前言

本書特色

本書是利用程式設計語言「Python」[1] 參加機器學習競賽「Kaggle」[2] 的入門書。一開始先以適合初學者學習的「Titanic：Machine Learning from Disater」競賽[3]（Titanic）學習 Kaggle 的基礎。從中除了可學習實踐 Titanic 的方法，還能掌握自行參加競賽所需的知識。

本書具有以下六點特色：

1. 為了 Kaggle 撰寫的習作教學書籍。
2. 每一章或每一節都有具體的主題，可讓讀者按部就班地掌握需要的知識。
3. 解說各種表單、圖片檔、文字檔的操作方法，作為進入下一個競賽的指引。
4. 兩位執筆者都擁有「Kaggle Master」的稱號，也有贏得獎金的經驗。
5. 除了說明之外，還有兩位筆者的對談，從不同的角度介紹 Kaggle 的魅力。
6. 會為程式設計與 Python 的初學者詳細講解範例程式。

兩位筆者都有撰寫 Kaggle 入門內容的經驗。

2019 年 3 月，筆者（石原）在「Qiita」[4] 這個工程師文章共享網站公開了 Kaggle 入門文章「註冊 Kaggle 之後，接下來要做的事～光是這些就很值得一戰！ Titanic 的入門 10 Kernel ～」[5]。該文章超過 1600 個讚，在 Qiita 網站有「Kaggle」標籤的文章裡，是得到最多讚的文章[6]。

---

〔1〕 Welcome to Python.org
https://www.python.org/ (Accessed: 30 November 2019).

〔2〕 Kaggle: Your Home for Data Science
https://www.kaggle.com/ (Accessed: 30 November 2019).

〔3〕 Titanic: Machine Learning from Disaster
https://www.kaggle.com/c/titanic (Accessed: 30 November 2019).

〔4〕 Qiita
https://qiita.com/ (Accessed: 30 November 2019).

〔5〕 Kaggleに登録したら次にやること ～ これだけやれば十分闘える！Titanicの先へ行く入門 10 Kernel ～
https://qiita.com/upura/items/3c10ff6fed4e7c3d70f0 (Accessed: 30 November 2019).

〔6〕 Kaggle - Qiita
https://qiita.com/tags/kaggle (Accessed: 30 November 2019).

2018 年 4 月，筆者（村田）出版了《Kaggle のチュートリアル》[7]。在內容分享網站「note」銷售的這本書共賣出 2500 本以上。

本書將以上述這兩種內容為主，並且另外補充各種內容，主要的目的是讓本書成為「**Kaggle 初學者必讀的入門書**」。

筆者（石原）是資料科學家，另一位筆者（村田）則是「專業 Kaggler」，平日就不斷應用機器學習的資料科學。兩位筆者根據己身經驗，將本書寫成了解 Kaggle 樂趣，又能學到實用、通用知識的內容。

本書的內容皆根據 2019 年 11 月的資訊寫成。Kaggle 的網站構造則支援 2020 年 2 月的變更。

本書的目標讀者

本書的目標讀者如下：

1. 對 Kaggle 有興趣，但不知該從何開始的人。
2. 有自行摸索的經驗，但想以系統化的方式學習 Kaggle 的人。
3. 想一邊動手做，一邊了解機器學習概要的人。
4. 對 Python 或機器學習有一定程度了解，但第一次面對 Kaggle 的人。

尤其前兩者更是本書訴求的目標讀者。

本書會以章或節為單位，帶領大家按部就班地學習 Kaggle 的精華，也會以 Titanic 這個具體的主題，帶領大家系統性地掌握相關知識。

初學者有可能會遇到「不懂機器學習」、「不懂 Python」、「不了解 Kaggle 的機制」、「很難讀懂英文的內容」這些障礙。

本書除了在內文說明 Kaggle 所需的基礎知識，也會透過附錄或補充說明，方便每個人參考，例如會在附錄詳細解說範例程式。本書也會隨時插入「note」這些補充以及解說英文 Kaggle 網頁的內容。

本書不需要具備 Kaggle 或機器學習的背景知識就能閱讀。Kaggle 是自行撰寫程式以及一邊參加競賽，一邊學習機器學習相關知識的平台，所以想一邊動手做，一邊了解機器學習概要的人，也非常適合閱讀本書。

〔7〕 村田秀樹『Kaggleのチュートリアル』
https://note.mu/currypurin/n/nf390914c721e (Accessed: 30 November 2019).

內文僅 150 頁，算是能快速讀完的份量。對 Python 或機器學習有一定程度了解的人，可以只閱讀內文，快速了解 Kaggle 的機制。

每位讀者應該都能根據自身的知識與經驗應用本書的內容。

本書的架構

本書大致分成 1 ～ 4 章的內文與附錄。

第 1 章會先介紹 Kaggle 的概要。一開始先介紹「Kaggle 是何物」，說明得先了解的機器學習概論，同時也會介紹註冊與登入 Kaggle 的方法，以及不需要事先建立環境就能使用的分析環境該如何使用。

第 2 章要介紹 Titanic。本章共由 8 個節組成，可以一邊提升分數，一邊學習 Kaggle 的精華。

第 3 章的主題是「往 Titanic 的下個階段前進」，介紹未於 Titanic 登場的 Kaggle 元素。這部分可幫助大家了解，如何憑一己之力參加正在舉辦的競賽。本章由 3 個節組成，從中可學習多個表單、圖片檔、文字檔的操作方式。

第 4 章是內文的總結，主要是介紹讀完本書之後，可能會需要的相關資訊。其中會說明適合初學者的競賽以及參賽方式，還介紹分析環境的相關資訊與有用的資料。

附錄則會詳細介紹本書的程式碼，同時為了程式設計與 Python 的初學者介紹變數、列表這類程式設計的基礎內容。

範例程式

本書的範例程式已於下列的「GitHub」[8] 公開。GitHub 主要是工程師分享程式碼的網站。

https://github.com/upura/python-kaggle-start-book

同樣的範例程式也於 Kaggle 上傳了。細節的部分請於 GitHub 確認。

範例程式可在 Python 3.6 版執行與確認。

執行 Python 的環境為「Docker」[9]。關於建構環境的方法將於 1.5 節說明。

[8] GitHub
http://github.com (Accessed: 30 November 2019).

[9] Docker: Enterprise Container Platform
https://www.docker.com/ (Accessed: 30 November 2019).

Kaggle 提供的分析環境雖然會隨時更新，但本書採用的是有「v68」這個標籤的版本[10]。主要套件的版本如下：

- lightgbm==2.3.0
- matplotlib==3.0.3
- numpy==1.16.4
- pandas==0.25.2
- scikit-learn==0.21.3

作者介紹

石原　祥太郎（u++）

- Kaggle Master（https://kaggle.com/sishihara）·
- 2019 年 4 月於「PetFinder.my Adoption Prediction」競賽[11]獲得冠軍。
- 2019 年 12 月協助舉辦「Kaggle Days Tokyo」[12]的競賽。
- 2019 年 3 月在 Qiita 公開的 Kaggle 入門文章得到 1600 個讚[5]。
- 於日本經濟新聞社從事資料分析[13]。

村田　秀樹（咖哩）

- Kaggle Master（https://kaggle.com/currypurin）·
- 2018 年 8 月於「Santander Value Prediction Challenge」競賽[14]得到 solo gold medal（第 8 名）。
- 2019 年 6 月於「LANL Earthquake Prediction」競賽[15]得到第三名。
- 為了 Kaggle 初學者所寫的同人誌《Kaggle 的習作》[7]累計賣出 2500 本。
- 從 2018 年 7 月開始成為專職 Kaggler。

〔10〕 Container Registry - Google Cloud Platform
https://console.cloud.google.com/gcr/images/kaggle-images/GLOBAL/python (Accessed: 30 November 2019).

〔11〕 PetFinder.my Adoption Prediction
https://www.kaggle.com/c/petfinder-adoption-prediction (Accessed: 30 November 2019).

〔12〕 Kaggle Days Tokyo
https://kaggledays.com/tokyo/ (Accessed: 30 November 2019).

〔13〕 機械学習を用いた日経電子版Proのユーザ分析 データドリブンチームの知られざる取り組み
https://logmi.jp/tech/articles/321077 (Accessed: 30 November 2019).

〔14〕 Santander Value Prediction Challenge
https://www.kaggle.com/c/santander-value-prediction-challenge (Accessed: 30 November 2019).

〔15〕 LANL Earthquake Prediction
https://www.kaggle.com/c/LANL-Earthquake-Prediction (Accessed: 30 November 2019).

1.1 何謂 Kaggle

Kaggle 是資料科學家、機器學習工程師的線上社群，常會舉辦各種競賽，供全世界的參賽者比拼機器學習模型的性能，也提供執行程式設計語言「Python」或「R」[16]的「Notebooks」環境。這個社群有許多公開的程式碼，相關的討論也非常熱絡，所以不管你是初學者還是專家，這裡可說是最適合學習機器學習的平台。

Kaggle 的競賽概要請參考圖 1.1，主要的流程如下：

1. 企業提供資料與獎金的企業委託 Kaggle 舉辦競賽，由 Kaggle 進行籌辦。
2. 參賽者可在分析資料後，submit（提交）預測結果，預測結果也會自動接受評分。
3. 競賽期間（一般都是為期 2 ～ 3 個月），參賽者可不斷 submit 預測結果，確認最終分數。
4. 競賽結束後，將由高至低排序分數，前段班的參賽者可得到獎金與獎牌。
5. 若是得到一定數量的獎牌，可獲得高階的稱號。

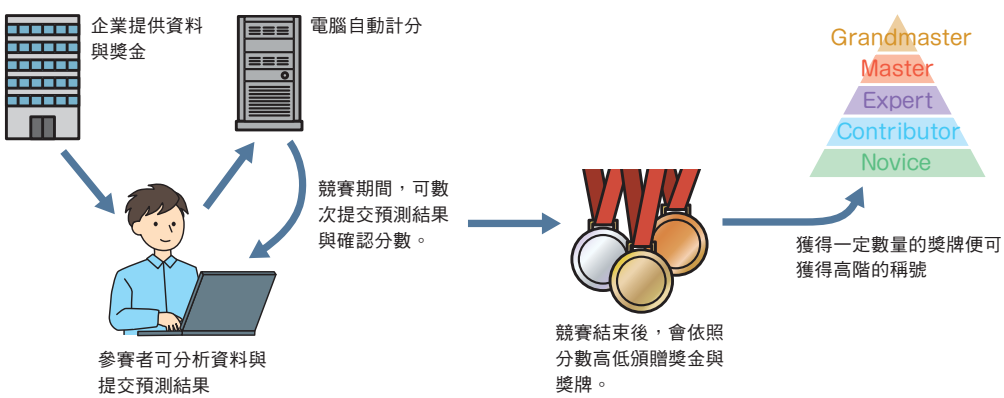

圖 1.1 Kaggle 的競賽概要

[16] R: The R Project for Statistical Computing
https://www.r-project.org/ (Accessed: 30 November 2019).

參賽者不需要自行準備資料，一旦闖進前幾名就能賺到獎金，名次不佳也不會受到任何處罰。之所以針對提交的預測結果計分與排名，為的是讓參賽者能像闖關般，提升自己的名次，享受學習機器學習的過程。

註冊帳號後，會先得到「Novice」（初學者）的稱號，等到在特定的競賽收集到一定數量的獎牌，就能獲得 Expert、Master、Grandmaster 這些高階的稱號，這些稱號也是讓參賽者想要贏得競賽的動力。

note

贏得獎金與稱號的條件

若在 Kaggle 的獎牌競賽闖進前幾名就能獲得獎牌。可獲得獎牌的名次會依照參賽隊伍的多寡調整，請參考圖 1.2 說明 [17]。

	0～99 隊	100～249 隊	250～999 隊	1000 隊～
銅牌	前 40%	前 40%	前 100 隊	前 10%
銀牌	前 20%	前 20%	前 50 隊	前 5%
金牌	前 10%	前 10 隊	前 10 隊＋0.2%	前 10 隊＋0.2%

圖 1.2　參賽隊伍數與獎牌順位的相關性

贏得金牌的條件「前 10 隊 +0.2%」的 0.2% 是指每 500 隊就會多增加一個獲得金牌的參賽隊伍，換言之，若有 1000 隊參賽，前 12 名的隊伍都能贏得金牌，若有 2000 隊參賽，則前 14 名的隊伍都能贏得金牌。

收集到一定數量的獎牌可贏得稱號，贏取方式請參考圖 1.3 [17]

稱號	贏取稱號的條件
Grandmaster	金牌 5 個、其中必須有一個是個人獎牌（Solo）
Master	金牌 1 個、銀牌 2 個
Expert	銅牌 2 個
Contributor	完成個人檔案（細節請參考下列說明）
Novice	於 Kaggle 註冊

圖 1.3　Competitions（競賽）的各稱號贏取條件

2.2 掌握全貌！了解 submit 之前的處理流程

2.2 節要帶大家具體了解在 2.1 節跳過的 Notebook 處理流程，請試著從最上方的 cell 執行，了解接下來的內容。

具體的處理流程如下：

1. 載入套件
2. 載入資料
3. 特徵工程
4. 機器學習演算法的學習與預測
5. submit

2.2.1 載入套件

```
1: import numpy as np
2: import pandas as pd
```

第一步要先 import 之後會用到的「套件」，如此一來就能使用未內建，但方便實用的擴充功能。

舉例來說，前面 import 的 NumPy〔26〕是善於計算的套件，Pandas〔27〕則可輕鬆操作 Titanic 這類表格格式的資料（表格資料）的套件。

這節先 import 一開始就會用到的這兩個套件。import 可在任何一個 cell 執行，但通常會在開頭執行，不過為了讓本書的範例程式更簡單易懂，偶爾會在要使用之前才 import。

〔26〕NumPy
https://numpy.org/ (Accessed: 30 November 2019).

〔27〕Pandas
https://pandas.pydata.org/ (Accessed: 30 November 2019).

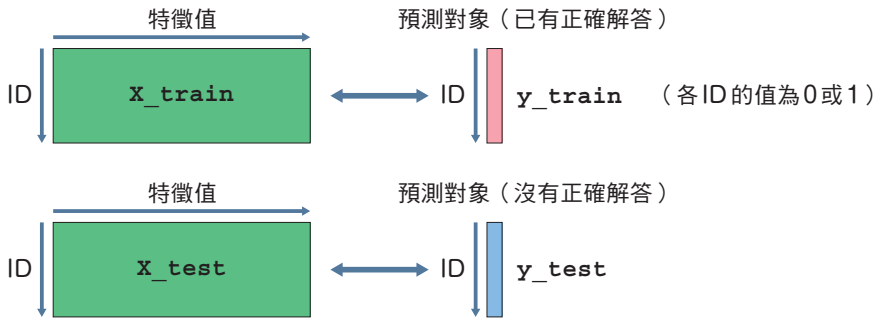

圖 2.13　經過特徵工程處理的資料

經過特徵工程處理之後，可從原始資料得到「X_train」、「y_train」、「X_test」這類可於機器學習演算法使用的資料集。

note

特徵值的標準化

舉例來說，Titanic 的 Sex 欄位只有 0 或 1 的值，也知道 Age 的最大值為 80。

本書採用的機器學習演算法「邏輯迴歸」有時會在各特徵值範圍不同時，無法正確學習，所以通常得調整各特徵值的範圍。

此時最常用來調整特徵值的方法就是「標準化」，也就是將各特徵值的平均轉換為 0，標準差轉換為 1。標準差代表的是資料的分散程度，寫程式的時候，可使用 sklearn.preprocessing.StandardScaler() [28]。

不過近年來，在 Kaggle 操作表格資料時，通常會使用不受上述標準化轉換特徵值影響的機器學習演算法，例如在 2.5 節用來取代邏輯迴歸的「決策樹」[29] 或「LightGBM」[30] 就是其中一種。

本書最終採用的是「LightGBM」這種機器學習演算法，所以在此先跳過標準化特徵值的處理。

〔28〕sklearn.preprocessing.StandardScaler
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html (Accessed: 30 November 2019).

〔29〕sklearn.ensemble.RandomForestClassifier
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (Accessed: 30 November 2019).

〔30〕LightGBM
https://lightgbm.readthedocs.io/en/latest/ (Accessed: 30 November 2019).

2.3 找出下一步！試著進行探索式資料分析

我們在 2.1 ～ 2.2 節將取得的資料改造成機器學習演算法能操作的格式，也利用機器學習演算法進行學習與預測，接下來我們要一步步改善學習與預測的流程，試著提升分數。

有待改善的重點之一就是特徵工程，這也是利用機器學習演算法根據 2.2 節介紹的資料進行預測，建立可用的新特徵值的方法。

要建立提升預測性能的特徵量，就必須不斷建立假設以及讓資料更具體化的步驟。在此試著將建立假設與讓資料變得更具體的流程整理成圖（圖 2.14）。

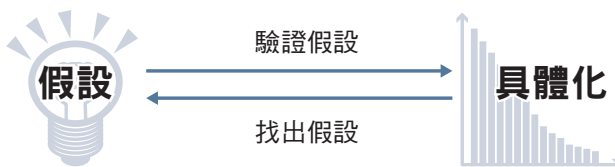

圖 2.14 假設與具體化

- 建立能提升預測性能的假設
- 讓預測變得更具體（為了找出提升性能的假設，也為了驗證假設是否正確）

每個人的想法不同，題目也不同，所以起點也會跟著改變。

■情況 1）具備領域知識的情況

所謂「領域知識」是特定業界或事業的專業知識。

如果要處理的課題剛好是你具備的領域知識，也就是你很熟悉的領域時，可能一開始會找出很多個假設，此時可試著讓資料更透明具體，確認假設是否真的有助於提升預測性能，有時甚至會因此建立截然不同的假設。

■情況 2）不具備領域知識的情況

如果不具備該領域知識，就可執行探索式資料分析再建立假設。以不同的座標軸俯瞰資料，進而建立足以提升預測性能的假設是實施探索式資料分析的目的。

note

分類變數的特徵工程

機器學習演算法通常無法操作非數值的字串資料，所以必須在進行特徵工程的時候，將分類變數轉換成適用的數值。

2.2 節是將 Sex 欄位的「male」與「female」分別轉換成 0 與 1。

```
1: data['Sex'].replace(['male', 'female'], [0, 1], inplace=True)
```

如果一個分類變數只有兩個值，只需要透過上述的處理轉換，但如果值超過三個，就必須另行處理。

比方說，這次將 Embarked 欄位的 S、C、Q 分別轉換成 0、1、2，但這種處理並不適合於邏輯迴歸這種機器學習演算法使用。

因為轉換成數值之後，很可能會出現原本沒有的關聯性，意即，機器學習演算法很可能會以為「C 位於 S 與 Q 之間」。

基於上述理由，在將值超過三種以上的分類變數轉換成數值時，會使用圖 2.16 這類手法。

Embarked
S
C
Q
C

Embarked_S	Embarked_C	Embarked_Q
1	0	0
0	1	0
0	0	1
0	1	0

圖 2.16　One-Hot 編碼

將 Embarked 的資料展成 3 欄，再以 0 或 1 標記各欄的特定值。這種轉換可避免產生不需要的關聯性，而這種手法則稱為「One-Hot 編碼」，要注意的是，如果一個分類變數有過多的值，欄數可能會大增。

2.5 節使用的「LightGBM」與少數的機器學習演算法都內建了替分類變數指定特徵值的功能，所以就算為了處理分類變數而將分類變數的資料轉換成數值，也不會發生上述產生多餘關聯性的問題。

本書最終是採用 LightGBM 這種機器學習演算法，所以不需要透過 One-Hot 編碼這種手法處理分類變數。

對談 ④ 不是只讓資料「具體化」

我透過 Kaggle 學到的事情之一，就是探索式資料分析有多麼重要。還記得剛開始學習機器學習的時候，總是將焦點放在演算法與建立模型這類事情上面，但最近越來越覺得探索式資料分析有其價值所在，尤其現在建立機器學習模型的門檻越來越低，要與別人拉開差距，就一定要了解探索式資料分析（EDA）。

自動執行前置處理與建立機器學習模型的技術已有長足的進步，只要輸入資料，就能自動建立堪用的模型，而且模型的性能也越來越強，但這類技術還沒辦法處理多個表格的資料，也無法處理特定的工作，因此 Kaggler 唯有透過 EDA 與建立模型的技術才能突顯自己的長處。

我從最近的表格資料競賽發現觀察資料這件事越來越重要。資料科學的套件與技術已慢慢普及，一般的迴歸、分類問題已可在公司內部自行解決，這導致 Kaggle 的表格資料競賽在資料集與課題設定上，都設定了比較有固定特性的題目。

例如 2019 年舉辦的「LANL Earthquake Prediction」[15]、「Santander Customer Transaction Prediction」[32]、「IEEE-CIS Fraud Detection」[33] 都是必須仔細觀察資料並且執行適當處理的團隊拿到前面的名次。

我覺得一般的表格資料競賽越來越少，該如何透過 EDA 解決問題的重要性似乎越來越高。

我覺得探索式資料分析主要分成兩種模式，一種是一開始先掌握資料的輪廓，一種是先嘗試預測，再找出假設。

〔15〕 LANL Earthquake Prediction
https://www.kaggle.com/c/LANL-Earthquake-Prediction (Accessed: 30 November 2019).

〔32〕 Santander Customer Transaction Prediction
https://www.kaggle.com/c/santander-customer-transaction-prediction (Accessed: 30 November 2019).

〔33〕 IEEE-CIS Fraud Detection
https://www.kaggle.com/c/ieee-fraud-detection (Accessed: 30 November 2019).

2.5 決策樹是最強的演算法？試著使用各種機器學習演算法

到目前為止，我們採用的是邏輯迴歸這種機器學習演算法。

接著讓我們試用各種機器學習演算法吧！主要是將邏輯迴歸的部分換掉，再進行學習與預測。

用於撰寫邏輯迴歸的 sklearn 套件最近統一了輸出／輸入的介面，所以可以輕鬆地變更機器學習演算法。

最近 Kaggle 競賽的前段班較常使用的機器學習演算法有「決策樹」與「類神經網路（Neural Network，NN）這兩種，而這兩種在敘述力與性能面上，都勝過邏輯迴歸這種演算法。

Kaggle 競賽的前段班特別愛用「LightGBM」這種決策樹的套件，這個套件與 sklearn 內建了相同的介面，但本節要使用「Python-package Introduction」這個頁面〔46〕介紹的方法撰寫，以便更有效率地使用記憶體。

2.5.1 sklearn

第一步要先變更 sklearn 之內的機器學習演算法。到目前為止，我們都是使用邏輯迴歸這個演算法。

```
1: from sklearn.linear_model import LogisticRegression
2:
3:
4: clf = LogisticRegression(penalty='l2', solver='sag', random_state=0)
```

只要更換 clf 宣告的模型，就能更換 sklearn 的機器學習演算法。讓我們試著使用「隨機森林」〔29〕這個機器學習演算法。

〔46〕 Python-package Introduction
https://lightgbm.readthedocs.io/en/latest/Python-Intro.html (Accessed: 30 November 2019).