropy)



執行

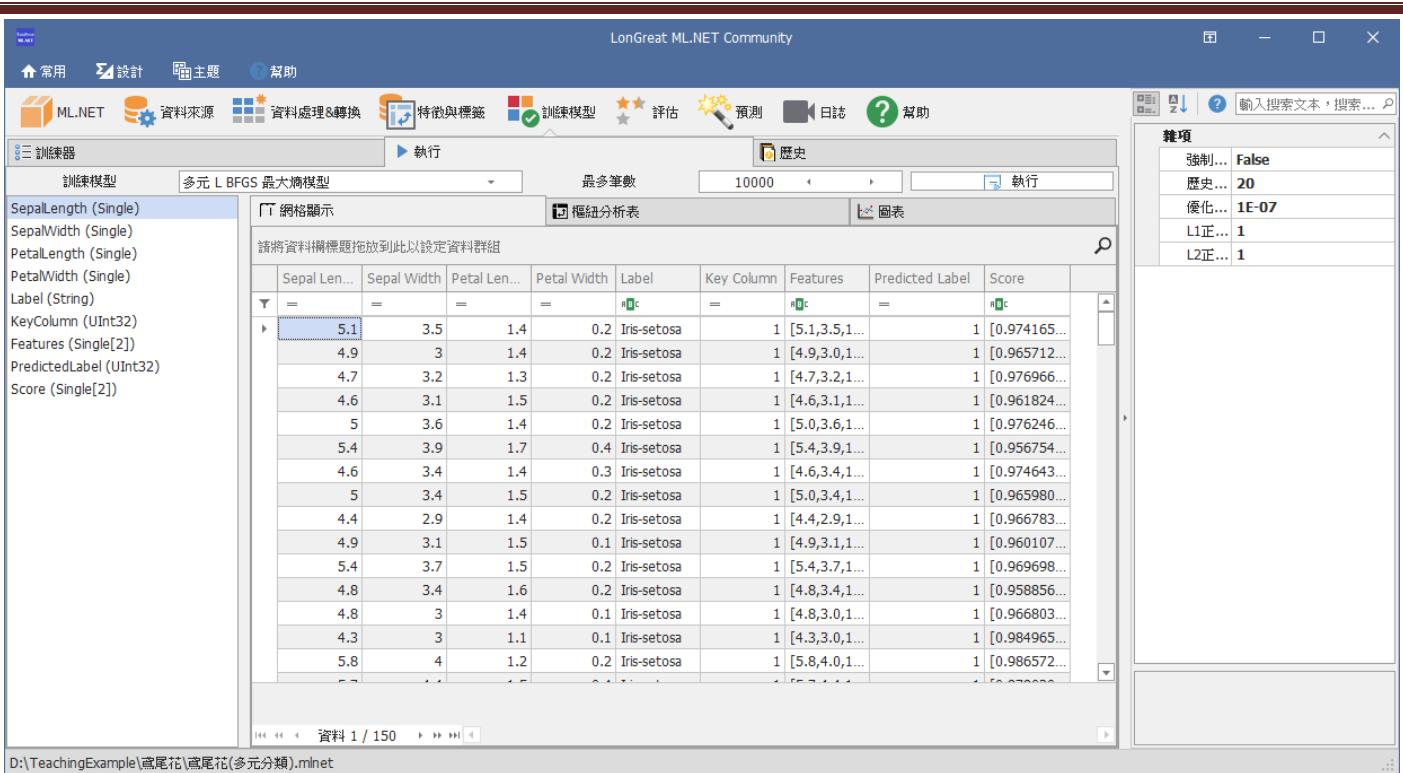
1. 選擇訓練模型
2. 要顯示最多筆數
3. 執行

結果多三個欄位

特徵 Features

預測標籤 PredictedLabel

分數 Score



評估

- A. 選取以訓練完合身的模型
- B. 選取設定資料須與原來相同
- C. 點擊預測按鈕=>產生出批次預測結果
- D. 點擊評估按鈕=>評估結果

資料	
宏觀準確率	0.9733333333333333
每類對數損失	(Collection)
所有 K 的 Top-K 準確度	(Collection)
混淆矩陣	MindCore.MlNet.Catalogs.ReturnConfusionMatrix
分類數	3
每類召回	(Collection)
每類精度	(Collection)
筆數	(Collection)
微觀準確率	0.973333333333334
對數損失	0.15254354312265206
對數損失減少	0.8611488832811196
Top-K準確率	0
Top-K預測計數	0

宏觀準確率

宏觀平均值是類級別的平均準確度。 計算每個類的準確度，宏觀準確度是這些準確度的平均值。宏觀平均指標為每個類賦予相同的權重，無論數據集中包含該類的多少實例。

LonGreat ML.NET Community

The screenshot shows the LonGreat ML.NET Community application interface. The main window displays a grid of data for the Iris dataset, with columns including Sepal Length, Sepal Width, Petal Length, Petal Width, Label, Key Column, Features, Predicted, and Score. A sidebar on the right provides detailed performance metrics, such as Macro Average Accuracy (0.973333333333333) and various collection counts (Collection, MindCore.MLNet.Cat...).

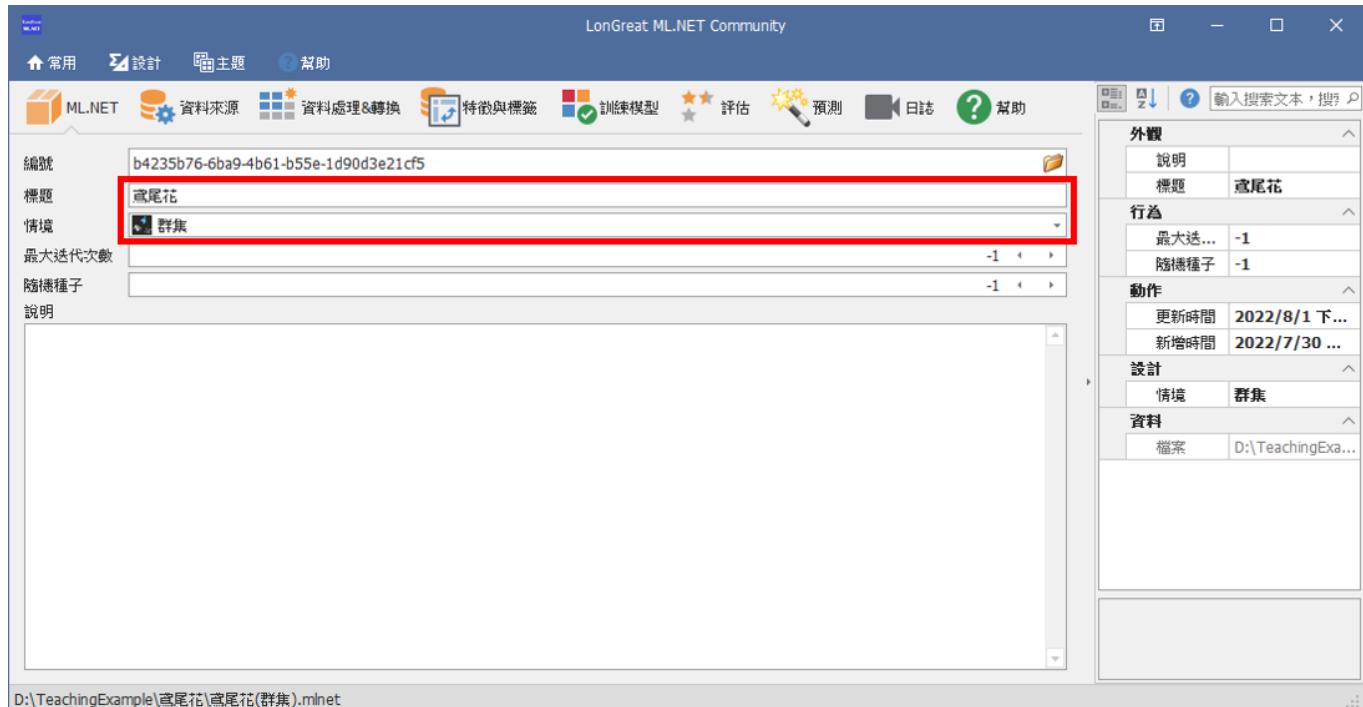
Sepal Length (Single)	Sepal Width (Single)	Petal Length (Single)	Petal Width (Single)	Label	Key Column	Features	Predicted	Score
5.1	3.5	1.4	0.2	Iris-setosa	1	[5.1,3.5,1...	1	[0.974165...
4.9	3	1.4	0.2	Iris-setosa	1	[4.9,3.0,1...	1	[0.965712...
4.7	3.2	1.3	0.2	Iris-setosa	1	[4.7,3.2,1...	1	[0.976966...
4.6	3.1	1.5	0.2	Iris-setosa	1	[4.6,3.1,1...	1	[0.961824...
5	3.6	1.4	0.2	Iris-setosa	1	[5.0,3.6,1...	1	[0.976246...
5.4	3.9	1.7	0.4	Iris-setosa	1	[5.4,3.9,1...	1	[0.956754...
4.6	3.4	1.4	0.3	Iris-setosa	1	[4.6,3.4,1...	1	[0.974643...
5	3.4	1.5	0.2	Iris-setosa	1	[5.0,3.4,1...	1	[0.965980...
4.4	2.9	1.4	0.2	Iris-setosa	1	[4.4,2.9,1...	1	[0.966783...
4.9	3.1	1.5	0.1	Iris-setosa	1	[4.9,3.1,1...	1	[0.960107...
5.4	3.7	1.5	0.2	Iris-setosa	1	[5.4,3.7,1...	1	[0.969698...
4.8	3.4	1.6	0.2	Iris-setosa	1	[4.8,3.4,1...	1	[0.958856...
4.8	3	1.4	0.1	Iris-setosa	1	[4.8,3.0,1...	1	[0.966803...
4.3	3	1.1	0.1	Iris-setosa	1	[4.3,3.0,1...	1	[0.984965...
5.8	4	1.2	0.2	Iris-setosa	1	[5.8,4.0,1...	1	[0.986572...
5.7	4.4	1.5	0.1	Iris-setosa	1	[5.7,4.4,1...	1	[0.970020...

D:\TeachingExample\鳶尾花\鳶尾花(多元分類).mlnet

鳶尾花(群集)

建立專案

1. 輸入標題 => 鳶尾花
2. 選擇情境 => 群集

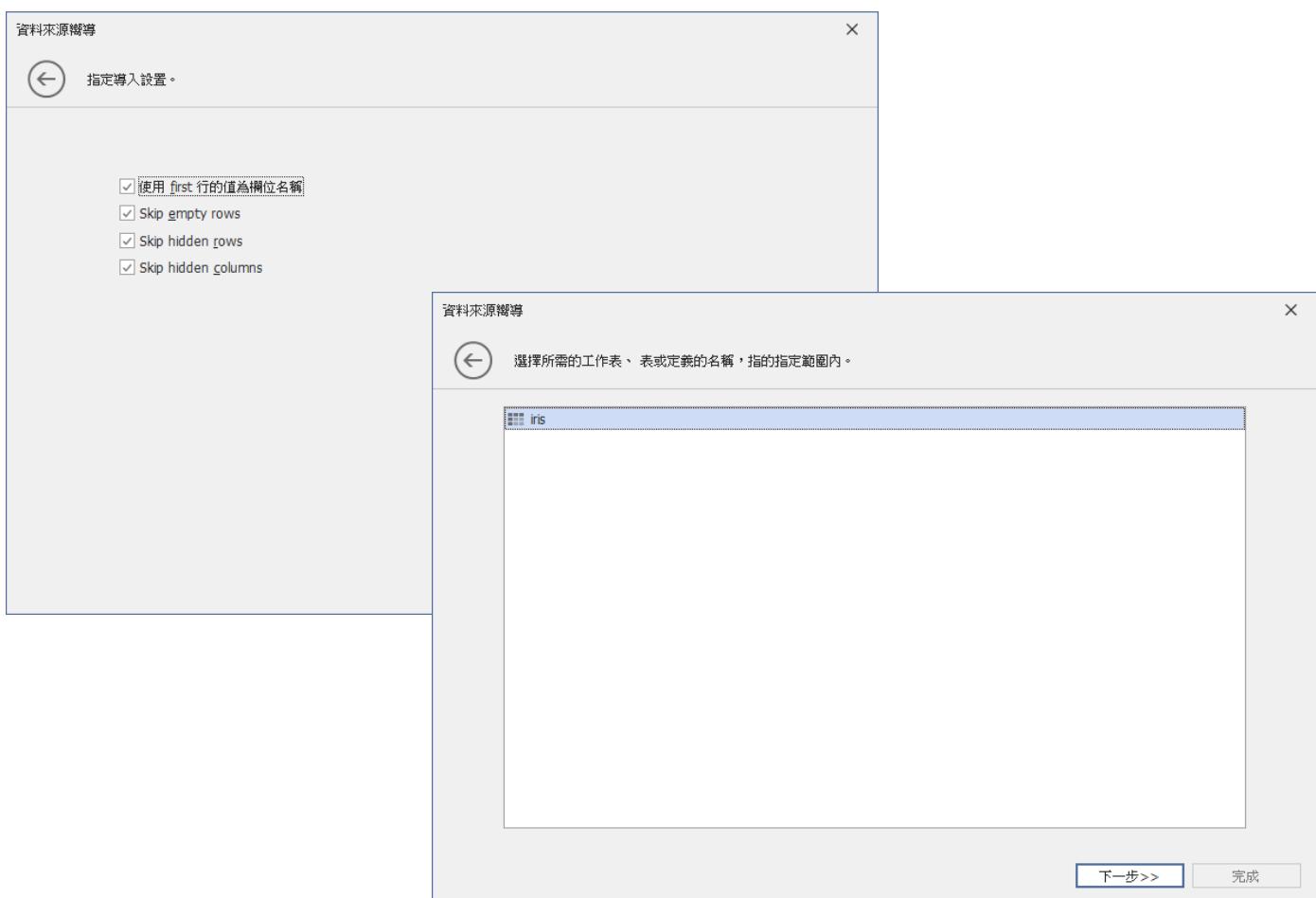
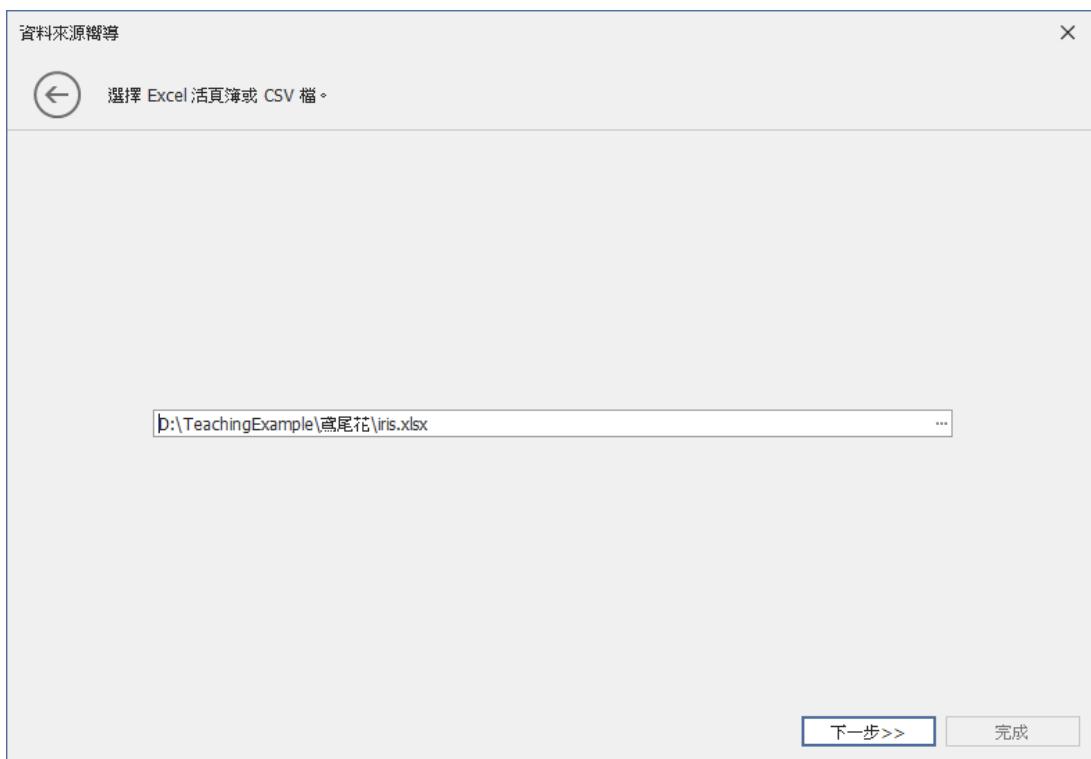


載入資料

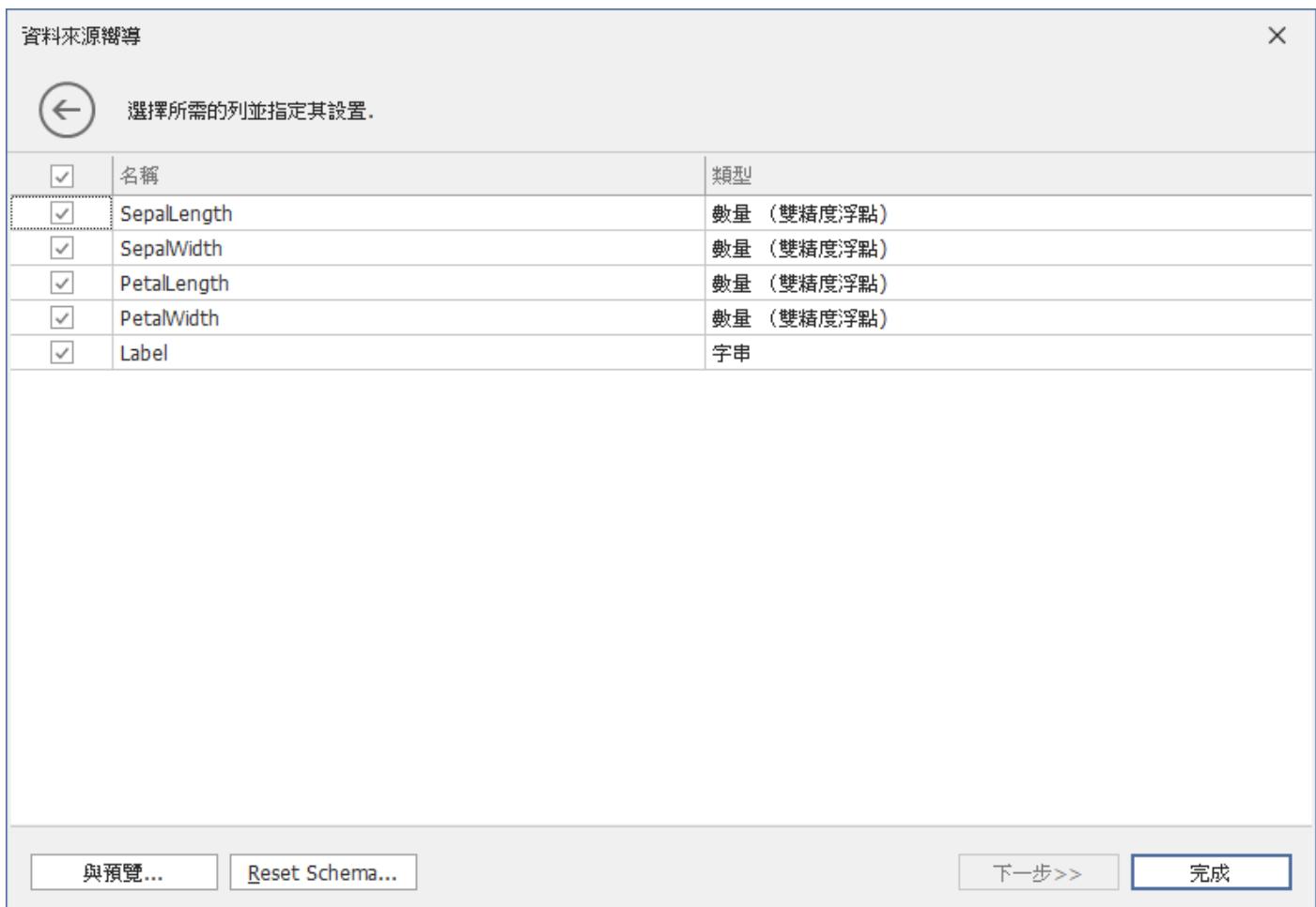
資料來源->增加->Excel 檔



選取 iris.xlsx



按完成



檢視資料

	Sepal Length	Sepal Width	Petal Length	Petal Width	Label
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	2.0	1.3	0.1	Iris-setosa
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					
31					
32					
33					
34					
35					
36					
37					
38					
39					
40					
41					
42					
43					
44					
45					
46					
47					
48					
49					
50					
51					
52					
53					
54					
55					
56					
57					
58					
59					
60					
61					
62					
63					
64					
65					
66					
67					
68					
69					
70					
71					
72					
73					
74					
75					
76					
77					
78					
79					
80					
81					
82					
83					
84					
85					
86					
87					
88					
89					
90					
91					
92					
93					
94					
95					
96					
97					
98					
99					
100					
101					
102					
103					
104					
105					
106					
107					
108					
109					
110					
111					
112					
113					
114					
115					
116					
117					
118					
119					
120					
121					
122					
123					
124					
125					
126					
127					
128					
129					
130					
131					
132					
133					
134					
135					
136					
137					
138					
139					
140					
141					
142					
143					
144					
145					
146					
147					
148					
149					
150					

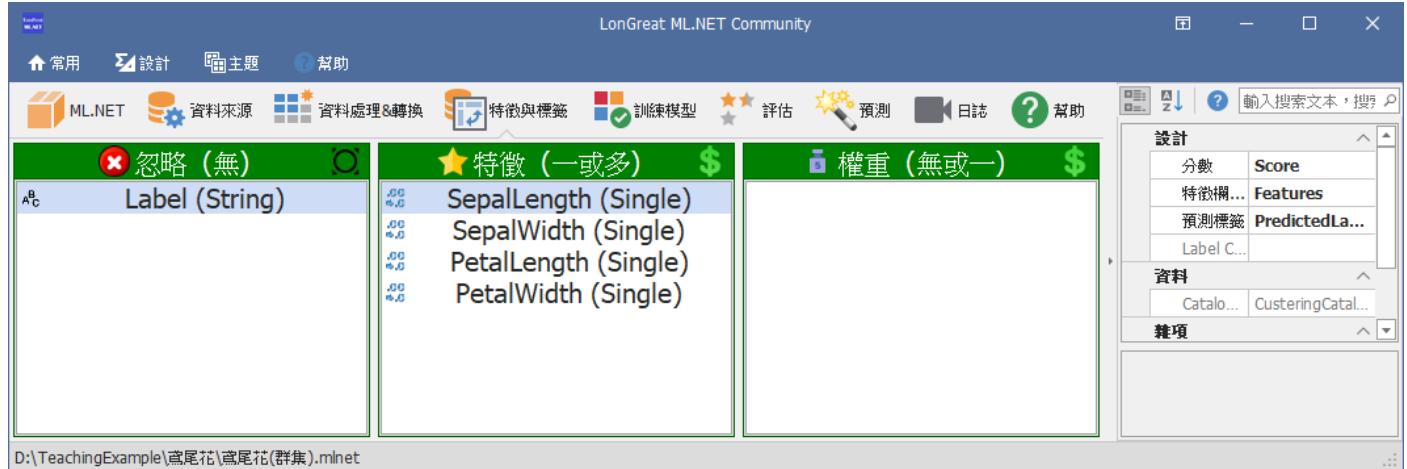
特徵與標籤

拖拉 SepalLength=>特徵

拖拉 SepalWidth=>特徵

拖拉 PetalLength=>特徵

拖拉 PetalWidth=>特徵



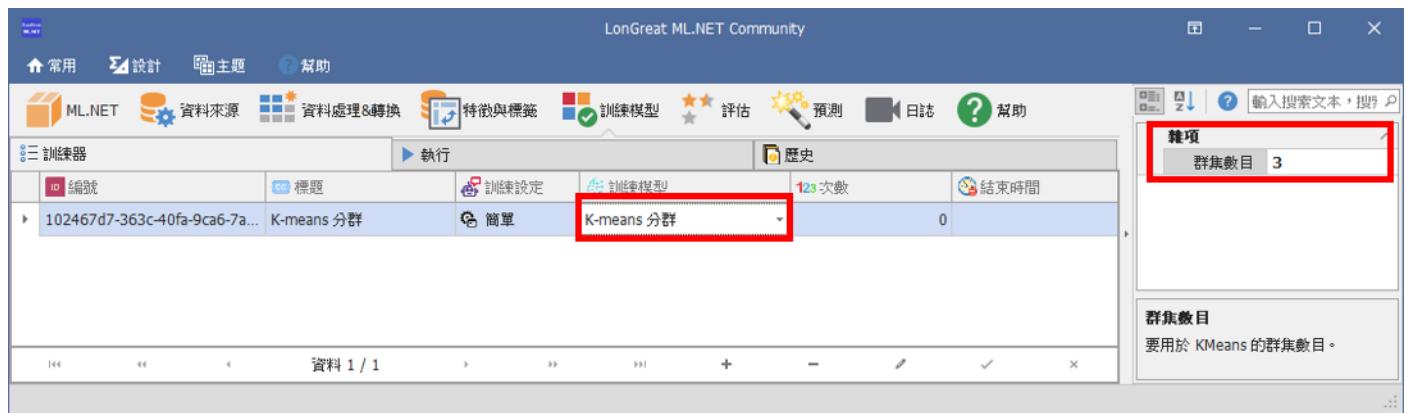
訓練模型

訓練器

按+新增

選擇 K-means 分群(KMeans)

群集數目改為 3



執行

1. 選擇訓練模型
2. 要顯示最多筆數
3. 執行

結果多三個欄位

- 特徵 Features
- 預測標籤 PredictedLabel
- 分數 Score

Sepal Length	Sepal Width	Petal Length	Petal Width	Label	Features	Predicted Label	Score
5.1	3.5	1.4	0.2	Iris-setosa	[5.1,3.5,1...]	3	[11.64523...]
4.9	3	1.4	0.2	Iris-setosa	[4.9,3.0,1...]	3	[11.48966...]
4.7	3.2	1.3	0.2	Iris-setosa	[4.7,3.2,1...]	3	[12.67441...]
4.6	3.1	1.5	0.2	Iris-setosa	[4.6,3.1,1...]	3	[11.64392...]
5	3.6	1.4	0.2	Iris-setosa	[5.0,3.6,1...]	3	[11.97375...]
5.4	3.9	1.7	0.4	Iris-setosa	[5.4,3.9,1...]	3	[9.875401...]
4.6	3.4	1.4	0.3	Iris-setosa	[4.6,3.4,1...]	3	[12.30015...]
5	3.4	1.5	0.2	Iris-setosa	[5.0,3.4,1...]	3	[11.08245...]
4.4	2.9	1.4	0.2	Iris-setosa	[4.4,2.9,1...]	3	[12.68146...]
4.9	3.1	1.5	0.1	Iris-setosa	[4.9,3.1,1...]	3	[11.22063...]
5.4	3.7	1.5	0.2	Iris-setosa	[5.4,3.7,1...]	3	[11.02097...]

評估

- A. 選取以訓練完合身的模型
- B. 選取設定資料須與原來相同
- C. 點擊預測按鈕=>產生出批次預測結果
- D. 點擊評估按鈕=>評估結果

資料	
平均距離	0.5263009389241536
標準化互資訊	非數值
戴維斯布爾丁指數	0.6663912498680068

平均距離

平均分。對於 K-Means 算法，'score' 是從質心到示例的距離。因此，平均分數是示例與聚類質心的接近程度的量度。換句話說，它是“聚類緊密度”的量度。但是請注意，該指標僅在集群數量增加時才會減少，並且在極端情況下（每個不同的示例都是其自己的集群），它將等於零。

The screenshot shows the ML.NET Studio interface with the 'Evaluation' tab selected. The main workspace displays a grid of data points from the Iris dataset, categorized into three clusters (Iris-setosa). The columns represent Sepal Length, Sepal Width, Petal Length, Petal Width, Label (Iris-setosa), Features, Predicted Label, and Score. The sidebar on the left lists the features used in the model: SepalLength (Single), SepalWidth (Single), PetalLength (Single), PetalWidth (Single), Label (String), Features (Single[2]), PredictedLabel (UInt32), and Score (Single[2]). The top-right panel shows the following summary statistics:

平均距離	0.5263009389241536
標準化互資訊	非數值
戴維斯布爾丁指數	0.6663912498680068

推薦的電影(矩陣分解模型)

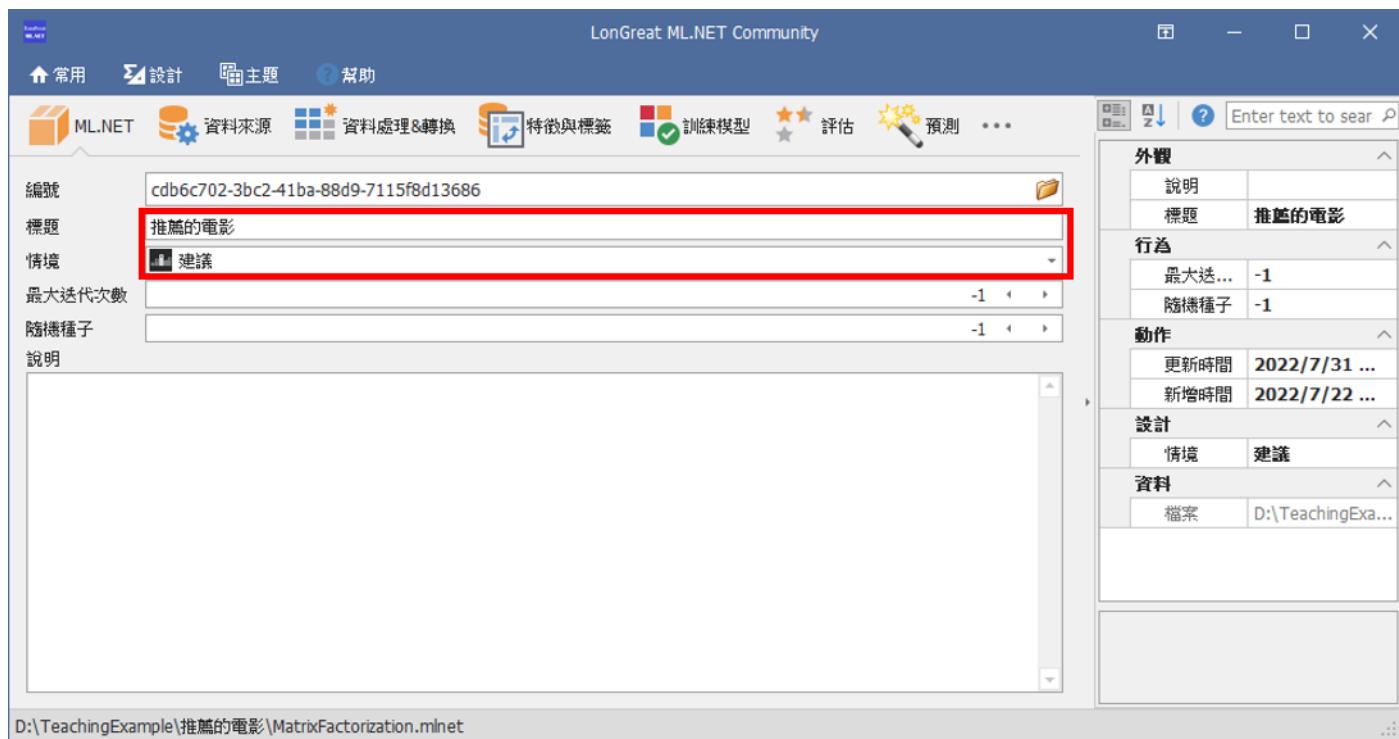
根據先前使用者的評分產生電影建議

選取適當的機器學習工作

有數種方式可以解決推薦問題，例如推薦電影清單或推薦相關產品清單；但在此情況下，您將預測使用者會給予該電影的評等 (1-5)，如果特定電影的預測評等高於所定義閾值，即推薦該電影（評等愈高，使用者喜歡特定電影的可能性愈高）。

建立專案

1. 輸入標題 => 推薦的電影
2. 選擇情境 => 建議



載入您的資料

以下是 *.csv 檔案中資料的預覽：

userId	movieId	rating	timestamp
1	1	4	964982703
1	3	4	964981247
1	6	4	964982224
1	47	5	964983815
1	50	5	964982931
1	70	3	964982400
1	101	5	964980868
1	110	4	964982176
1	151	5	964984041

在 *.csv 檔案中，有四個數據行：

- userId
- movieId
- rating
- timestamp

在機器學習服務中，用來進行預測的資料行稱為 特徵，而傳回預測的資料行稱為 標籤。

您希望預測電影評等，因此評等資料行是 Label。 其他三個資料行 userId、movieId 和 timestamp 都是 Features，用來預測 Label。

功能

userId

movieId

timestamp

標籤

rating

由您決定使用哪些 Features 來預測 Label。 您也可以使用 排列特徵重要性 等方法，協助選取最佳 Features。

在此情況下，您應該排除 timestamp 資料行為 Feature，因為時間戳記並不會實際影響使用者對特定影片的評分方式，因此無法提供更精確的預測：

功能

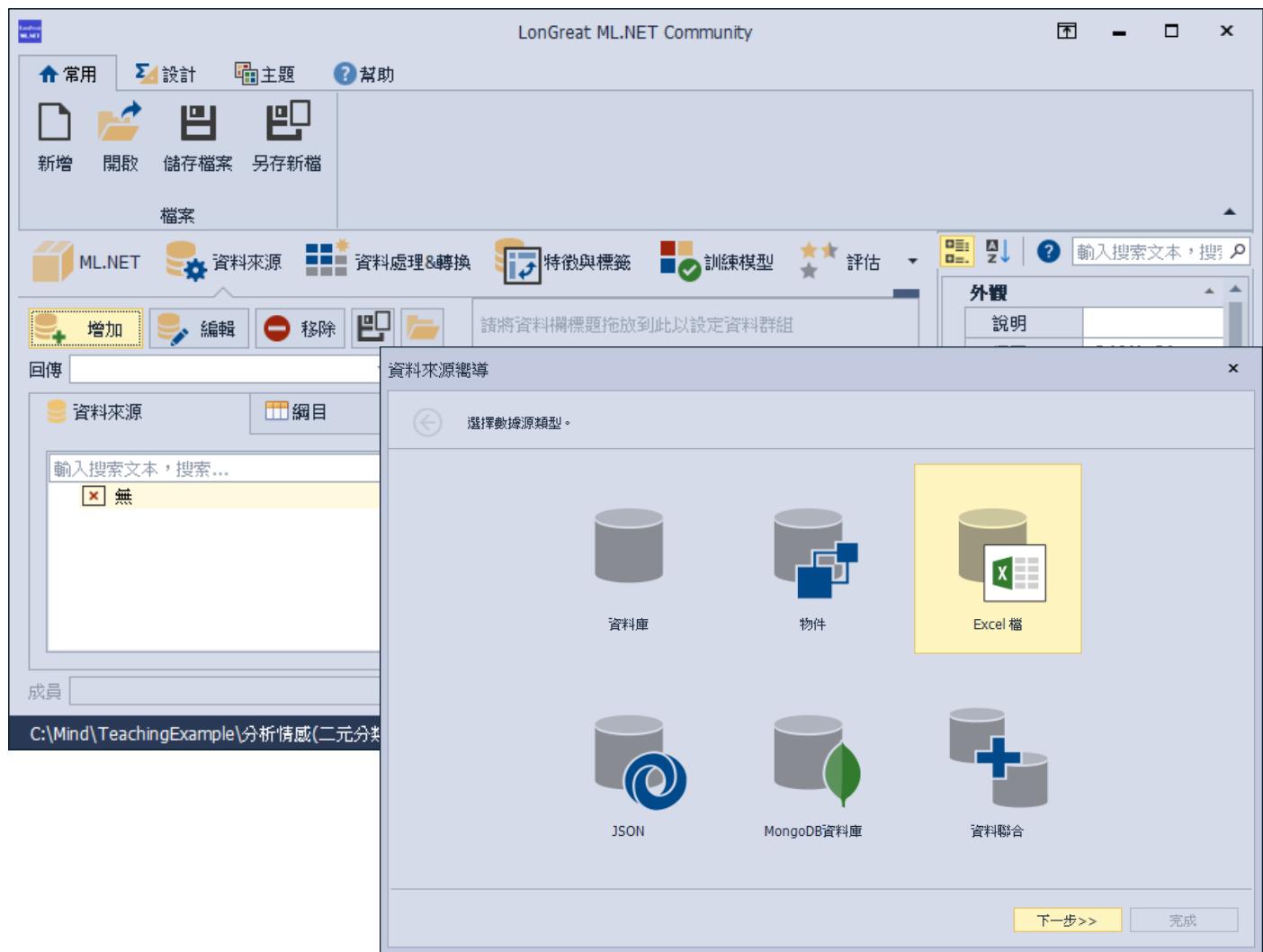
userId

標籤

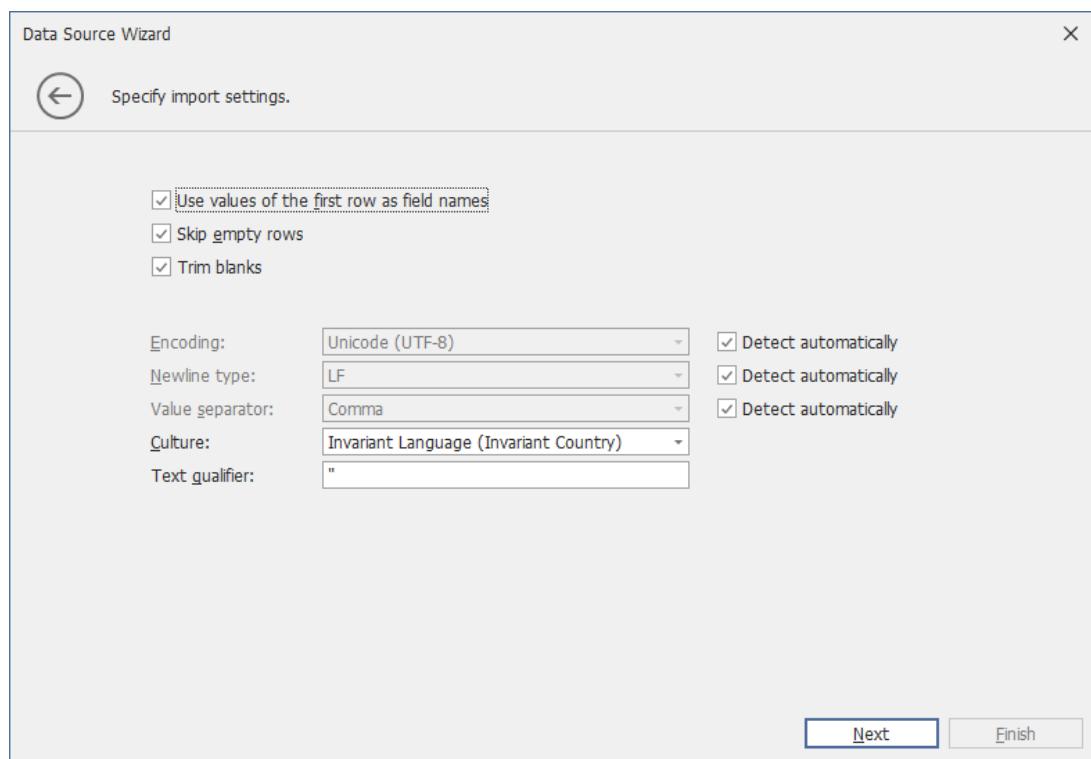
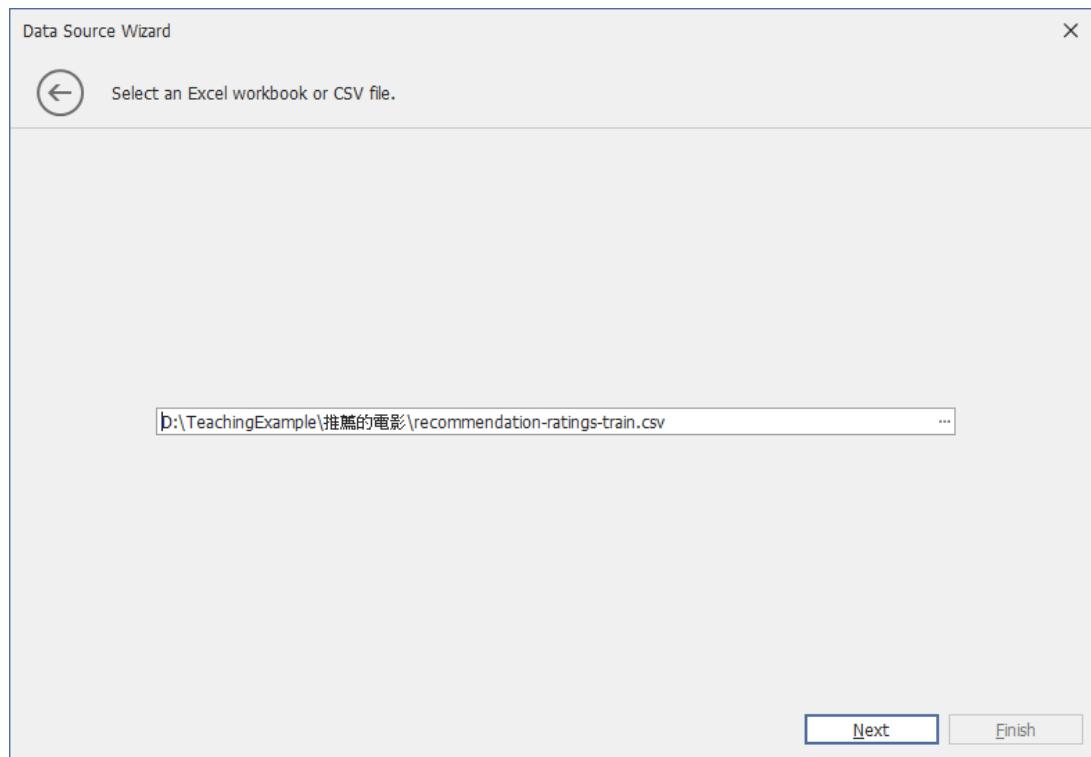
rating

movieId

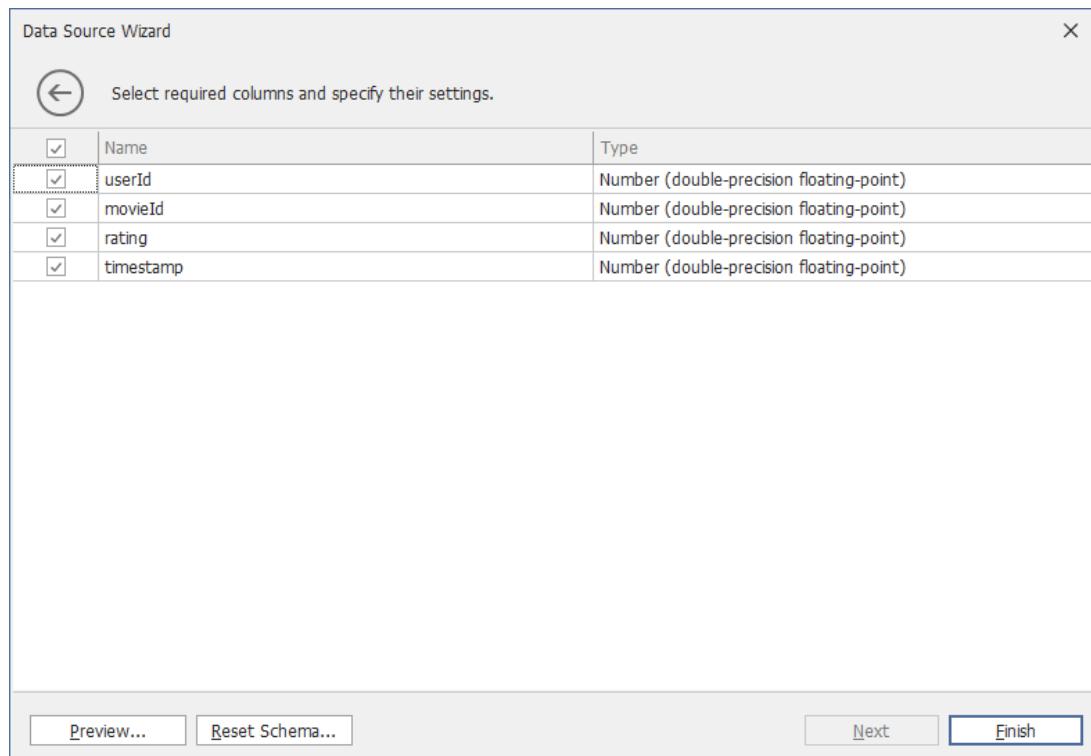
資料來源->增加->Excel 檔



選取 recommendation-ratings-train.csv



按完成



檢視資料

The screenshot shows the Visual Studio IDE with the 'ML.NET' tab selected. The 'Data Explorer' window displays a table named 'excelDataSource1' with columns 'user Id', 'movie Id', 'rating', and 'timestamp'. The table contains 99979 rows of data. To the right, the 'Properties' window is open for the 'excelDataSource1' item, showing details like '說明' (Description) set to '推薦的電影', '行為' (Behavior) set to '-1', and '動作' (Action) set to '2022/7/31 ...'.

資料轉換

由於 userId 和 movieId 代表使用者與電影標題，而非真正的值，所以您會使用 [MapValueToKey\(\)](#) 方法來將每個 userId 和每個 movieId 轉換成數值索引鍵類型 Feature 資料行（推薦演算法所接受的格式），並將其新增為新的資料集資料行：

userId	movieId	標籤	userIdEncoded	movieIdEncoded
1	1	4	userKey1	movieKey1
1	3	4	userKey1	movieKey2
1	6	4	userKey1	movieKey3

新增資料轉換

1. 類別轉換建 => 將 userId 轉換為鍵值

2. 類別轉換建 => 將 movieId 轉換為鍵值

結果檢視

I. 至檢視資料頁籤

II. 選擇顯示筆數

III. 按充滿資料

特徵與標籤

拖拉 rating=>標籤

拖拉 userIdEncoded=>欄列

拖拉 movieIdEncoded=>筆行

訓練模型

訓練器

矩陣分解模型是您的推薦定型演算法。當您擁有使用者過去如何評等產品的資料時，[矩陣分解](#)是推薦的常見方法，此亦為本教學課程資料集的情況。當您有不同的可用資料時，也有其他推薦演算法（請參閱[其他推薦演算法](#)一節以深入瞭解）。

在此案例中，Matrix Factorization 演算法使用的方法稱為「共同篩選」，此方法假設如果使用者 1 與使用者 2 對特定問題具有相同的意見，則使用者 1 對其他問題的想法較可能與使用者 2 相同。

比方說，如果使用者 1 對電影的評分與使用者 2 類似，則使用者 2 較可能享受使用者 1 已觀看並給予高度評分的電影：

	Incredibles 2 (2018)	The Avengers (2012)	Guardians of the Galaxy (2014)
使用者 1	已觀看及已按讚的電影	已觀看及已按讚的電影	已觀看及已按讚的電影
使用者 2	已觀看及已按讚的電影	已觀看及已按讚的電影	尚未觀看 -- 推薦電影

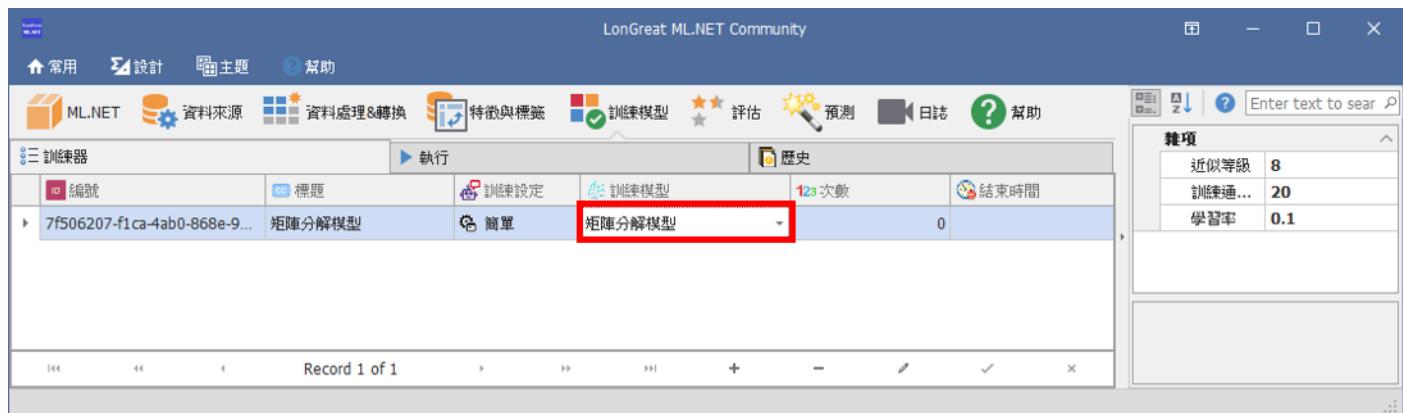
您可以在以下演算法參數

NumberOfIterations = 20,

ApproximationRank = 100

按+新增

選擇矩陣分解模型(MatrixFactorization)



執行

1. 選擇訓練模型
2. 要顯示最多筆數

3. 執行

結果多一個欄位

分數 Score

The screenshot shows the ML.NET Studio interface with the following details:

- Top Bar:** Includes tabs for 常用 (常用), 設計 (Design), 主題 (Theme), 幫助 (Help), and several ML.NET specific tabs: ML.NET, 資料來源 (Data Source), 資料處理&轉換 (Data Processing & Transformation), 特徵與標籤 (Features & Labels), 訓練模型 (Train Model), 評估 (Evaluate), 預測 (Predict), 日誌 (Logs), and a search bar.
- Left Sidebar:** Shows the "訓練器" (Trainer) section with the "矩陣分解模型" (Matrix Factorization Model) selected. It lists various model parameters and their types: userId (Single), movieId (Single), rating (Single), timestamp (Single), userIdEncoded (UInt32), movieIdEncoded (UInt32), and Score (Single).
- Central Area:** Displays the "執行" (Execute) results. A table titled "網格顯示" (Grid View) shows 10000 records of data. The columns are: user Id, movie Id, rating, timestamp, user Id En..., movie Id ..., and Score. The data includes various user and movie IDs, ratings (e.g., 1, 3, 6, 47, 50, 70, 101, 110, 151, 157, 163, 216, 223, 231, 235, 260, 296), timestamps (e.g., 964982700, 964981250, 964982200, 964983800, 964982900, 964982400, 964980860, 964982140, 964984060, 964984100, 964983700, 964981200, 964981000, 964981200, 964980900, 964981700, 964983000), and scores (e.g., 4.6904664, 3.840612, 4.5491705, 4.7377234, 4.978429, 3.981197, 4.4167533, 4.7335863, 4.29742, 3.833571, 4.2958946, 4.0183554, 4.456947, 3.6249235, 4.1606374, 5.101451, 4.914312).
- Right Sidebar:** Contains a "雜項" (Miscellaneous) panel with settings: 近似等級 (Approximation Level) set to 100, 訓練通... (Training...) set to 20, and 學習率 (Learning Rate) set to 0.1.

其他建議演算法

具備共同篩選的矩陣分解演算法，僅為執行電影推薦的其中一種方法。在許多情況下，您可能會沒有可用的評等資料，並只有使用者的電影觀看記錄。而在其他情況下，您擁有的資料可能不只是使用者評等資料。

演算法	案例
單一類別 矩陣分解	當您只需要 userId 和 movieId 時，請使用此選項。此推薦類型乃根據共同採購案例或經常同時購買的產品，也就是會根據客戶自己的採購訂單記錄向客戶推薦一組產品。
欄位感知 分解機器	當您所擁有的功能多於 userId、productId 和評等（如產品描述或產品價格）時，請使用此選項來進行推薦。此方法也會使用共同作業篩選方法。

評估

- A. 選取以訓練完合身的模型
- B. 選取設定資料須與原來相同
- C. 點擊預測按鈕=>產生出批次預測結果
- D. 點擊評估按鈕=>評估結果

資料

平均絕對誤差(MAE)	0.4829922852118623
判定係數	0.6410528044739426
均方誤差(MSE)	0.3913956413984396
根均方誤差(RMSE)	0.6256162093475837
損失函數	0.3913956413638045

平均絕對誤差(MAE)

絕對損失定義為 $L1 = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i|$, 其中 m 是測試集中的實例數, \hat{y}_i 是每個實例的預測標籤, 和 y_i 是每個實例的正確標籤。L1 損失是一個非負的遞減度量。較小的值表明該指標的模型更好。

LonGreat ML.NET Community

評估

訓練合身 2022-07-31-21-40-58 矩陣分解模型

平均絕對誤差(MAE)

絕對損失定義為 $L1 = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i|$, 其中 m 是測試集中的實例數, \hat{y}_i 是每個實例的預測標籤, 和 y_i 是每個實例的正確標籤。L1 損失是一個非負的遞減度量。較小的值表明該指標的模型更好。

花瓣識(影像分類)

示範如何使用定型現有的 TensorFlow 模型，以建立自訂影像分類器。

影像分類傳輸學習範例概觀

可使用預先定型的深度學習 TensorFlow 模型來分類影像。

瞭解問題

影像分類是電腦視覺問題。 影像分類會採用影像作為輸入，並將其分類為指定的類別。 影像分類模型通常會使用深度學習和神經網路來定型。 如需詳細資訊，[請參閱深度學習與機器學習](#)。

影像分類很有用的一些案例包括：

- 臉部辨識
- 情緒偵測
- 醫療診斷
- 地標偵測

本教學課程會訓練自訂影像分類模型，以執行花品總辨識的自動化視覺檢查，以識別因破解而損毀的結構。

ML.NET 影像分類 API

ML.NET 提供各種執行影像分類的方式。 本教學課程會使用影像分類 API 來套用傳輸學習。 影像分類 API 會使用 [TensorFlow.NET](#)，這是提供 TensorFlow C++ API C# 低階程式庫。

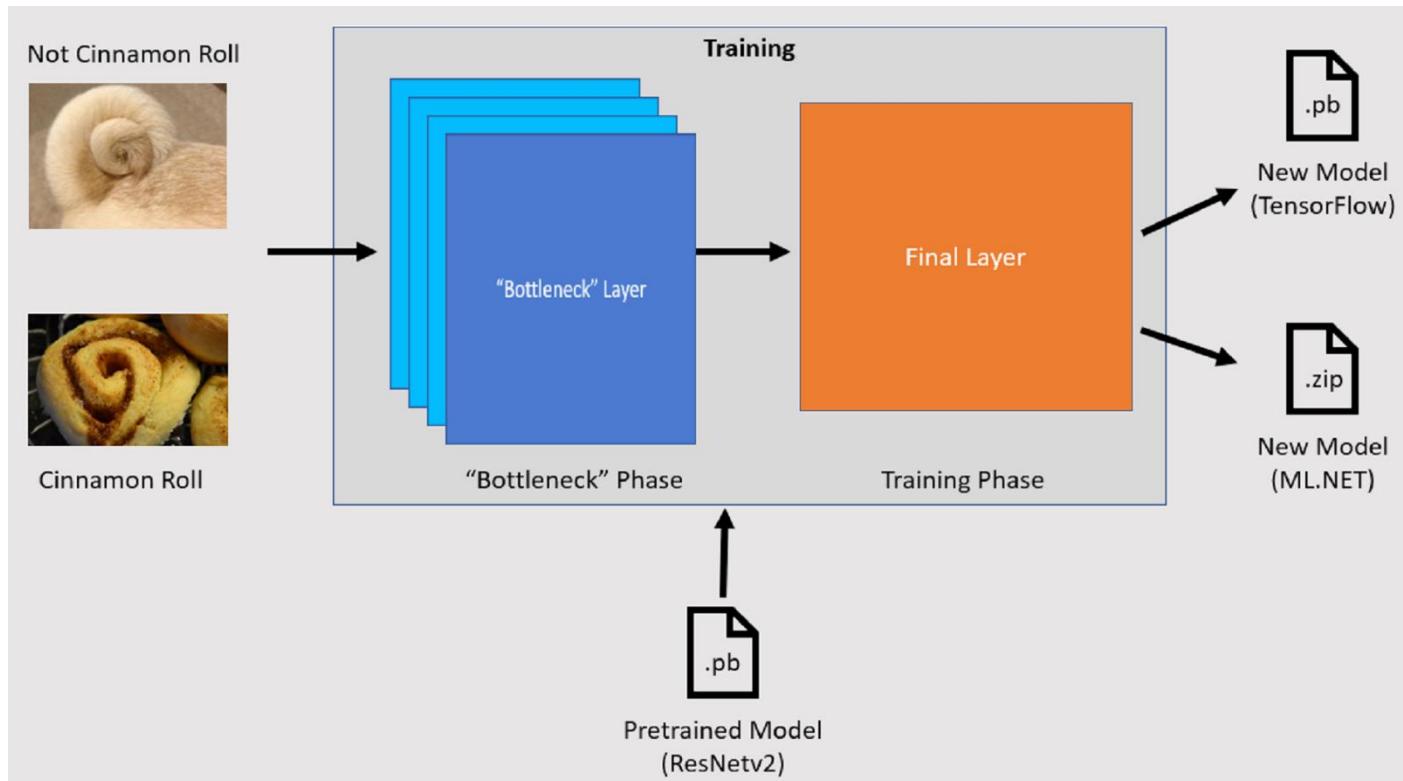
什麼是傳輸學習？

轉移學習會套用從解決一個問題到另一個相關問題所得到的知識。從頭開始定型深度學習模型需要設定數個參數、大量加上標籤的定型資料，以及大量的計算資源，（數百個 GPU 時數）。 使用預先定型的模型以及傳輸學習可讓您將定型程式快捷方式。

訓練程式

影像分類 API 會載入預先定型的 TensorFlow 模型，以啟動定型程式。 定型套裝程式含兩個步驟：

1. 瓶頸階段
2. 訓練階段



瓶頸階段

在瓶頸階段，會載入定型影像集，並將圖元值當做預先定型模型的凍結層輸入或特徵使用。 凍結層包含神經網路中的所有層，最多到第二層，非正式稱為瓶頸層。 這些層稱為凍結，因為這些層上不會發生任何定型，而且作業是傳遞的。 其位於這些凍結層中，可協助模型區分不同類別的較低層級模式。 層數目愈大，此步驟的計算密集程度就越高。 幸運的是，由於這是一次性計算，因此可以在試驗不同的參數時快取結果，並在稍後執行時使用。

訓練階段

一旦計算瓶頸階段的輸出值，就會用來作為輸入來重新定型模型的最後一層。 此程式是反復的，會針對模型參數所指定的次數執行。 在每次執行期間，都會評估遺失和精確度。 然後，會進行適當的

調整，以改善模型，目標是將損失降至最低，並將精確度最大化。定型完成後，會輸出兩種模型格式。其中一個是 .pb 模型的版本，另一個是 .zip 模型 ML.NET 序列化版本。在 ML.NET 支援的環境中工作時，建議使用 .zip 模型的版本。不過，在不支援 ML.NET 的環境中，您可以選擇使用 .pb 版本。

瞭解預先定型的模型

本教學課程中使用的預先定型模型是剩餘網路 (ResNet) v2 模型的 101 層變體。原始模型已定型，以將影像分類成千個類別。模型會接受大小為 224 x 224 的影像做為輸入，並輸出其定型的每個類別的類別機率。此模型的一部分是用來使用自訂影像來定型新的模型，以在兩個類別之間進行預測。

準備並瞭解資料



向日葵



玫瑰



蒲公英



雛菊



鬱金香



40411019_526f3fc8d9_m.jpg



40411100_7fbe10ec0f_n.jpg



44079668_34df ee3da1_n.jpg



45045003_30bb d0a142_m.jpg



45045005_5735 4ee844.jpg



50987813_7484 bfbcd9.jpg



58636535_bc53 ef0a21_m.jpg



127192624_afaa 3d9cb84.jpg



145303599_262 7e23815_n.jpg



147804446_ef9 244c8ce_m.jpg



151898652_b5f 1c70b98_n.jpg



164668737_aea b0cb55e_n.jpg



164670176_9f5 b9c7965.jpg



164670455_29d 8e02bbd_n.jpg



164671753_ab3 6d9cbb7_n.jpg



164672339_f2b 5b164f6.jpg



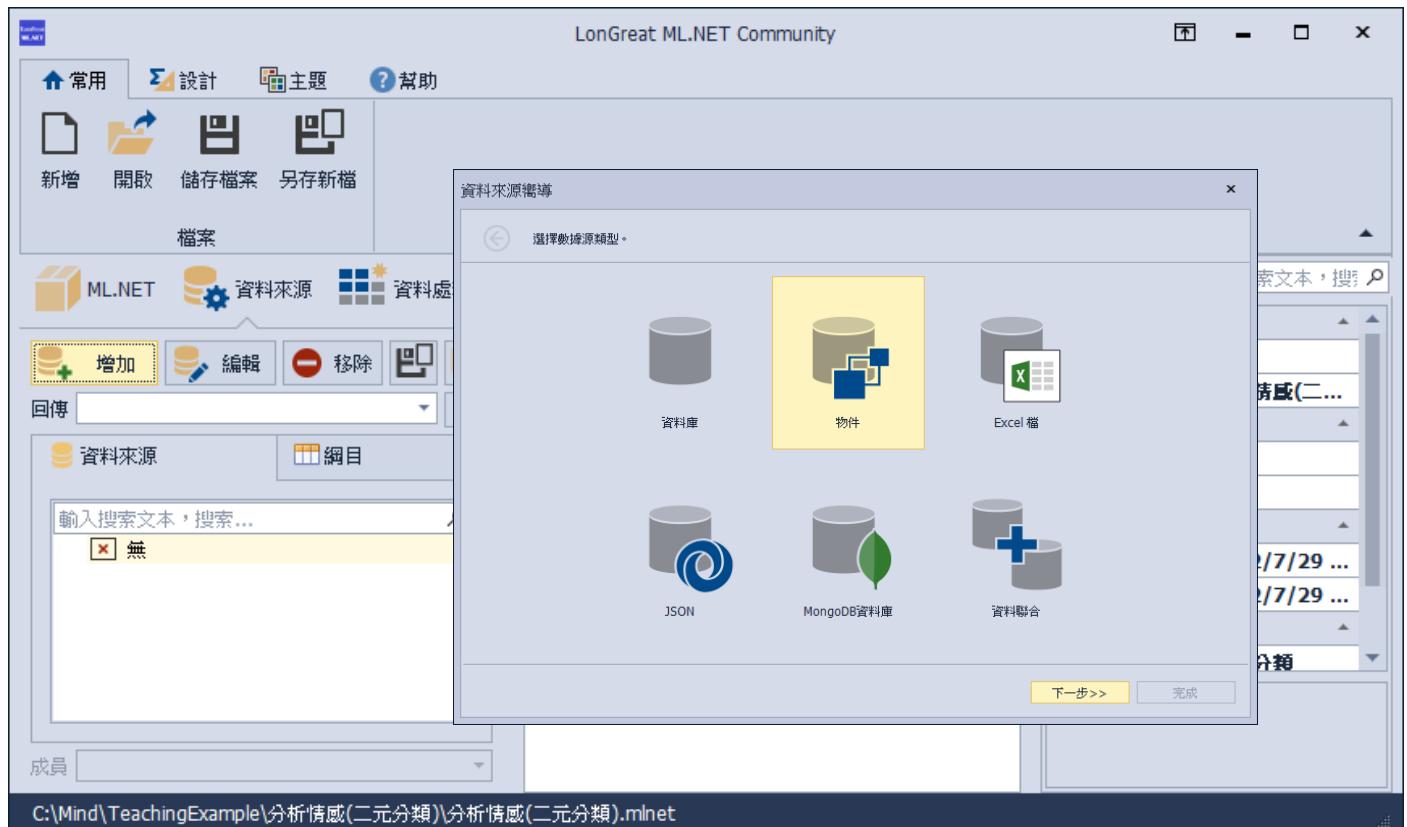
175638423_058 c07afb9.jpg



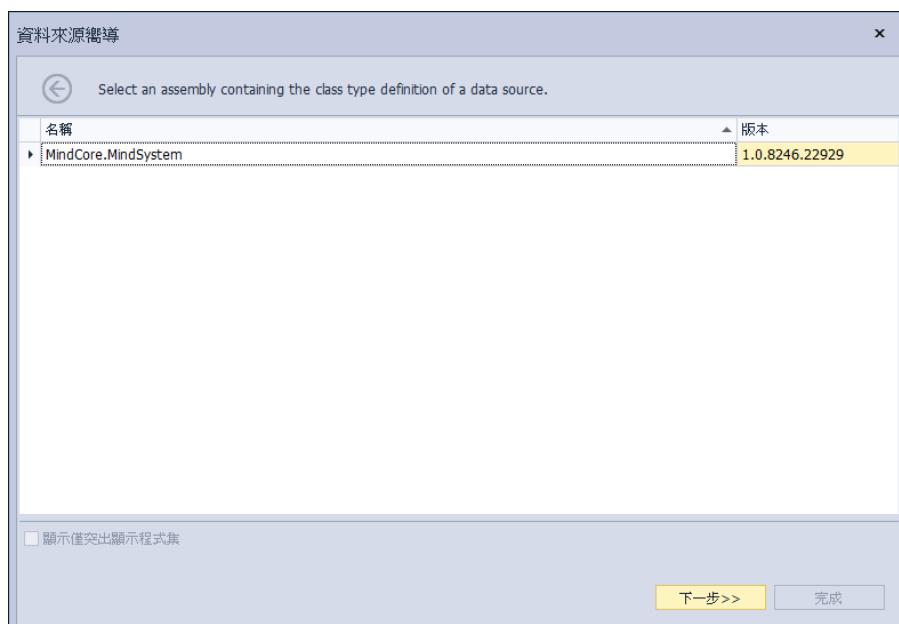
184682095_46f 8607278.jpg

載入資料

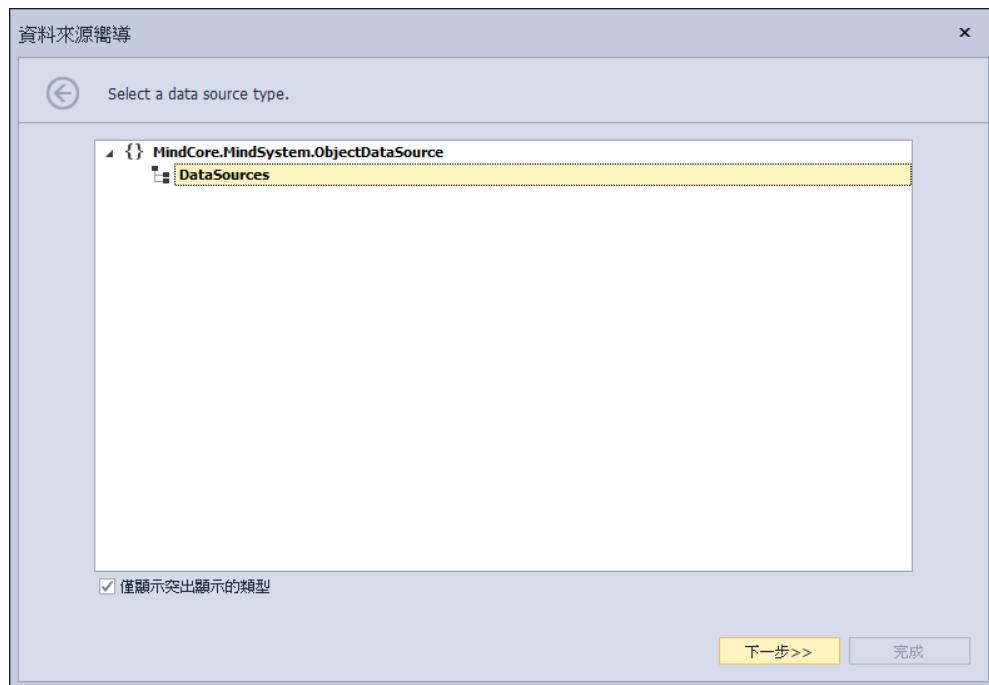
資料來源->增加->物件->下一步



選取 MindCore.MindSystem -> 下一步



選取 DataSources -> 下一步



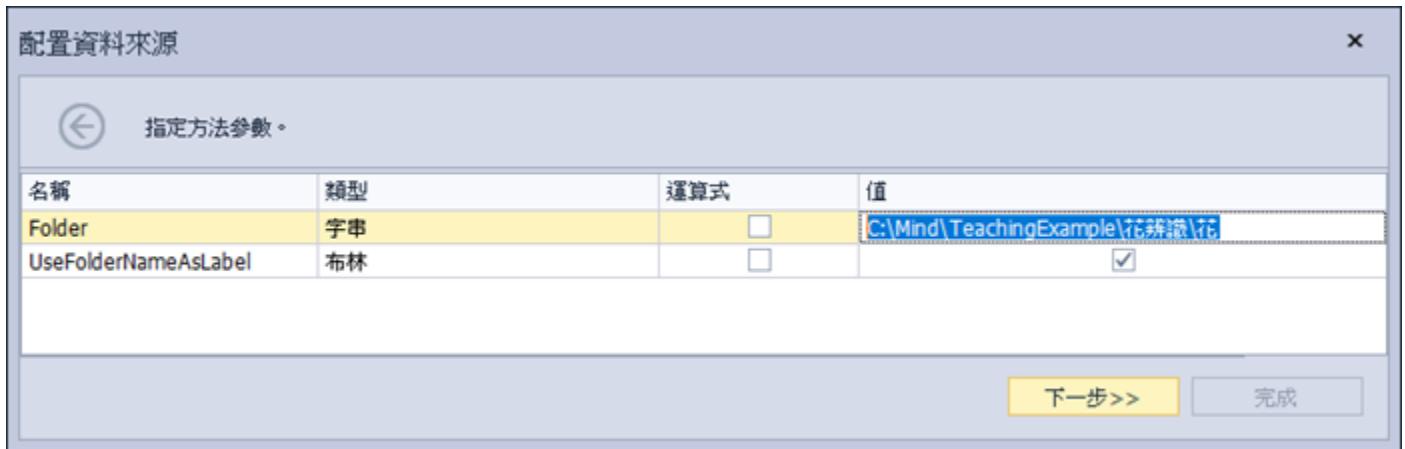
選取圖資料夾轉成資料表 -> 下一步



資料夾(Folder)值 = C:\Mind\TeachingExample\花辨識\花

使用文件夾名稱作為標籤(UseFolderNameAsLabel) = True

下一步 → 按完成



按檢視 → 可看出產生的資料表

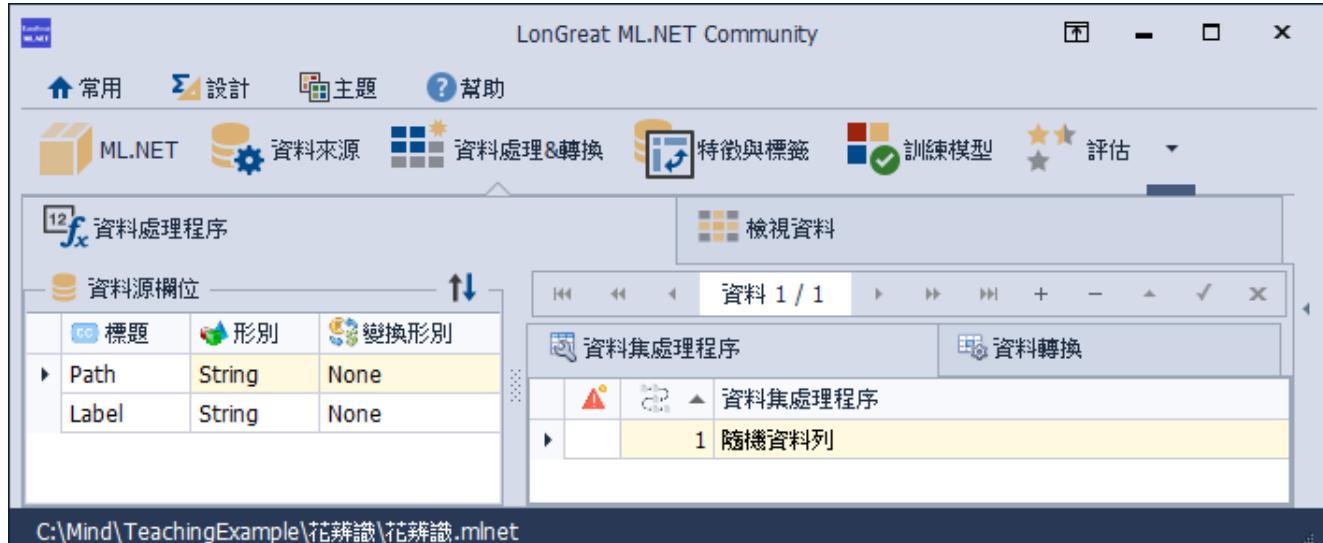
欄位 = 路徑與標籤

路徑	標籤
C:\Mind\TeachingExample\花辨識\花\向日葵\127192624_afab3d9cb84.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\145303599_262f7e23815_n.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\147804446_ef9244c8ce_m.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\151898652_b5f1c70b98_n.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\164668737_aeab0cb55e_n.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\164670176_9f5b9c7965.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\164670455_29d8e02bbd_n.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\164671753_ab36d9ccb7_n.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\164672339_f2b5b164f6.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\175638423_058c07afb9.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\184682095_46f8607278.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\184682320_73cf74710.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\184682506_8a9b8c662d.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\184682652_c927a49226_m.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\184682920_97ae41ce60_m.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\184683023_737fec5b18.jpg	向日葵
C:\Mind\TeachingExample\花辨識\花\向日葵\193874852_fb633d8d00_n.jpg	向日葵

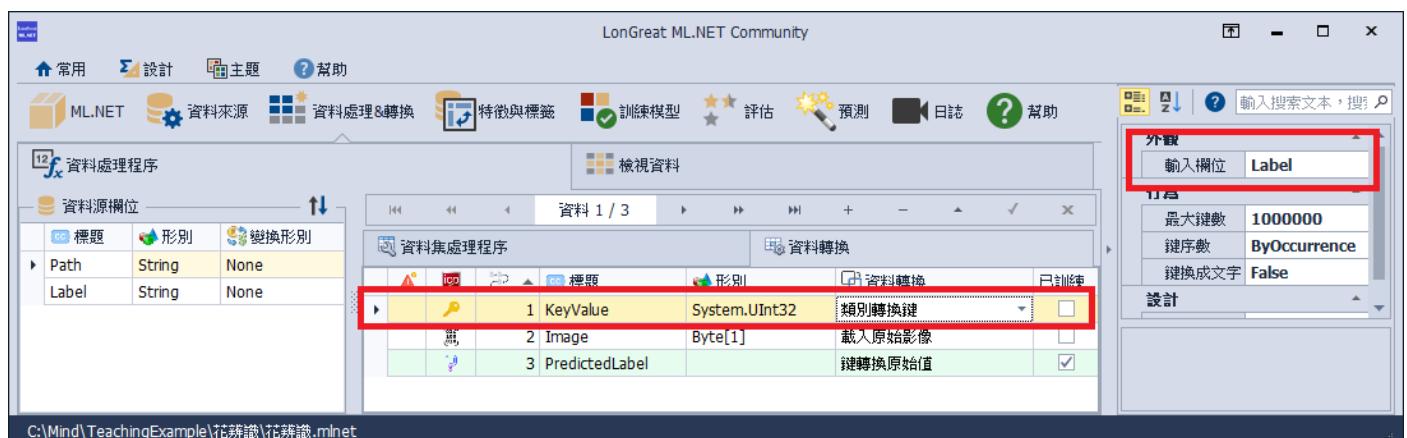
資料轉換

隨機資料列

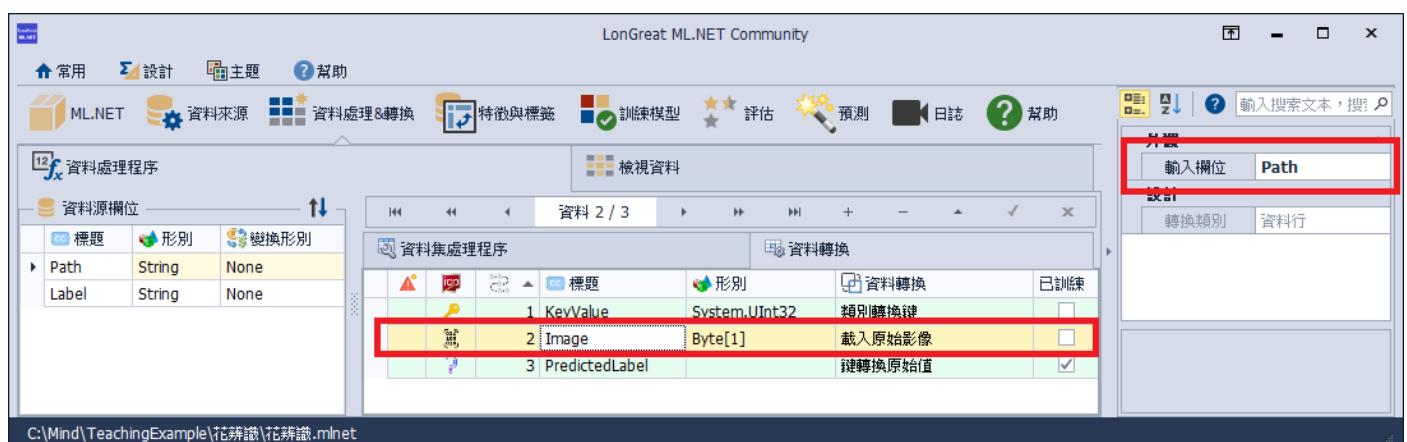
1. 將資料列隨機打散



2. 類別轉換鍵 => 將 Label 轉換為鍵值



3. 載入原始影像=>將 Path 欄位影像載入到 Image 欄位



4. 鍵轉換原始值 => 預測標籤(PredictedLabel)轉換回來原始值

結果檢視

I. 至檢視資料頁籤

II. 選擇顯示筆數

III. 按充滿資料

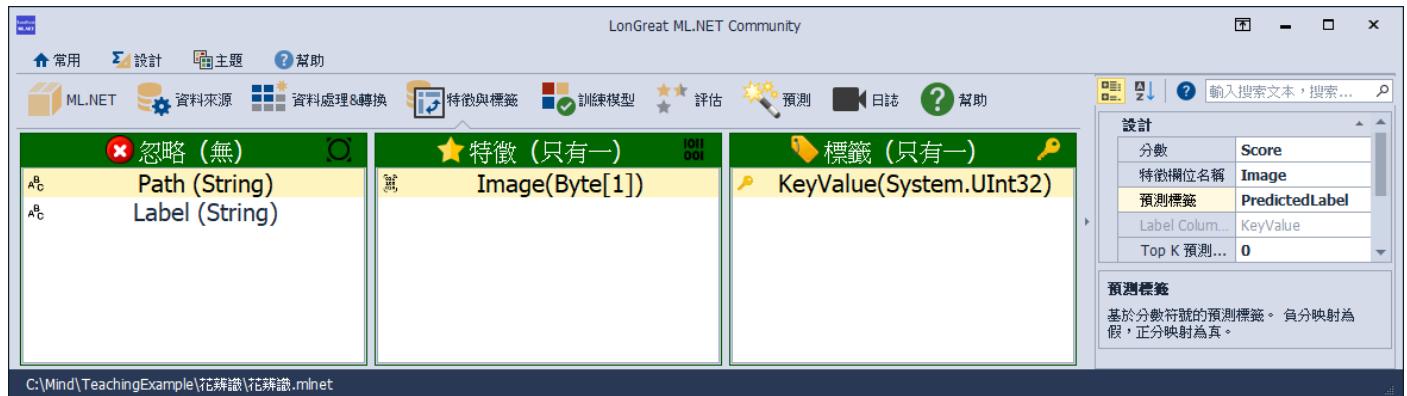
The screenshot shows the application's main window with several tabs at the top: 常用 (Common), 設計 (Design), 主題 (Theme), 幫助 (Help), 資料來源 (Data Source), 資料處理&轉換 (Data Processing & Transformation), 特徵與標籤 (Features & Labels), 訓練模型 (Training Model), 評估 (Evaluation), 預測 (Prediction), 日誌 (Log), and 幫助 (Help). The 'View Data' tab is selected.

In the center, there is a grid view showing data from a file named 'C:\Mind\TeachingExample\花卉識\花卉.mlnet'. The columns are Path (String), Label (String), KeyValue (UInt32), and Image (Byte[2]). The 'Label' column contains values like 向日葵, 雛菊, 玫瑰, 蒲公英, etc. The 'KeyValue' column contains UInt32 values. The 'Image' column displays thumbnail images of flowers. A specific row is selected, showing a yellow rose flower image. At the bottom right of the grid, there are '確定(O)' (Confirm) and '取消(C)' (Cancel) buttons.

特徵與標籤

拖拉 Image=>特徵

拖拉 KeyValue=>標籤



訓練模型

訓練器

按+新增

選擇 DNN 影像分類模型 (ImageClassification)



執行

1. 選擇訓練模型
2. 要顯示最多筆數
3. 執行

結果多二個欄位

- ✚ 預測標籤 PredictedLabel
- ✚ 分數 Score

The screenshot shows the LonGreat ML.NET Community application's user interface. The main window title is "LonGreat ML.NET Community". The top menu bar includes "常用" (Common), "設計" (Design), "主題" (Theme), and "幫助" (Help). Below the menu is a toolbar with icons for "新增" (New), "開啟" (Open), "儲存檔案" (Save File), "另存新檔" (Save As), "檔案" (File), "ML.NET", "資料來源" (Data Source), "資料處理&轉換" (Data Processing & Transformation), "特徵與標籤" (Features & Labels), "訓練模型" (Training Model), "評估" (Evaluation), "預測" (Prediction), "日誌" (Log), and "歷史" (History). The central workspace displays a table titled "MindCore.MINet.DataViews.ITableDataView" with the following columns: "Path (String)", "Label (String)", "KeyValue (UInt32)", "Image (Byte[2])", "Predicted Label (String)", and "Score (Single[2])". The table contains 200 rows of data, with the first few rows visible. The data includes various flower names like 蒲公英, 向日葵, 玫瑰, etc., along with their predicted labels and scores.

Path (String)	Label (String)	KeyValue (UInt32)	Image (Byte[2])	Predicted Label (String)	Score (Single[2])
C:\Mind\TeachingExample\花辨識\花\蒲公英\15987457_49dc11...	蒲公英	1	[Image]	蒲公英	[0.9997749,0...]
C:\Mind\TeachingExample\花辨識\花\向日葵\184682506_8a9b8...	向日葵	2	[Image]	向日葵	[0.01333415...]
C:\Mind\TeachingExample\花辨識\花\197011740_21825...	向日葵	2	[Image]	向日葵	[0.02435277...]
C:\Mind\TeachingExample\花辨識\花\玫瑰\18974357_0faa23c...	玫瑰	3	[Image]	玫瑰	[3.597713E-0...]
C:\Mind\TeachingExample\花辨識\花\雛菊\175106495_53ebdef...	雛菊	4	[Image]	雛菊	[0.00097999...]
C:\Mind\TeachingExample\花辨識\花\向日葵\193878348_43571...	向日葵	2	[Image]	向日葵	[0.04289465...]
C:\Mind\TeachingExample\花辨識\花\蒲公英\140951103_69847...	蒲公英	1	[Image]	蒲公英	[0.937225,0...]
C:\Mind\TeachingExample\花辨識\花\雛菊\173350276_02817aa...	雛菊	4	[Image]	雛菊	[0.02567177...]
C:\Mind\TeachingExample\花辨識\花\向日葵\127192624_afad3...	向日葵	2	[Image]	向日葵	[0.00162402...]
C:\Mind\TeachingExample\花辨識\花\玫瑰\295257304_de893fc...	玫瑰	3	[Image]	玫瑰	[0.00025254...]
C:\Mind\TeachingExample\花辨識\花\鬱金香\132538273_33524...	鬱金香	5	[Image]	鬱金香	[0.01251243...]
C:\Mind\TeachingExample\花辨識\花\玫瑰\410425647_458666...	玫瑰	3	[Image]	玫瑰	[6.996205E-0...]
C:\Mind\TeachingExample\花辨識\花\向日葵\200011914_q3f57	向日葵	2	[Image]	向日葵	[0.00578804]

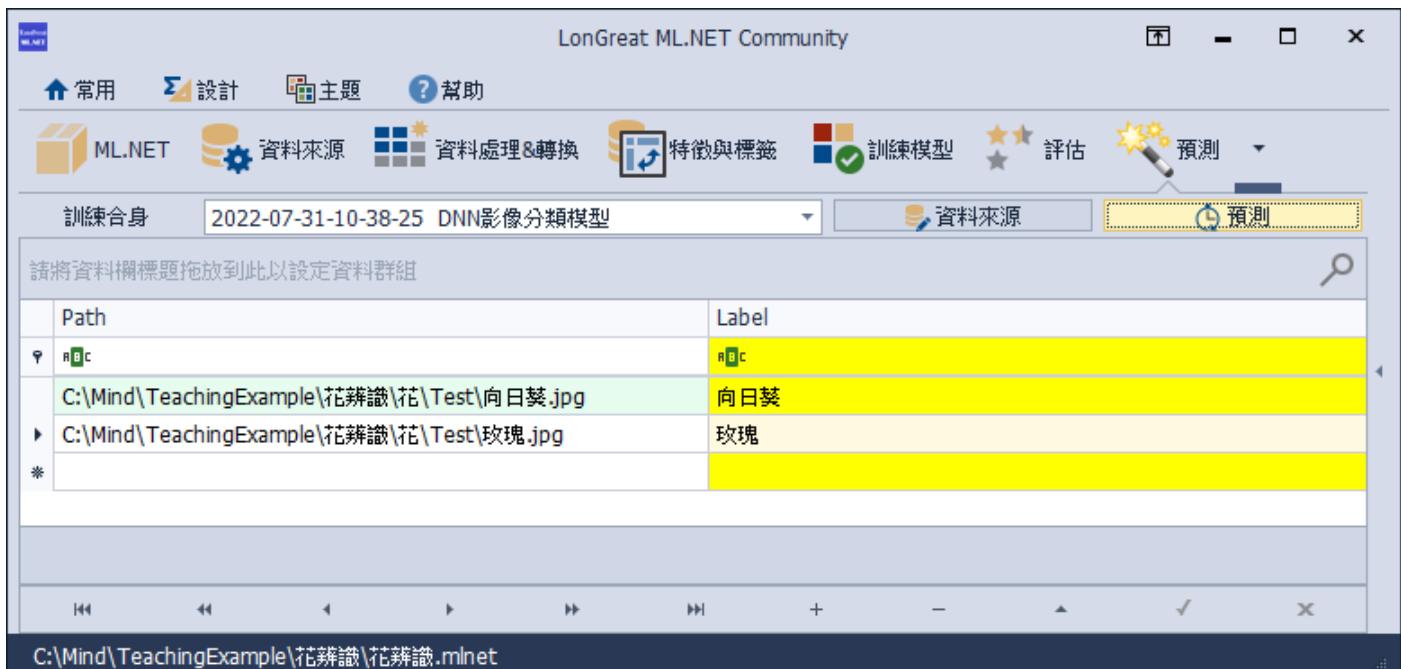
評估

- A. 選取以訓練完合身的模型
- B. 選取設定資料須與原來相同
- C. 點擊預測按鈕=>產生出批次預測結果
- D. 點擊評估按鈕=>評估結果

資料	
宏觀準確率	0.99
每類對數損失	(Collection)
所有 K 的 Top-K 準確度	(Collection)
混淆矩陣	MindCore.MLNet.Catalogs.Re...
微觀準確率	0.99
對數損失	0.081601647059613
對數損失減少	0.9492980459642588
Top-K準確率	0
Top-K預測計數	0

預測

1. 選取以訓練完合身的模型
2. 點擊資料來源按鈕=>產生資料表
3. 輸入資料表 Path 欄位
4. 點擊預測按鈕
5. 產生出 Label 預測結果



相關連結

 GitHub	github.longgreat.net
 gitee	gitee.longgreat.net