推薦序

為響應企業智慧化管理巨量資料的需求，協助企業克服同時管理結構化與非結構化資料、並整合異質資料庫的挑戰，微軟於 2019 年底正式推出 SQL Server 2019，以 AI 整合巨量資料叢集全新架構、領先業界的效能與安全，幫助企業與開發人員解決現今資料工作環境的困難。

SQL Sever 2019 透過資料虛擬化 Polybase 能直接查詢外部資料源，協助企業高效打破資料孤島，解決無法整合異質資料庫的痛點。全新架構巨量資料叢集，透過 Apache Spark™ 與 HDFS 的整合，將能輕鬆在 Kubernetes 叢集上部署容器，同時管理結構與非結構化資料，進而從大量資料中獲得決策先機。

本書完美結合資料庫理論與實務，使用大量實作範例來說明資料庫系統理論、實體關聯模型和正規化，與資料庫設計，讀者不僅可以實際在資料庫設計工具繪製專案的實體關聯圖，更可將設計成果建立成 SQL Server 資料庫，來驗證實體關聯模型的資料庫設計理論。本書完整說明 T-SQL 語法、預存程序、順序物件、自訂函數、觸發程序、資料指標和交易處理，幫助讀者精通 T-SQL 程式設計，更有實例說明如何使用 C#和 LINQ 建立用戶端程式。

陳會安老師多年扎實功力在本書中體現，詳盡介紹了 T-SQL 語法，相信除了對學校資料庫相關課程是一本很好的教材之外，對資料庫開發人員來說更是一本深入淺出的好工具書。我想推薦這本書給您—身處於海量資料快速迭代的現今，與時俱進地熟稔資料平台已經成為資料庫開發人員職涯上和自我進修不可取代的技能。歡迎您一起加入 SQL Server 資料庫平台的這個大家庭！

台灣微軟產品行銷資深經理

廖育萱 Sherry Liao

Chapter

19

SQL Server 機器學習服務

19-1 認識 SQL Server 機器學習服務

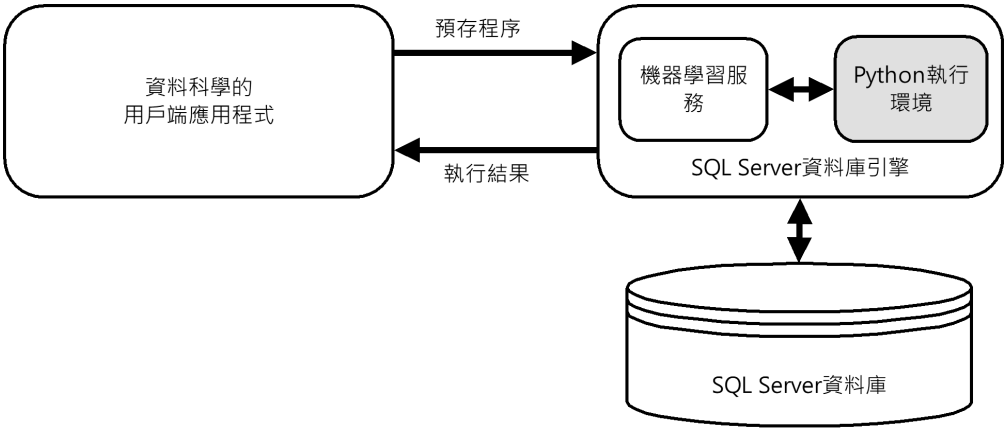
微軟 SQL Server 機器學習服務可以讓 SQL Server 支援資料科學（Data Science）所需的工作，直接讓我們在 SQL Server 訓練和儲存預測模型，在本章是使用 Python 語言來說明 SQL Server 機器學習服務。

Memo

本章測試的 SQL 指令是使用【教務系統】資料庫，請執行本書範例「Ch19\Ch19_School.sql」的 SQL 指令碼檔案來建立本章測試所需的資料庫、資料表和記錄資料。

認識機器學習服務

SQL Server 從 2017 版開始內建 Python 執行環境，可以讓 SQL Server 執行 Python 程式碼，換句話說，我們可以將相關資料科學運算直接在 SQL Server 中執行來增進執行效能，而不必將資料從資料庫拉出來，這就是 SQL Server 機器學習服務（SQL Server Machine Learning Services），如下圖所示：



上述 SQL Server 機器學習服務是 SQL Server 內建功能，提供一個 Python 執行環境（即 Anaconda），能夠使用資料表的關聯式資料來執行 Python 程式（透過預存程序），我們可以使用此服務來載入和清理資料、進行探索式資料分析和資料視覺化，然後直接在資料庫訓練機器學習的模型，和儲存訓練結果的預測模型。

微軟除了將常用 Python 套件安裝至 SQL Server 機器學習服務外，更提供微軟專屬的 Python 套件，如下表所示：

套件名稱	說明
revoscalepy	提供可攜式、可擴展和分散式 Python 函數，可以用來匯入、轉換和進行資料分析，使用在敘述統計學、廣義線性模型（Generalized Linear Model，GLM）、Logistic 迴歸、分類、迴歸樹（Regression Trees）和決策森林（Decision Forests）
microsoftml	微軟新增的機器學習演算法，可以建立模型來進行特徵擷取、文字、影像和情感分析

機器學習服務能作什麼

在 SQL Server 機器學習服務內建眾多機器學習和深度學習模型，我們可以直接在 SQL Server 使用關聯式資料來進行訓練和將模型儲存在資料庫，或將現有模型部署到機器學習服務後，直接使用關聯式資料來進行資料預測。

SQL Server 機器學習服務可以進行的預測類型，如下表所示：

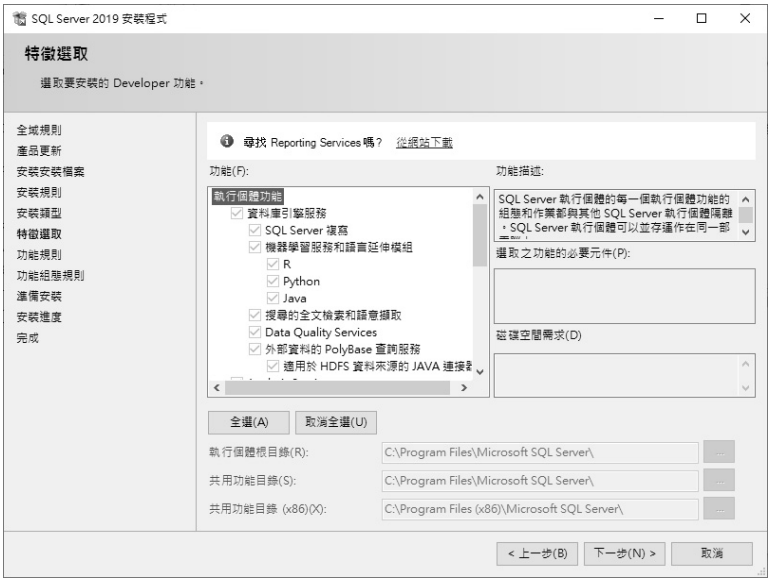
預測類型	說明
分類	將資料進行分類，例如：將客戶意見分為正面與負面；將圖片自動進行分類
迴歸分析	依據連續資料的迴歸分析進行預測，例如：根據地點和面積來預測房屋價格；年分和哩程數預測二手車車價
異常偵測	預測資料中的異常值，例如：偵測銀行詐騙交易
提供建議	預測資料中的關聯性，例如：根據顧客先前的購買記錄，建議在下一次消費時想要購買的產品

19-2 安裝與啟用 SQL Server 機器學習服務

在 SQL Server 使用機器學習服務之前，我們需要先安裝與啟用 SQL Server 機器學習服務。

安裝 SQL Server 的機器學習服務

在第 4 章安裝 SQL Server 時因為是勾選全部功能，所以預設已經安裝好 SQL Server 機器學習服務，我們可以在安裝程式看到安裝的元件，如下圖所示：



上述圖例如果沒有勾選【機器學習服務和語言延伸模組】下的【Python】功能，請勾選，按【下一步】鈕來安裝 SQL Server 機器學習服務。

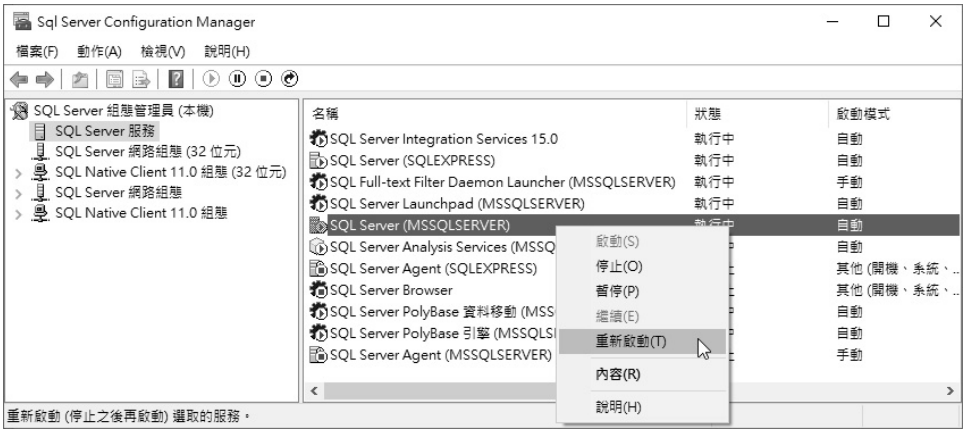
啟用 SQL Server 的機器學習服務

在確認已經安裝 SQL Server 機器學習服務後，我們只需啟用 SQL Server 執行外部 Python 腳本程式碼，就可以在 T-SQL 指令執行 Python 程式碼，其步驟如下所示：

1. 請啟動 Management Studio 連接安裝有機器學習服務的 SQL Server 資料庫引擎後，開啟和執行 Ch19_2.sql 指令碼檔案，如下所示：

```
EXEC sp_configure 'external scripts enabled', 1
RECONFIGURE WITH OVERRIDE
```

2. 然後啟動設定管理員，在 SQL Server 上執行【右】鍵快顯功能表的【重新啟動】指令，重新啟動 SQL Server 服務，如下圖所示：



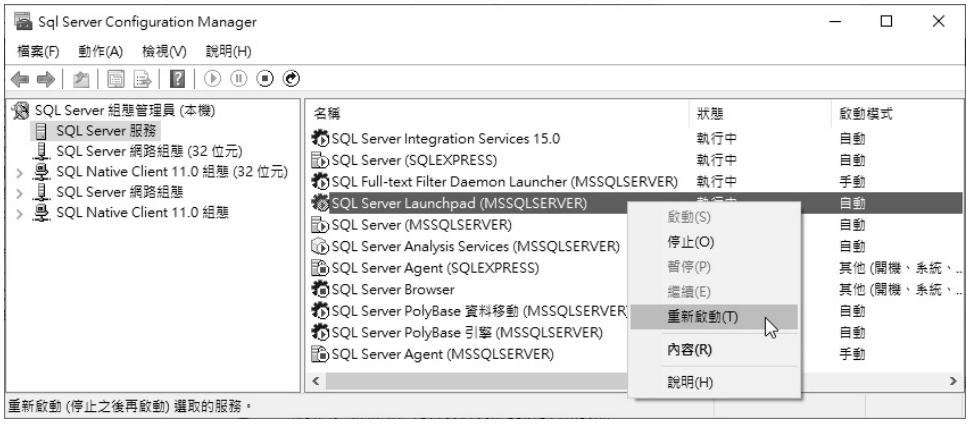
3. 在 Management Studio 開啟和執行 Ch19_2a.sql 指令碼檔案，可以看到 config_value 值已經改為 1，如下所示：

```
EXEC sp_configure 'external scripts enabled'
```

結果 訊息

	name	minimum	maximum	config_value	run_value
1	external scripts enabled	0	1	1	1

4. 最後，在設定管理員的 SQL Server Launchpad 上，執行【右】鍵快顯功能表的【重新啟動】指令來重新啟動服務，如下圖所示：

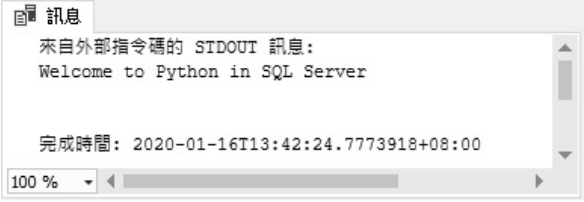


在 SQL Server 測試執行 Python 程式碼：Ch19_2b.sql

在成功安裝和啟用 SQL Server 機器學習服務後，我們可以建立 SQL 指令來測試在 SQL Server 執行 Python 程式碼，如下所示：

```
EXEC sp_execute_external_script
@language = N'Python',
@script = N'print("Welcome to Python in SQL Server")'
GO
```

上述 SQL 指令是執行 sp_execute_external_script 系統預存程序，執行的 Python 程式碼字串是呼叫 print()函數來輸出一段文字內容，其執行結果就是參數的訊息文字，如下圖所示：



19-3 在 SQL Server 執行 Python 程式碼

SQL Server 機器學習服務提供二種方式來執行 Python 程式碼，如下所示：

- 使用 sp_execute_external_script 系統預存程序來執行 Python 程式碼。
- 我們也可以使用 Python 開發工具撰寫 Python 程式，然後送至 SQL Server 機器學習服務來執行。

在本章主要說明如何使用 sp_execute_external_script 系統預存程序來執行 Python 程式碼，此預存程序的詳細語法說明請參閱第 19-4-1 節。

Python 變數與運算式：Ch19_3.py

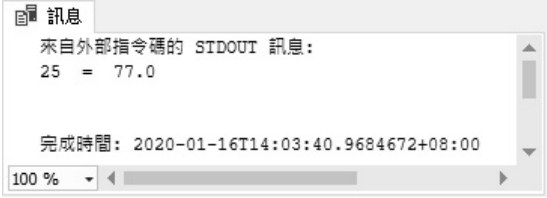
SQL Server 機器學習服務可以執行合法的 Python 程式碼字串，首先是 Python 變數與運算式，如下所示：

```
EXEC sp_execute_external_script
@language = N'Python',
@script = N'
cels = 25
fahr = cels * 9 /5 + 32
print(cels, " = ", fahr)
'
```

上述 SQL 指令使用系統預存程序 sp_execute_external_script 執行 Python 程式碼字串，其參數說明如下所示：

- @language 參數：指定外部腳本程式是使用 Python 語言。
- @script 參數：定義送至機器學習服務的腳本程式碼字串，使用的是 Unicode 字串 N' '，其字串內容就是合法的 Python 程式碼。

上述 Python 程式碼計算出攝氏 25 度轉換成的華氏溫度後，使用 print()函數輸出執行結果，如右圖所示：

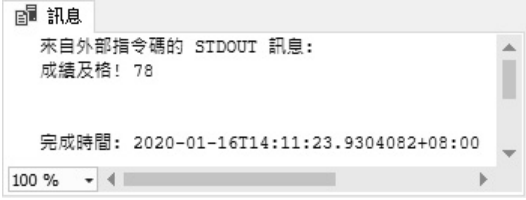


Python 條件判斷：Ch19_3a.py

Python 條件判斷的程式區塊需要在程式碼字串中縮排，例如：判斷成績變數 grade 值是否大於等於 60 分，如下所示：

```
EXEC sp_execute_external_script
@language = N'Python',
@script = N'
grade = 78
if grade >= 60:
    print("成績及格!", grade)
else:
    print("成績不及格!", grade)
'
```

上述 Python 程式碼使用 if/else 條件判斷 grade 變數的成績，可以看到 2 個縮排的程式區塊，其執行結果顯示成績及格和分數，如右圖所示：

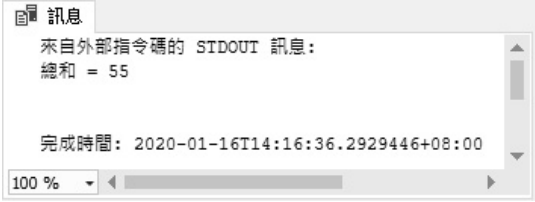


Python 迴圈結構：Ch19_3b.py

在 Python 使用迴圈計算 1 加到 10 的總和，如下所示：

```
EXEC sp_execute_external_script
@language = N'Python',
@script = N'
sum = 0
for i in range(11):
    sum = sum + i
print("總和 = " + str(sum))
'
```

上述 Python 程式碼使用 for/in 迴圈來計算 1 加到 10 的總和，其執行結果顯示總和是 55，如右圖所示：



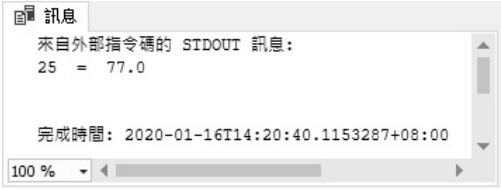
Python 函數：Ch19_3c.py

在 Python 程式碼改用 convert_to_f()函數來轉換攝氏成為華氏溫度，如下所示：

```
EXEC sp_execute_external_script
@language = N'Python',
@script = N'
def convert_to_f(c):
    f = (9.0 * c) / 5.0 + 32.0
    return f

cels = 25
fahr = convert_to_f(cels)
print(cels, " = ", fahr)
'
```

上述 Python 程式碼建立 convert_to_f()函數後，呼叫函數將攝氏 25 度轉換成華氏溫度後，顯示溫度的轉換結果，其執行結果如右圖所示：



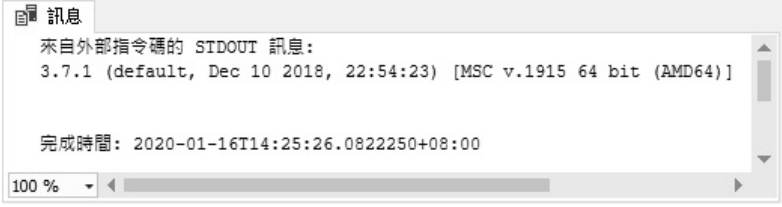
檢查 Python 的版本：Ch19_3d.py

Python 程式可以匯入 sys 模組來檢查 SQL Server 機器學習服務使用的 Python 版本，如下所示：

```
EXEC sp_execute_external_script
@language = N'Python',
@script = N'
import sys

print(sys.version)
'
```

上述 Python 程式碼使用 sys.version 屬性取得 Python 程式的版本 3.7.1 版，如下圖所示：



19-4 Python 程式碼的輸入與輸出參數

第 19-3 節我們是使用 Python 標準輸出的 print()函數來輸出執行結果，事實上，因為是在 SQL Server 執行 Python 程式，預設情況的輸入應該是單一資料表的 SQL 查詢結果，輸出是 Python 的 DataFrame 資料框架物件。

19-4-1 輸入與輸出參數

在說明如何使用 Python 輸入與輸出參數前，我們需要先了解 sp_execute_external_script 系統預存程序的語法，如下所示：

```
sp_execute_external_script
@language = N'language',
@script = N'script'
[ , @input_data_1 = N'input_data_1' ]
[ , @input_data_1_name = N'input_data_1_name' ]
[ , @output_data_1_name = N'output_data_1_name' ]
[ , @params = N'@parameter_name data_type [ OUT | OUTPUT ] [ ,...n ]' ]
[ , @parameter1 = 'value1' [ OUT | OUTPUT ] [ ,...n ] ]
WITH RESULT SETS((欄位名稱 資料類型, [欄位名稱 資料類型]));
GO
```

上述@language 和@script 參數已經在第 19-3 節說明過，其他參數和子句的說明，如下所示：

- @input_data_1 參數：指定 Python 程式碼使用的輸入資料，其資料類型是 nvarchar(max)。
- @input_data_1_name 參數：指定@input_data_1 定義輸入資料的變數名稱。Python 輸入變數的預設名稱是 InputDataSet。
- @output_data_1_name 參數：指定完成預存程序呼叫後，回傳給 SQL Server 的資料變數名稱。在 Python 程式碼的輸出資料必須指定給輸出變數，輸出變數的預設名稱是 OutputDataSet，變數值是 pandas 套件的 DataFrame 資料框架物件。
- @params 和@parameter1 參數：分別是外部腳本輸入參數名稱和值宣告的清單。
- WITH RESULT SETS 子句：定義回傳資料表的結構描述，可以將回傳的 DataFrame 物件轉換成 SQL Server 資料表。

顯示 InputDataSet 輸入參數的資料：Ch19_4_1.sql

Python 預設的輸入參數名稱是 InputDataSet 變數，當我們指定輸入資料的 SELECT 查詢指令後，就可以使用 print()函數顯示輸入參數的資料，這是一個 DataFrame 物件，如下所示：

```
USE 教務系統
GO
EXEC sp_execute_external_script
@language = N'Python',
@script = N'
print(type(InputDataSet))
print(InputDataSet)
',
@input_data_1 = N'SELECT 課程編號, dbo.UnicodeString(名稱) AS 名稱, 學分 FROM 課程'
```

上述 Python 程式碼使用 2 個 print()函數依序使用 type()函數顯示輸入參數的型態和參數值，@input_data_1 參數指定輸入資料的 SELECT 指令，如下所示：

```
SELECT 課程編號,dbo.UnicodeString(名稱) AS 名稱,學分 FROM 課程
```

上述 SELECT 指令可以查詢課程資料表的記錄資料，因為名稱欄位並不是 Unicode 編碼，在轉換成 Python 輸入資料時會產生錯誤，所以在欄位呼叫 dbo.UnicodeString()純量值函式轉換成 Unicode 編碼，這是我們自行建立的一個 SQL Server 自訂函數，如下圖所示：

```
df = pd.DataFrame(s, index=[1])
OutputDataSet = df
'
WITH RESULT SETS(([值] int))
```

上述 Python 程式碼使用 Series 物件建立 DataFrame 物件，第 2 個參數 index=[1]指定只取出索引值 1（即第 2 個元素），其執行結果如右圖所示：

19-5 匯入訓練資料至 SQL Server 資料庫

在使用 SQL Server 訓練機器學習模型前，我們需要先將訓練資料匯入 SQL Server 資料庫。

19-5-1 使用 Python 將資料匯入 SQL Server 資料庫

在 Python 的 Scikit-learn 套件內建的 Iris 資料集是鳶尾花的資料，可以讓我們訓練模型使用花瓣和花萼來分類鳶尾花。我們準備將 Iris 資料集匯入儲存至 SQL Server 資料庫。

步驟一：建立資料庫與資料表

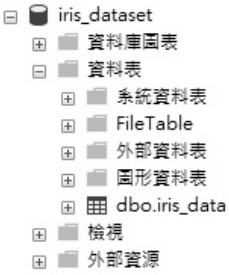
首先，我們準備建立名為 iris_dataset 資料庫，和新增 iris_data 資料表來匯入 Iris 資料集的資料（SQL 指令碼檔案：Ch19_5_1.sql），如下所示：

```
USE master
GO
CREATE DATABASE iris_dataset
```

執行上述 SQL 指令可以建立名為 iris_dataset 的資料庫，然後，我們就可以在此資料庫新增 iris_data 資料表來匯入資料（SQL 指令碼檔案：Ch19_5_1a.sql），如下所示：

```
USE iris_dataset
GO
DROP TABLE IF EXISTS iris_data;
GO
CREATE TABLE iris_data (
id           INT NOT NULL IDENTITY PRIMARY KEY,
Sepal_Length FLOAT NOT NULL,
Sepal_Width  FLOAT NOT NULL,
Petal_Length FLOAT NOT NULL,
Petal_Width  FLOAT NOT NULL,
Species      VARCHAR(100) NOT NULL,
SpeciesId    INT   NOT NULL
);
```

上述 SQL 指令在切換至 iris_dataset 資料庫後，檢查資料表 iris_data 是否存在，如果存在就刪除資料表，然後建立 iris_data 資料表，執行後，我們可以在 iris_dataset 資料庫建立 iris_data 資料表，如右圖所示：



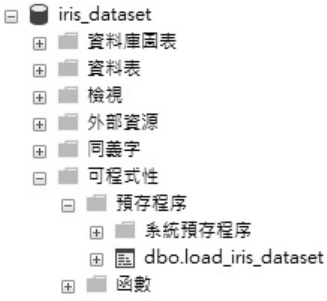
步驟二：將訓練資料載入 SQL Server 資料表

在成功建立 iris_dataset 資料庫和 iris_data 資料表後，我們就可以將 Iris 資料集匯入儲存至 SQL Server 資料庫，首先，我們需要建立預存程序來載入 Iris 資料集（SQL 指令碼檔案：Ch19_5_1b.sql），如下所示：

```
USE iris_dataset
GO
CREATE PROCEDURE load_iris_dataset
AS
BEGIN
EXEC sp_execute_external_script @language = N'Python',
@script = N'
from sklearn import datasets
iris = datasets.load_iris()
iris_data = pandas.DataFrame(iris.data)
iris_data["Species"] = pandas.Categorical.from_codes(iris.target,
iris.target_names)
iris_data["SpeciesId"] = iris.target
```

```
',
@input_data_1 = N'',
@output_data_1_name = N'iris_data'
WITH RESULT SETS ((Sepal_Length float not null,
    Sepal_Width float not null,
    Petal_Length float not null, Petal_Width float not null,
         Species varchar(100) not null, SpeciesId int not null));
END;
```

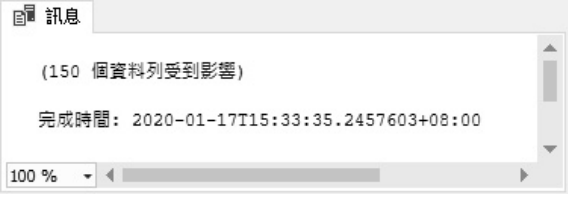
上述 SQL 指令可以建立名為 load_iris_dataset 的預存程序，Python 程式碼是載入 Scikit-learn 套件內建的 Iris 資料集，如右圖所示：



然後，我們可以呼叫預存程序，將 Iris 資料集匯入 iris_data 資料表（SQL 指令碼檔案：Ch19_5_1c.sql），如下所示：

```
USE iris_dataset
GO
INSERT INTO iris_data (Sepal_Length, Sepal_Width,
     Petal_Length, Petal_Width, Species, SpeciesId)
EXEC dbo.load_iris_dataset;
```

上述 SQL 指令呼叫 load_iris_dataset 預存程序來匯入 Iris 資料集，其執行結果可以看到匯入 150 筆記錄，如下圖所示：



步驟三：查詢載入 SQL Server 資料表的訓練資料

在成功匯入 Iris 資料集後，我們可以查詢前 10 筆記錄資料（SQL 指令碼檔案：Ch19_5_1d.sql），如下所示：

```
USE iris_dataset
GO
SELECT TOP(10) * FROM iris_data
```

上述 SQL 指令的執行結果可以顯示前 10 筆記錄，如下圖所示：

結果 訊息

	id	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species	SpeciesId
1	1	5.1	3.5	1.4	0.2	setosa	0
2	2	4.9	3	1.4	0.2	setosa	0
3	3	4.7	3.2	1.3	0.2	setosa	0
4	4	4.6	3.1	1.5	0.2	setosa	0
5	5	5	3.6	1.4	0.2	setosa	0
6	6	5.4	3.9	1.7	0.4	setosa	0
7	7	4.6	3.4	1.4	0.3	setosa	0
8	8	5	3.4	1.5	0.2	setosa	0
9	9	4.4	2.9	1.4	0.2	setosa	0
10	1...	4.9	3.1	1.5	0.1	setosa	0

上述資料表的前幾個欄位分別是花萼（Sepal）和花瓣（Petal）的長和寬，單位是公分，Species 是哪一種鳶尾花的名稱 setosa、versicolor 和 virginica，最後是種類值：0 是 setosa；1 是 versicolor；2 是 virginica。

19-5-2 使用 SQL Server 的匯入和匯出精靈

在第 19-5-1 節我們是使用 Python 程式匯入內建 Iris 資料集，如果擁有外部資料，例如：CSV 檔案，我們可以直接使用 SQL Server 匯入和匯出精靈，將多種資料來源的資料匯入 SQL Server 資料庫。

步驟一：建立資料庫

首先建立 nba_dataset 資料庫來匯入 HOU_players_stats_2017.csv 的 CSV 檔案的休士頓火箭隊球員的統計資料（SQL 指令碼檔案：Ch19_5_2.sql），如下所示：

```
USE master
GO
CREATE DATABASE nba_dataset
```

上述 SQL 指令可以建立名為 nba_dataset 資料庫。

步驟二：使用 SQL Server 的匯入和匯出精靈匯入 CSV 檔案

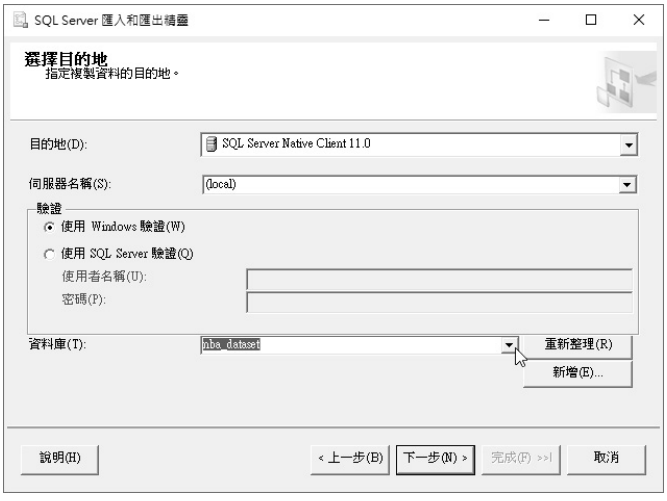
在成功建立 nba_dataset 資料庫後，我們準備將訓練資料的 CSV 檔案：HOU_players_stats_2017.csv 匯入成資料表，其步驟如下所示：

1. 請在 Windows 作業系統執行「開始>SQL Server 2019>SQL Server 2019 匯入及匯出資料 (64 位元)」命令啟動匯入和匯出精靈，在歡迎畫面按【下一步】鈕，如下圖所示：

2. 首先選擇資料來源，在上方【資料來源】欄選【一般檔案來源】後，按【瀏覽】鈕選【HOU_players_stats_2017.csv】檔案後，可以看到自動填入的文字檔案格式，請按【下一步】鈕，如下圖所示：

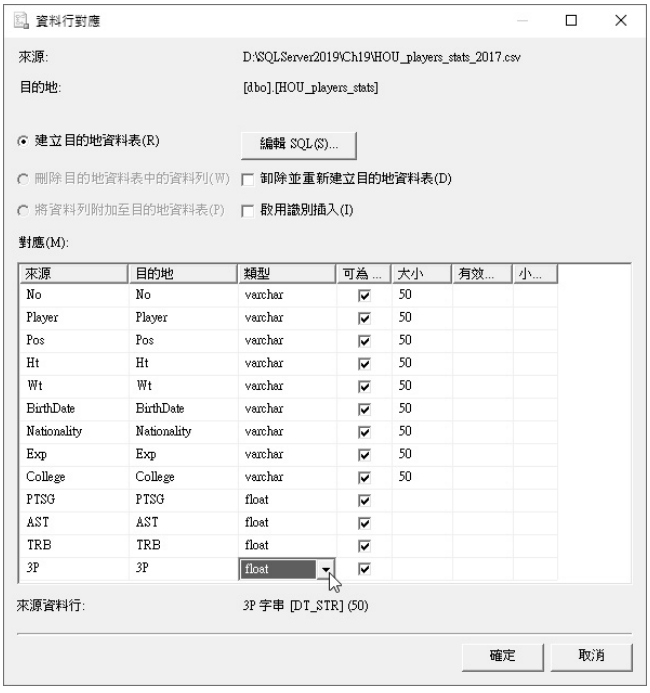
3 指定資料列記錄的分隔符號，和分隔資料行欄位的符號後，可以在下方預覽記錄資料，然後按【下一步】鈕，如下圖所示：

4. 接著選擇目的地，在【目的地】欄選【SQL Server Native Client 11.0】後，依序選 SQL Server 伺服器名稱、Windows 驗證，在【資料庫】欄選【nba_dataset】後，按【下一步】鈕，如下圖所示：



5. 在選取【目的地】後，更改成【[dbo].[HOU_players_stats]】資料表後，按【編輯對應】鈕編輯欄位對應資料，如下圖所示：

6. 在「資料行對應」對話方塊編輯來源和目的地的欄位對應，請將最後 4 個欄位的資料類型改為 float 後，按【確定】鈕，然後在精靈步驟按【下一步】鈕，如下圖所示：

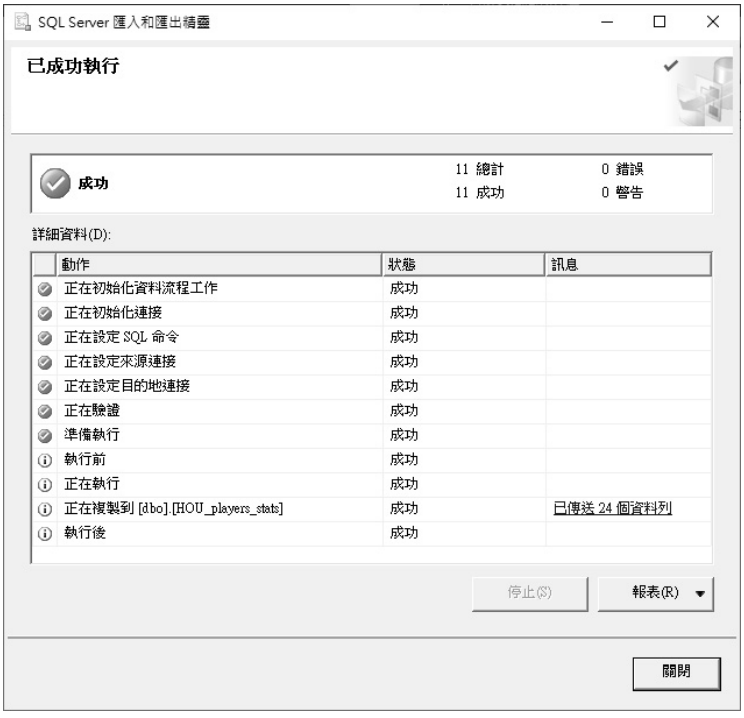


❼ 在檢視資料類型對應步驟顯示欄位資料轉換設定，SQL Server 會自動判決是否需進行轉換，沒有問題，請按【下一步】鈕，如下圖所示：

8 勾選【立即執行】（如需重複執行，請儲存成 SSIS 封裝）後，按【下一步】鈕，如下圖所示：

9 可以看到選擇的作業內容，請按【完成】鈕執行作業，如下圖所示：

10 可以看到正在執行從來源至目的地的匯入和匯出作業，成功執行後，請按【關閉】鈕完成操作，如下圖所示：



在 Management Studio 開啟【HOU_players_stats】資料表，可以看到成功匯入的 24 筆記錄資料，如下圖所示：

No	Player	Pos	Ht	Wt	BirthDate	Nationality	Exp	College	PTSG
33	Ryan Anderson	PF	6-10	240	"May 6 1988"	us	9	University of C...	9.3
1	Trevor Ariza	SF	6-8	215	"June 30 1985"	us	13	"University of ...	11.7
28	Tarik Black	C	6-9	250	"November 22...	us	3	"University of ...	3.5
6	Bobby Brown	PG	6-2	175	"September 2...	us	3	"California Sta...	2.5
26	Markel Brown	SG	6-3	190	"January 29 19...	us	2	Oklahoma Sta...	1.3
98	Isaiah Canaan	SG	6-0	201	"May 21 1991"	us	4	Murray State ...	0
15	Clint Capela	C	6-10	240	"May 18 1994"	ch	3		13.9
10	Eric Gordon	SG	6-4	215	"December 25...	us	9	Indiana Univer...	18
14	Gerald Green	SG	6-7	205	"January 26 19...	us	10		12.1