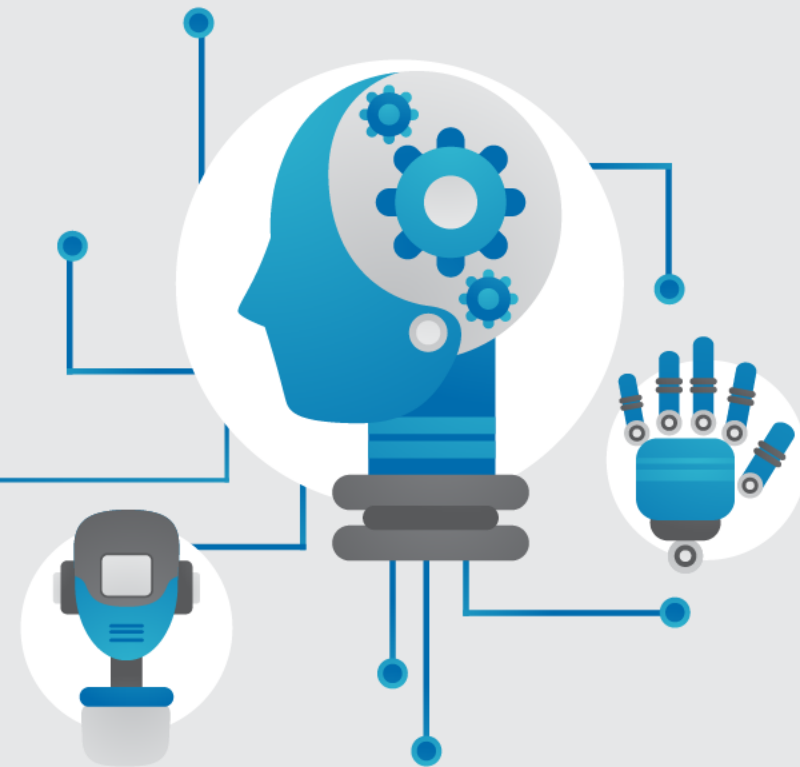


# MLP模型建置流程





# Kaggle競賽 - Digit Recognizer

機器學習實務



› 網址：<https://www.kaggle.com/c/digit-recognizer>

The screenshot shows the Kaggle website interface for the 'Digit Recognizer' competition. The browser address bar displays 'kaggle.com/c/digit-recognizer'. The page header includes the Kaggle logo, a search bar, and a notification bell. The main content area features a large banner with the text 'Digit Recognizer' and 'Learn computer vision fundamentals with the famous MNIST data'. Below the banner, there are tabs for 'Overview', 'Data', 'Notebooks', 'Discussion', 'Leaderboard', 'Rules', 'Team', 'My Submissions', and 'Submit Predictions'. The 'Overview' tab is selected, showing a 'Description' section with the text 'Start here if...' and a paragraph about the competition's focus on computer vision and the MNIST dataset.

Digit Recognizer  
Learn computer vision fundamentals with the famous MNIST data  
Kaggle 2,224 teams · Ongoing

Overview Data Notebooks Discussion Leaderboard Rules Team My Submissions Submit Predictions

Overview

Description

Tutorial

Evaluation

Start here if...

You have some experience with R or Python and machine learning basics, but you're new to computer vision. This competition is the perfect introduction to techniques like neural networks using a classic dataset including pre-extracted features.



# Kaggle競賽 - Digit Recognizer

機器學習實務



› 資料檔案：train.csv, test.csv和sample\_submission.csv

- 訓練資料：train.csv
  - ✓ 共42,000筆資料 ( row )
  - ✓ 每個row有785 columns
  - ✓ 第1 column為label，表示該影像的正確答案
  - ✓ 後面784 columns為row major的28x28像素值

	A	B	C	D	E	F	G
1	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5
2	1	0	0	0	0	0	(
3	0	0	0	0	0	0	(
4	1	0	0	0	0	0	(
5	4	0	0	0	0	0	(
6	0	0	0	0	0	0	(
7	0	0	0	0	0	0	(
8	7	0	0	0	0	0	(
9	3	0	0	0	0	0	(
10	5	0	0	0	0	0	(



# Kaggle競賽 - Digit Recognizer

機器學習實務



- 資料檔案：train.csv, test.csv和sample\_submission.csv
  - 測試資料：test.csv
    - ✓ 共28,000筆資料 ( row )
    - ✓ 每個row只有784 columns

	A	B	C	D	E	F
1	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0



# Kaggle競賽 - Digit Recognizer

機器學習實務



- 資料檔案：train.csv, test.csv和sample\_submission.csv
  - 預測結果：sample\_submission.csv
    - ✓ 共28,000筆資料 ( row )

	A	B
1	ImageId	Label
2	1	0
3	2	0
4	3	0
5	4	0
6	5	0
7	6	0
8	7	0
9	8	0
10	9	0



# MLP模型建置流程

機器學習實務



1. 資料前處理



2. 決定模型架構



3. 編譯與訓練模型



4. 模型評估



5. 調整超參數



6. 重複步驟2~5  
直到模型效率無法再改進

7. 進行預測







# 資料前處理



- › 從檔案train.csv讀進資料，區分資料x和標記y

```
3 # 從檔案讀取資料
4 with open('train.csv', 'r') as file:
5     csv_lines = file.readlines()
6 x = []
7 y = []
8 for i in range(1, len(csv_lines)):
9     # 去掉換行符號並以逗號分割
10    row = csv_lines[i].replace('\n', '').split(',')
11    # 去掉label欄位，並將字串轉為整數
12    x.append(list(map(int, row[1:])))
13    # 抓出label欄位，並將字串轉為整數
14    y.append(list(map(int, row[0])))
15
16 from keras.utils import to_categorical
17 # one-hot encoding
18 y = to_categorical(y, num_classes=10)
```

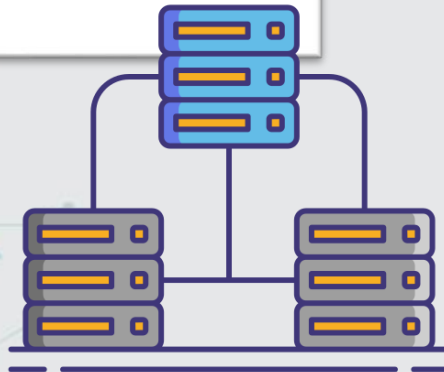


# 資料前處理



## 資料正規化

```
20 import numpy as np
21 #轉成np.array，正規化
22 x=np.array(x)/255.0
23 y=np.array(y)
```





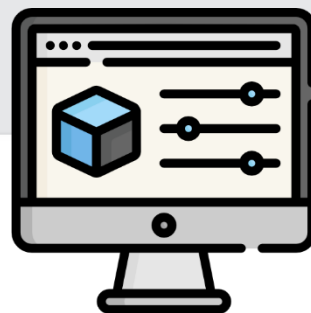


# 決定模型架構



## › 模型建置

```
27 # 建置模型
28 from keras.models import Sequential
29 from keras.layers import Dense
30 # 宣告Sequential循序模型
31 model = Sequential()
32 # 加入輸入層
33 model.add(Dense(512, activation='relu', input_shape=(784,)))
34 # 加入隱藏層
35 model.add(Dense(512, activation='relu'))
36 # 指定 輸出層模型
37 model.add(Dense(10, activation='softmax'))
```





# 編譯與訓練模型



## › 設定參數，訓練模型

```
40 # 編譯與訓練模型
41 from keras.optimizers import RMSprop
42 # 指定 loss function, optimizer, metrics
43 model.compile(loss='categorical_crossentropy',
44               optimizer=RMSprop(),
45               metrics=['acc'])
46 # 指定 batch_size, epochs, validation_split後，開始訓練模型
47 history = model.fit(x, y,
48                     batch_size=128,
49                     epochs=20,
50                     validation_split=0.2,
51                     verbose=1)
```



# 模型評估



## › 顯示訓練歷程

```
53 # 顯示訓練歷程
54 import matplotlib.pyplot as plt
55 def show_train_history(train_history):
56     plt.plot(train_history.history['acc'])
57     plt.plot(train_history.history['val_acc'])
58     plt.xticks([i for i in range(0, len(train_history.history['acc']))])
59     plt.title('Train History')
60     plt.ylabel('acc')
61     plt.xlabel('epoch')
62     plt.legend(['train', 'validation'], loc='upper left')
63     plt.show()
64
65 show_train_history(history)
```

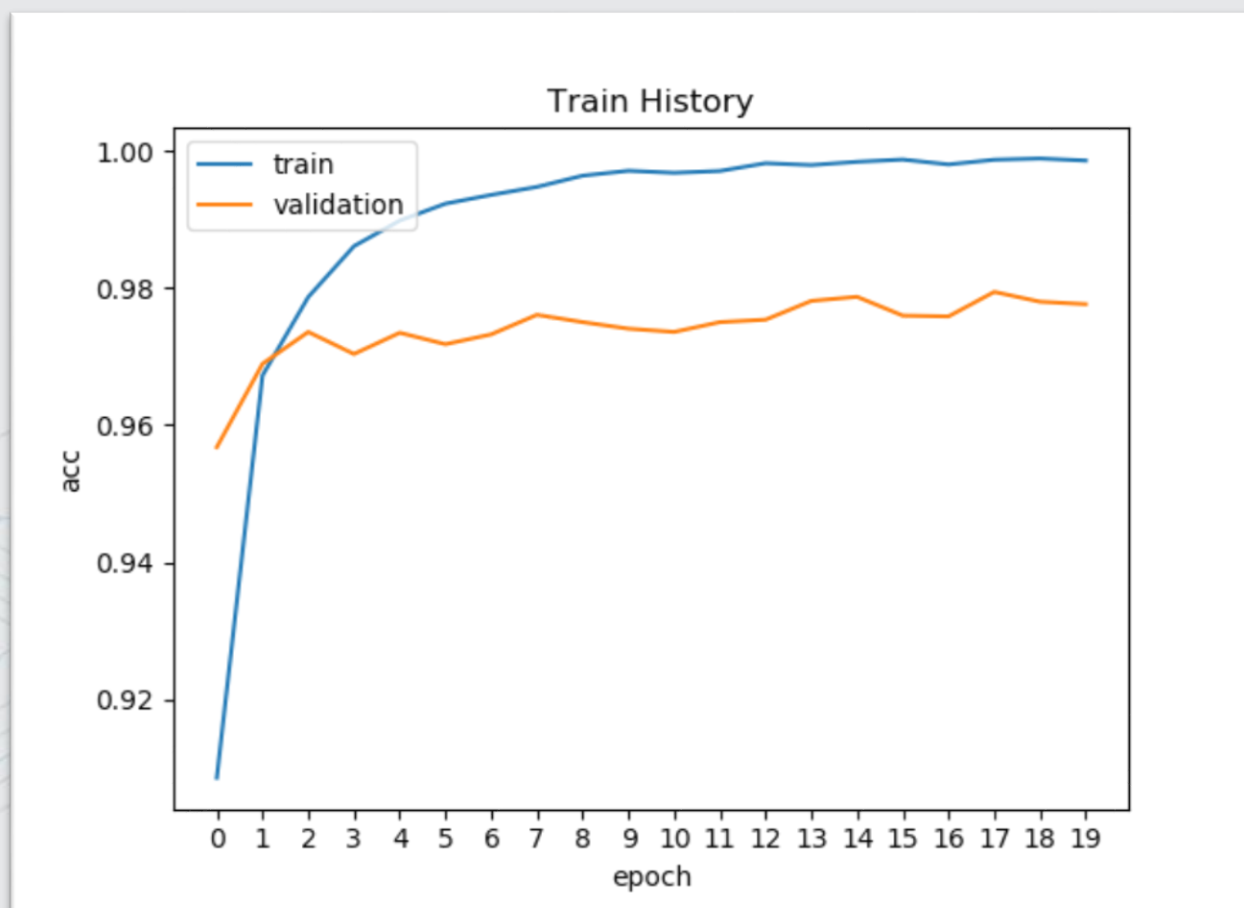


# 模型評估

機器學習實務



› 顯示訓練歷程

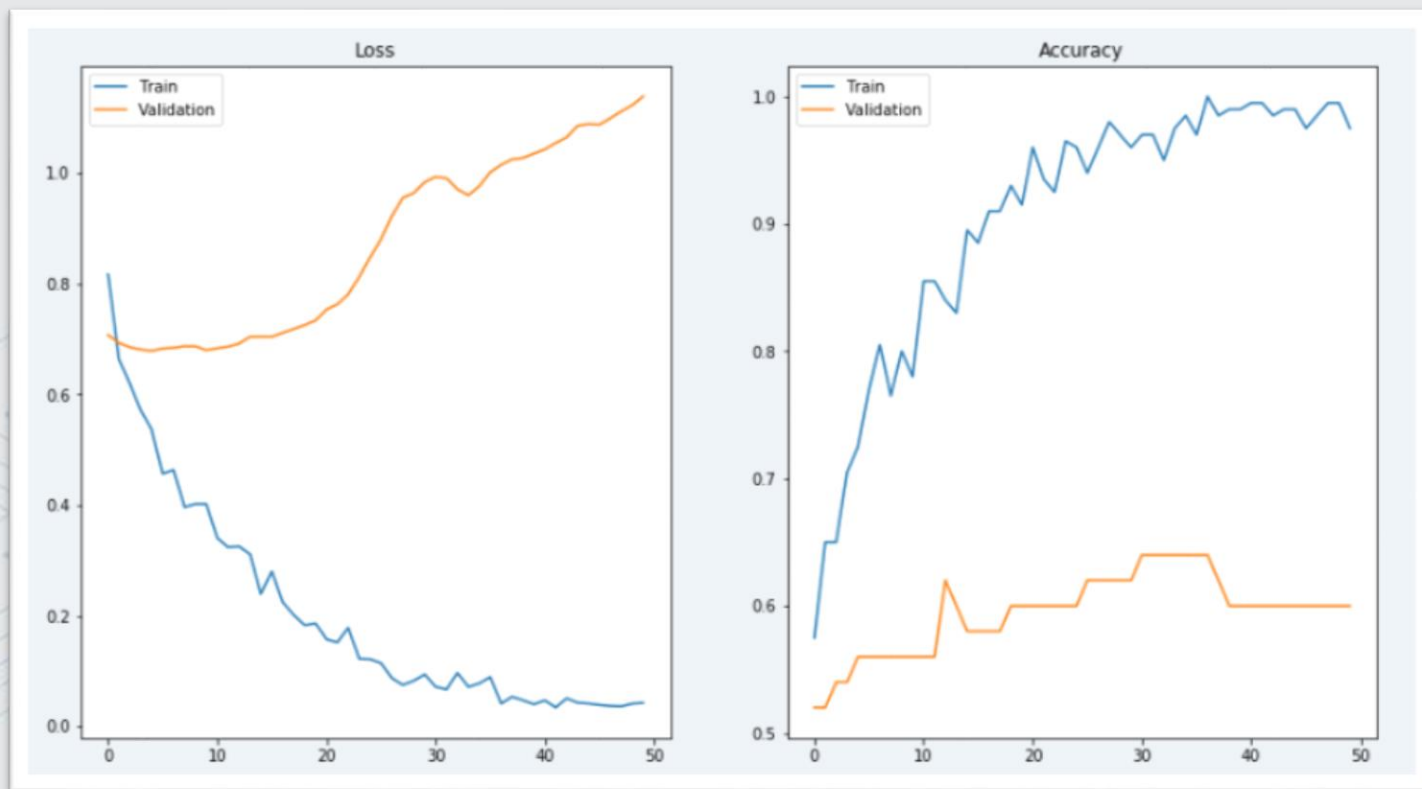




# 過度學習



- 過度學習 ( Overfitting ) 通常是說訓練出來的模型對於訓練集的預測結果很好，但是對於測試集則是不好。也就是說模型的通用能力 ( generalization ) 不好。





# 避免過度學習的方法



- › 增加訓練集資料數量或資料擴增 ( data augmentation )
- › 拿掉影響力小的特徵 ( feature )
- › 資料集進行打亂 ( shuffle )
- › 減少訓練次數 ( epoch )





# 避免過度學習的方法



## › 隨機拋棄 ( Dropout )

在訓練過程中，隨機拿掉一些連結（weight 設為0），只針對某部分的權重進行調整，更能精確調整權重。

## › 正則化 ( Regularization )

在損失函數中添加一個會懲罰大權重值的附加項。

## › 整合 ( Ensembling )

整合多種模型，例如根據多種模型的預測結果用投票法決定最後預測值。

## › 交叉驗證 ( Cross Validation )

將訓練集分割成若干個子樣本集，輪流使用其中一個子樣本集進行驗證，其他則用於訓練，最終整合所有預測結果。



# 修改模型架構



## › 加入Dropout和減少訓練次數

```
27 # 模型建置
28 from keras.models import Sequential
29 from keras.layers import Dense, Dropout
30 # 宣告Sequential循序模型
31 model = Sequential()
32 # 加入輸入層
33 model.add(Dense(512, activation='relu', input_shape=(784,)))
34 # 加入Dropout
35 model.add(Dropout(0.5))
36 # 加入隱藏層
37 model.add(Dense(512, activation='relu'))
38 # 加入Dropout
39 model.add(Dropout(0.5))
40 # 指定輸出層模型
41 model.add(Dense(10, activation='softmax'))
```

53

epochs=10,

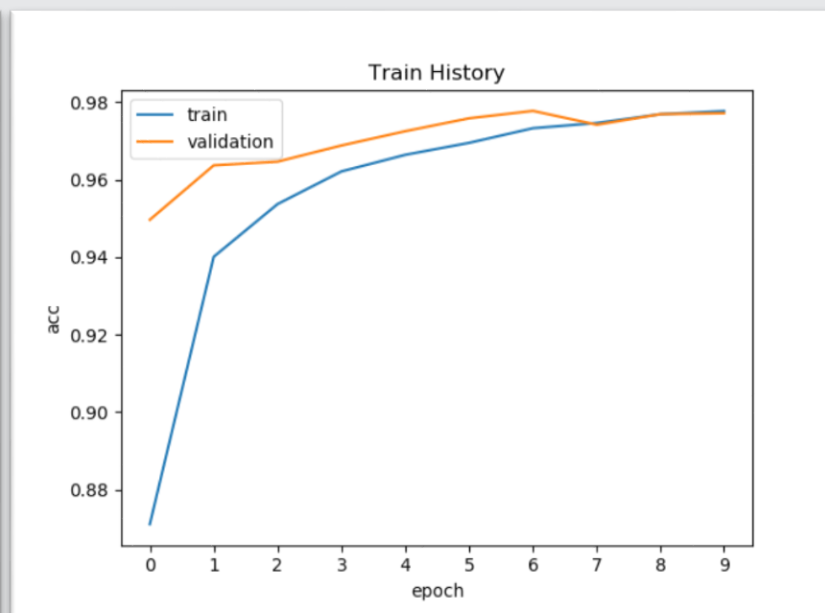
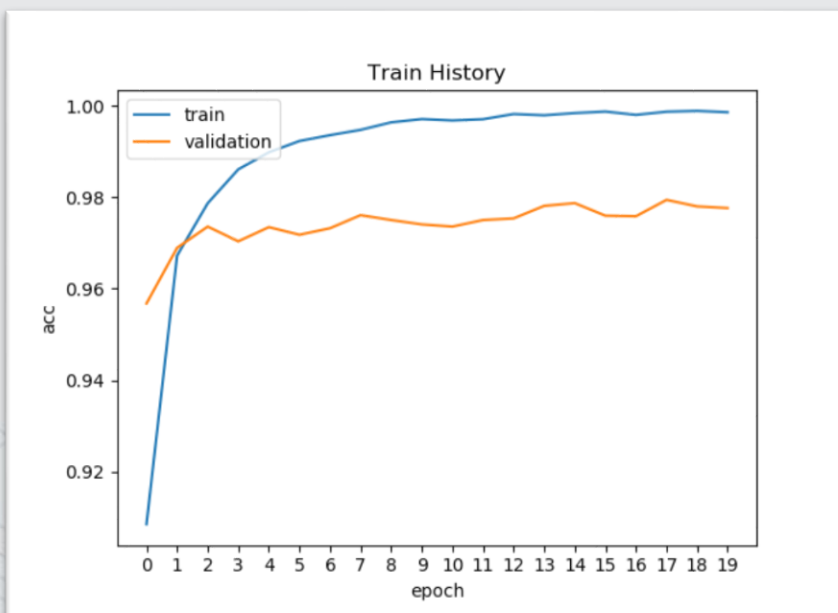


# 模型再評估

機器學習實務



## 訓練歷程比較





# 進行預測



› 將預測結果寫入檔案，上傳Kaggle平台

```
71 # 進行預測
72 submit = 'ImageId,Label\n'
73 with open('test.csv', 'r') as file:
74     csv_lines=file.readlines()
75 image_id = 1
76 for i in range(1, len(csv_lines)):
77     # 去掉換行符號並以逗號分割
78     row = csv_lines[i].replace('\n', '').split(',')
79     # 並將字串轉為整數，正規化並預測
80     result = model.predict_classes(np.array([list(map(int, row))])/255.0)[0]
81     submit += str(image_id) + ',' + str(result) + '\n'
82     image_id += 1
83 # 存成CSV檔
84 open('answer.csv', 'w').write(submit)
```

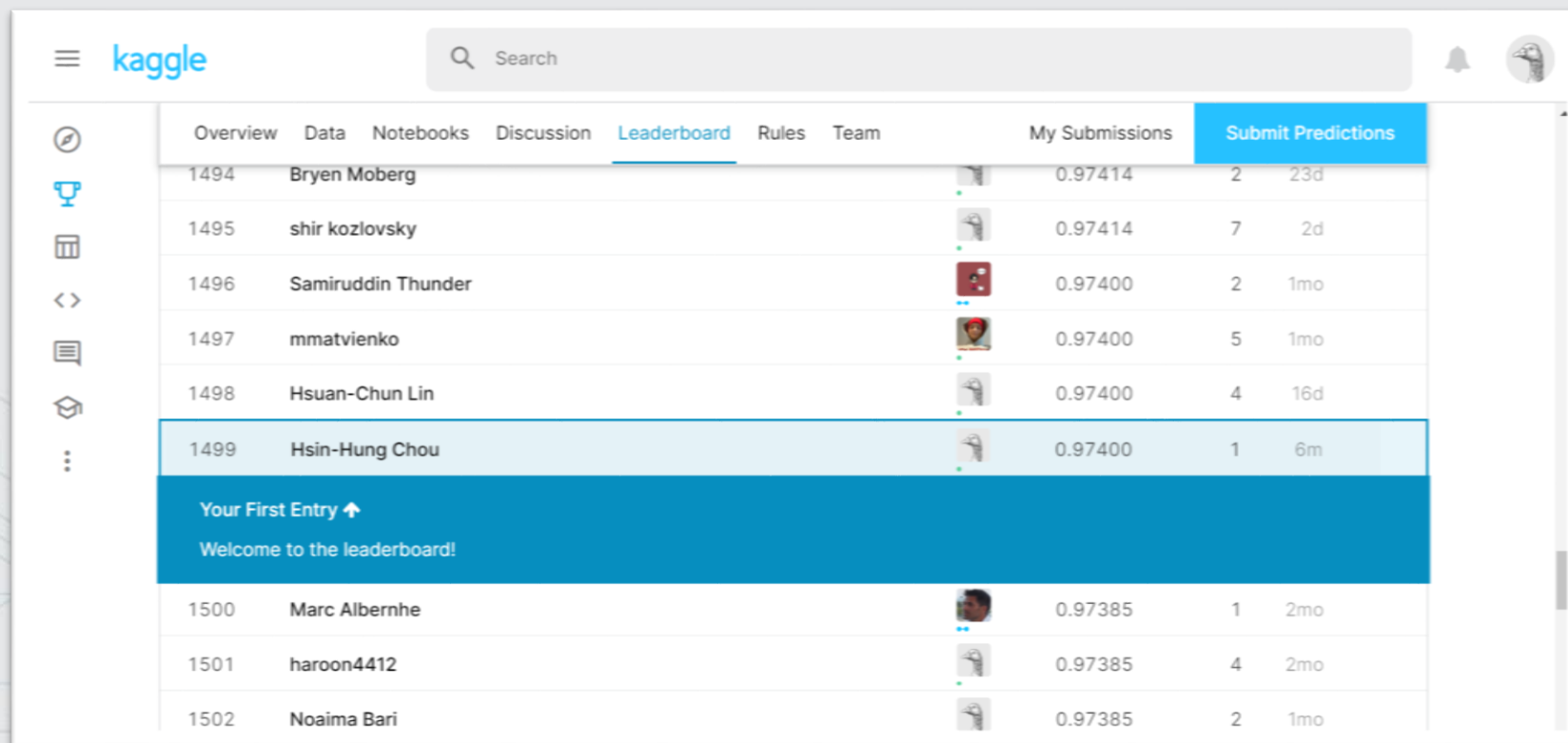


# 進行預測

機器學習實務



› 將預測結果寫入檔案，上傳Kaggle平台



The screenshot shows the Kaggle website's Leaderboard for a specific competition. The interface includes a search bar at the top, navigation tabs (Overview, Data, Notebooks, Discussion, Leaderboard, Rules, Team), and a 'My Submissions' button. The main table lists users with their rank, name, profile picture, score, number of submissions, and time since last submission. The user 'Hsin-Hung Chou' is highlighted as the top performer with a score of 0.97400 and 1 submission 6 minutes ago. Below the table, a blue banner reads 'Your First Entry' and 'Welcome to the leaderboard!'.

	Overview	Data	Notebooks	Discussion	Leaderboard	Rules	Team	My Submissions	Submit Predictions
1494	Bryen Moberg							0.97414	2 23d
1495	shir kozlovsky							0.97414	7 2d
1496	Samiruddin Thunder							0.97400	2 1mo
1497	mmatvienko							0.97400	5 1mo
1498	Hsuan-Chun Lin							0.97400	4 16d
1499	Hsin-Hung Chou							0.97400	1 6m
Your First Entry ↑ Welcome to the leaderboard!									
1500	Marc Albernhe							0.97385	1 2mo
1501	haroon4412							0.97385	4 2mo
1502	Noaima Bari							0.97385	2 1mo