數據分析團隊

Spark 教育訓練

02 Hadoop 與機器學習

數據經營部

李嘉桓

國泰人壽
Cathay Life Insurance

課前準備

教材文件：

\\cxlsvr174\國壽商業分析教育資源(期限20221231)\01 課程資料\2020課程資料\20200818 Spark教育訓練

教材 Gitlab：

http://10.95.42.31:8282/i9h00211/hadoop_spark_practice

課堂聊天室：

https://tlk.io/i9h00

Hadoop 機器學習開發

使用Pyspark、Rspark、MLlib開發
(整個叢集資源)

使用 Python、R、sklearn開發
(Explore node 單機資源)

如果要使用到 Hadoop 的資源，就必須使用相關的語言或套件開發!!

Hadoop 機器學習的痛點

1. Hadoop 是叢集架構，模型訓練會將任務派給各個 worknode，在訊息迭代上沒有單一實體機強，精確度不足，甚至因為 IO 多訓練時間更長，尤其是複雜的神經網路。

2. Spark 有專屬機器學習的套件庫 MLlib，但只有基礎演算法，不支持複雜演算法，如 RNN、LSTM 等，也沒有大量的 activation function 調參，所以實用度待觀察。

3. 機器學習語法、套件與一般常用的 Sklearn 有所差異，學習曲線較高，如 Spark 的類別可以用向量表示進行訓練，對初學者相當不友善。 (尤其是對線性代數不熟的)

那還要學嗎?

非常需要學一下唷！

Hadoop 資料儲存的成本優勢還是在，所以各大廠都在積極解決類似問題，為了更好銜接新興的工具或框架，不會有浪費時間的狀況。

- Intel BigDL、騰訊 Spark Angel、Deeplearning4j 等框架都已經釋出，但版本未穩定。
- Spark Mllib 可以解決基本的模型需求，也可以接 Spark streaming 提高服務流程，未來發展還很難說。
- 最終備案依然可以拿 Spark 進行資料處理，再轉回 Pandas 進行模型運算，至少解決前面資料問題。

分析資料集 ABT

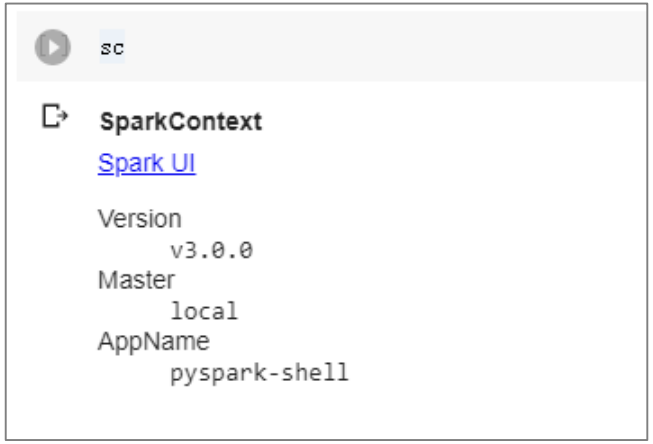
Kaggle 連結：https://www.kaggle.com/c/titanic/data

使用 Kaggle 的經典案例，鐵達尼號資料集做為後續分析的 ABT：

1 使用 Colab，開啟新增筆記本，並安裝 Spark 環境。

```
!pip install pyspark
```

```
from pyspark.sql import SparkSession
from pyspark import SparkContext
spark = SparkSession.builder.master("local").getOrCreate()
sc = SparkContext.getOrCreate()
```



2 讀取資料集。

```
# 下載資料集
from pyspark import SparkFiles
url_train = 'https://raw.githubusercontent.com/chia313339/Spark_practice/master/Titanic/train.csv'
url_test = 'https://raw.githubusercontent.com/chia313339/Spark_practice/master/Titanic/test.csv'
spark.sparkContext.addFile(url_train)
spark.sparkContext.addFile(url_test)

# 讀取資料集
sdf_train = spark.read.csv(SparkFiles.get("train.csv"), header=True, inferSchema= True)
sdf_test = spark.read.csv(SparkFiles.get("test.csv"), header=True, inferSchema= True)
```

分析資料集 ABT

Kaggle 連結：https://www.kaggle.com/c/titanic/data

sdf_train 訓練資料集

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen ...	male	22.0	1	0	A/5 21171	7.25	null	S
2	1	1	Cumings, Mrs. Joh...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. ...	female	26.0	0	0	STON/O2. 3101282	7.925	null	S
4	1	1	Futrelle, Mrs. Ja...	female	35.0	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. Willia...	male	35.0	0	0	373450	8.05	null	S
6	0	3	Moran, Mr. James	male	null	0	0	330877	8.4583	null	Q
7	0	1	McCarthy, Mr. Tim...	male	54.0	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. ...	male	2.0	3	1	349909	21.075	null	S
9	1	3	Johnson, Mrs. Osc...	female	27.0	0	2	347742	11.1333	null	S
10	1	2	Nasser, Mrs. Nich...	female	14.0	1	0	237736	30.0708	null	C
11	1	3	Sandstrom, Miss. ...	female	4.0	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. El...	female	58.0	0	0	113783	26.55	C103	S
13	0	3	Saundercock, Mr. ...	male	20.0	0	0	A/5. 2151	8.05	null	S
14	0	3	Andersson, Mr. An...	male	39.0	1	5	347082	31.275	null	S
15	0	3	Vestrom, Miss. Hu...	female	14.0	0	0	350406	7.8542	null	S
16	1	2	Hewlett, Mrs. (Ma...	female	55.0	0	0	248706	16.0	null	S
17	0	3	Rice, Master. Eugene	male	2.0	4	1	382652	29.125	null	Q
18	1	2	Williams, Mr. Cha...	male	null	0	0	244373	13.0	null	S
19	0	3	Vander Planke, Mr...	female	31.0	1	0	345763	18.0	null	S
20	1	3	Masselmani, Mrs. ...	female	null	0	0	2649	7.225	null	C

only showing top 20 rows

sdf_test 測試資料集

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	null	Q
893	3	Wilkes, Mrs. Jame...	female	47.0	1	0	363272	7.0	null	S
894	2	Myles, Mr. Thomas...	male	62.0	0	0	240276	9.6875	null	Q
895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	null	S
896	3	Hirvonen, Mrs. Al...	female	22.0	1	1	3101298	12.2875	null	S
897	3	Svensson, Mr. Joh...	male	14.0	0	0	7538	9.225	null	S
898	3	Connolly, Miss. Kate	female	30.0	0	0	330972	7.6292	null	Q
899	2	Caldwell, Mr. Alb...	male	26.0	1	1	248738	29.0	null	S
900	3	Abrahim, Mrs. Jos...	female	18.0	0	0	2657	7.2292	null	C
901	3	Davies, Mr. John ...	male	21.0	2	0	A/4 48871	24.15	null	S
902	3	Ilieff, Mr. Ylio	male	null	0	0	349220	7.8958	null	S
903	1	Jones, Mr. Charle...	male	46.0	0	0	694	26.0	null	S
904	1	Snyder, Mrs. John...	female	23.0	1	0	21228	82.2667	B45	S
905	2	Howard, Mr. Benjamin	male	63.0	1	0	24065	26.0	null	S
906	1	Chaffee, Mrs. Her...	female	47.0	1	0	W.E.P. 5734	61.175	E31	S
907	2	del Carlo, Mrs. S...	female	24.0	1	0	SC/PARIS 2167	27.7208	null	C
908	2	Keane, Mr. Daniel	male	35.0	0	0	233734	12.35	null	Q
909	3	Assaf, Mr. Gerios	male	21.0	0	0	2692	7.225	null	C
910	3	Ilmakangas, Miss....	female	27.0	1	0	STON/O2. 3101270	7.925	null	S
911	3	"Assaf Khalil, Mr...	female	45.0	0	0	2696	7.225	null	C

only showing top 20 rows

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

EDA 探索分析 – 視覺化處理

略。

Spark 沒有視覺化套件，建議轉回 Pandas 處理。

EDA 探索分析 – 資料探索

```
# 觀察資料型態
sdf_train.printSchema()
```

```
root
 |-- PassengerId: integer (nullable = true)
 |-- Survived: integer (nullable = true)
 |-- Pclass: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Sex: string (nullable = true)
 |-- Age: double (nullable = true)
 |-- SibSp: integer (nullable = true)
 |-- Parch: integer (nullable = true)
 |-- Ticket: string (nullable = true)
 |-- Fare: double (nullable = true)
 |-- Cabin: string (nullable = true)
 |-- Embarked: string (nullable = true)
```

```
# 順便觀察哪些欄位有缺失值
sdf_train.describe().show()
```

summary	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
count	891	891	891	891	891	714	891	891	891	891	204	889
mean	446.0	0.3838383838383838	2.308641975308642	null	null	29.69911764705882	0.5230078563411896	0.38159371492704824	260318.54916792738	32.2042079685746	null	null
stddev	257.3538420152301	0.4865924542648575 3	0.8360712409770491	null	null	14.526497332334035	1.1027434322934315	0.8060572211299488	471609.26868834975	49.69342859718089	null	null
min	1	0	1	"Andersson, Mr. A...	female	0.42	0	0	110152	0.0	A10	C
max	891	1	3	van Melkebeke, Mr...	male	80.0	8	6	WE/P 5735	512.3292	T	S

缺失值處理

缺失值處理方法很多，先用簡單的假 Dataframe 來說明。

```
# 建一個假的 Dataframe，說明數值及字串的缺失處理
df_tmp = spark.createDataFrame([(1,'Alice',28,None),
                                (2,'QQcat',45,'M'),
                                (3,'Kobe',None,None),
                                (4,'Joan',33,'F'),
                                (5,'Stone',54,'M'),
                                (6,'Ruby',None,'F'),
                                (7,'Ray',42,'M')],
                               ['id', 'Name', 'Age', 'Gender'])
df_tmp.show()
```

```
+---+-----+----+------+
| id| Name| Age|Gender|
+---+-----+----+------+
|  1|Alice|  28|  null|
|  2|QQcat|  45|     M|
|  3| Kobe|null|  null|
|  4| Joan|  33|     F|
|  5|Stone|  54|     M|
|  6| Ruby|null|     F|
|  7|  Ray|  42|     M|
+---+-----+----+------+
```

缺失值處理

```
# 找出有缺失的欄位
from pyspark.sql.functions import when, count, col
df_tmp.select([count(when(col(c).isNull(), c)).alias(c) for c in df_tmp.columns]).show()
```

```
+---+----+---+------+
| id|Name|Age|Gender|
+---+----+---+------+
|  0|   0|  2|     2|
+---+----+---+------+
```

```
# 情境一 移除有缺失的資料
df_tmp.na.drop().show()
```

```
+---+-----+---+------+
| id| Name|Age|Gender|
+---+-----+---+------+
|  2|QQcat| 45|     M|
|  4| Joan| 33|     F|
|  5|Stone| 54|     M|
|  7|  Ray| 42|     M|
+---+-----+---+------+
```

```
# 情境二 移除指定欄位有缺失的資料
df_tmp.na.drop(subset=['Age']).show()
```

```
+---+-----+---+------+
| id| Name|Age|Gender|
+---+-----+---+------+
|  1|Alice| 28|  null|
|  2|QQcat| 45|     M|
|  4| Joan| 33|     F|
|  5|Stone| 54|     M|
|  7|  Ray| 42|     M|
+---+-----+---+------+
```

缺失值處理

```
# 情境三 移除指定欄位中，有任意缺失的資料
df_tmp.na.drop(how='all',subset=['Age','Gender']).show()
```

```
+---+-----+----+------+
| id| Name| Age|Gender|
+---+-----+----+------+
|  1|Alice|  28|  null|
|  2|QQcat|  45|     M|
|  4| Joan|  33|     F|
|  5|Stone|  54|     M|
|  6| Ruby|null|     F|
|  7|  Ray|  42|     M|
+---+-----+----+------+
```

how –

default ‘any’. Determine if row or column is removed from DataFrame, when we have at least one NA or all NA.

- any : If any NA values are present, drop that row or column.
- all : If all values are NA, drop that row or column.

```
# 情境四 取代字串缺失值，spark 會自動判斷字串欄位
df_tmp.na.fill('???').show()
```

```
+---+-----+----+------+
| id| Name| Age|Gender|
+---+-----+----+------+
|  1|Alice|  28|   ???|
|  2|QQcat|  45|     M|
|  3| Kobe|null|   ???|
|  4| Joan|  33|     F|
|  5|Stone|  54|     M|
|  6| Ruby|null|     F|
|  7|  Ray|  42|     M|
+---+-----+----+------+
```

缺失值處理

```
# 情境五 取代數值缺失值，spark 會自動判斷數值欄位
df_tmp.na.fill(999).show()
```

```
+---+-----+---+------+
| id| Name|Age|Gender|
+---+-----+---+------+
|  1|Alice| 28|  null|
|  2|QQcat| 45|     M|
|  3| Kobe|999|  null|
|  4| Joan| 33|     F|
|  5|Stone| 54|     M|
|  6| Ruby|999|     F|
|  7|  Ray| 42|     M|
+---+-----+---+------+
```

```
# 情境六 各自指定補值
df_tmp.na.fill({'Age':999,'Gender':'???'}).show()
```

```
+---+-----+---+------+
| id| Name|Age|Gender|
+---+-----+---+------+
|  1|Alice| 28|   ???|
|  2|QQcat| 45|     M|
|  3| Kobe|999|   ???|
|  4| Joan| 33|     F|
|  5|Stone| 54|     M|
|  6| Ruby|999|     F|
|  7|  Ray| 42|     M|
+---+-----+---+------+
```

缺失值處理

```
# 情境七 統計量補值
from pyspark.sql.functions import mean
mean_age = df_tmp.select(mean(df_tmp.Age)).collect()
mean_age
```

```
[Row(avg(Age)=40.4)]
```

collect()

類似show()、count()的執行運算指令，通常用於向量處理

```
# 年齡缺失值補上平均年齡
df_tmp.na.fill({'Age':mean_age[0][0]}).show()
```

```
+---+-----+---+------+
| id| Name|Age|Gender|
+---+-----+---+------+
|  1|Alice| 28|  null|
|  2|QQcat| 45|     M|
|  3| Kobe| 40|  null|
|  4| Joan| 33|     F|
|  5|Stone| 54|     M|
|  6| Ruby| 40|     F|
|  7|  Ray| 42|     M|
+---+-----+---+------+
```

ABT 缺失處理

```
# 計算每個欄位缺失值數量
from pyspark.sql.functions import isnan, when, count, col
sdf_train.select([count(when(col(c).isNull(), c)).alias(c) for c in sdf_train.columns]).show()
```

```
+-----------+--------+------+----+---+---+-----+-----+------+----+-----+--------+
|PassengerId|Survived|Pclass|Name|Sex|Age|SibSp|Parch|Ticket|Fare|Cabin|Embarked|
+-----------+--------+------+----+---+---+-----+-----+------+----+-----+--------+
|          0|       0|     0|   0|  0|177|    0|    0|     0|   0|  687|       2|
+-----------+--------+------+----+---+---+-----+-----+------+----+-----+--------+
```

```
# 教育訓練時間問題，Age先無腦貼中位數，Cabin有77%都是Null，刪欄位，Embarked只有2筆缺失，刪缺的資料
def handle_missing(dataframe, age_med):
  tmp = dataframe
  tmp = tmp.drop('Cabin').na.drop(subset=['Embarked']).na.fill({'Age':age_med})
  return tmp
```

```
# 算一下Age中位數
age_med = sdf_train.approxQuantile('Age',[0.5],0.25)[0]
age_med
```

approxQuantile(col, probabilities, relativeError)

- col：The name of the numerical column.
- probabilities：A list of quantile probabilities. Each number must belong to [0, 1]. For example 0 is the minimum, 0.5 is the median, 1 is the maximum.
- relativeError：The relative target precision to achieve (>= 0). If set to zero, the exact quantiles are computed, which could be very expensive. Note that values greater than 1 are accepted but give the same result as 1.

ABT 缺失處理

```
# 將資料集套用缺失值處理函數
sdf_train_hadle_missing = handle_missing(sdf_train,age_med)
sdf_test_hadle_missing = handle_missing(sdf_test,age_med)
```

sdf_train_hadle_missing

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
1	0	3	Braund, Mr. Owen ...	male	22.0	1	0	A/5 21171	7.25	S
2	1	1	Cumings, Mrs. Joh...	female	38.0	1	0	PC 17599	71.2833	C
3	1	3	Heikkinen, Miss. ...	female	26.0	0	0	STON/O2. 3101282	7.925	S
4	1	1	Futrelle, Mrs. Ja...	female	35.0	1	0	113803	53.1	S
5	0	3	Allen, Mr. Willia...	male	35.0	0	0	373450	8.05	S
6	0	3	Moran, Mr. James	male	21.0	0	0	330877	8.4583	Q
7	0	1	McCarthy, Mr. Tim...	male	54.0	0	0	17463	51.8625	S
8	0	3	Palsson, Master. ...	male	2.0	3	1	349909	21.075	S
9	1	3	Johnson, Mrs. Osc...	female	27.0	0	2	347742	11.1333	S
10	1	2	Nasser, Mrs. Nich...	female	14.0	1	0	237736	30.0708	C
11	1	3	Sandstrom, Miss. ...	female	4.0	1	1	PP 9549	16.7	S
12	1	1	Bonnell, Miss. El...	female	58.0	0	0	113783	26.55	S
13	0	3	Saundercock, Mr. ...	male	20.0	0	0	A/5. 2151	8.05	S
14	0	3	Andersson, Mr. An...	male	39.0	1	5	347082	31.275	S
15	0	3	Vestrom, Miss. Hu...	female	14.0	0	0	350406	7.8542	S
16	1	2	Hewlett, Mrs. (Ma...	female	55.0	0	0	248706	16.0	S
17	0	3	Rice, Master. Eugene	male	2.0	4	1	382652	29.125	Q
18	1	2	Williams, Mr. Cha...	male	21.0	0	0	244373	13.0	S
19	0	3	Vander Planke, Mr...	female	31.0	1	0	345763	18.0	S
20	1	3	Masselmani, Mrs. ...	female	21.0	0	0	2649	7.225	C

only showing top 20 rows

特徵工程

類別欄位通常會進行 Encode 處理，可以使用 Spark 現有的套件處理。

```
# 以label-encoding為例，Embarked示範，spark比較貼近的函數是StringIndexer()
from pyspark.ml.feature import StringIndexer
string_indexer = StringIndexer(inputCol = 'Embarked', outputCol = 'Embarked_StringIndexer')
sdf_train_LE = string_indexer.fit(sdf_train_hadle_missing).transform(sdf_train_hadle_missing)
sdf_train_LE.show()
```

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+--------+----------------------+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|   Fare|Embarked|Embarked_StringIndexer|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+--------+----------------------+
|          1|       0|     3|Braund, Mr. Owen ...|  male|22.0|    1|    0|       A/5 21171|   7.25|       S|                   0.0|
|          2|       1|     1|Cumings, Mrs. Joh...|female|38.0|    1|    0|        PC 17599|71.2833|       C|                   1.0|
|          3|       1|     3|Heikkinen, Miss. ...|female|26.0|    0|    0|STON/O2. 3101282|  7.925|       S|                   0.0|
|          4|       1|     1|Futrelle, Mrs. Ja...|female|35.0|    1|    0|          113803|   53.1|       S|                   0.0|
|          5|       0|     3|Allen, Mr. Willia...|  male|35.0|    0|    0|          373450|   8.05|       S|                   0.0|
|          6|       0|     3|    Moran, Mr. James|  male|21.0|    0|    0|          330877| 8.4583|       Q|                   2.0|
|          7|       0|     1|McCarthy, Mr. Tim...|  male|54.0|    0|    0|           17463|51.8625|       S|                   0.0|
|          8|       0|     3|Palsson, Master. ...|  male| 2.0|    3|    1|          349909| 21.075|       S|                   0.0|
|          9|       1|     3|Johnson, Mrs. Osc...|female|27.0|    0|    2|          347742|11.1333|       S|                   0.0|
|         10|       1|     2|Nasser, Mrs. Nich...|female|14.0|    1|    0|          237736|30.0708|       C|                   1.0|
|         11|       1|     3|Sandstrom, Miss. ...|female| 4.0|    1|    1|         PP 9549|   16.7|       S|                   0.0|
|         12|       1|     1|Bonnell, Miss. El...|female|58.0|    0|    0|          113783|  26.55|       S|                   0.0|
|         13|       0|     3|Saundercock, Mr. ...|  male|20.0|    0|    0|       A/5. 2151|   8.05|       S|                   0.0|
|         14|       0|     3|Andersson, Mr. An...|  male|39.0|    1|    5|          347082| 31.275|       S|                   0.0|
|         15|       0|     3|Vestrom, Miss. Hu...|female|14.0|    0|    0|          350406| 7.8542|       S|                   0.0|
|         16|       1|     2|Hewlett, Mrs. (Ma...|female|55.0|    0|    0|          248706|   16.0|       S|                   0.0|
|         17|       0|     3|Rice, Master. Eugene|  male| 2.0|    4|    1|          382652| 29.125|       Q|                   2.0|
|         18|       1|     2|Williams, Mr. Cha...|  male|21.0|    0|    0|          244373|   13.0|       S|                   0.0|
|         19|       0|     3|Vander Planke, Mr...|female|31.0|    1|    0|          345763|   18.0|       S|                   0.0|
|         20|       1|     3|Masselmani, Mrs. ...|female|21.0|    0|    0|            2649|  7.225|       C|                   1.0|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+--------+----------------------+
only showing top 20 rows
```

特徵工程

下一頁開始內容可能感到不適，請先做好心理準備

特徵工程

Spark 的 one-hot-encoding 需要從前面的 label-encoding 轉，而且會是向量格式存在。

```
# 以one-hot-encoding為例，Embarked示範，spark比較貼近的函數是OneHotEncoder()
from pyspark.ml.feature import OneHotEncoder
one_hot_encoder = OneHotEncoder(inputCol = 'Embarked_StringIndexer', outputCol = 'Embarked_OneHotEncoder')
sdf_train_OE = one_hot_encoder.fit(sdf_train_LE).transform(sdf_train_LE)
sdf_train_OE.show()
```

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+--------+----------------------+----------------------+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|   Fare|Embarked|Embarked_StringIndexer|Embarked_OneHotEncoder|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+--------+----------------------+----------------------+
|          1|       0|     3|Braund, Mr. Owen ...|  male|22.0|    1|    0|       A/5 21171|   7.25|       S|                   0.0|         (2,[0],[1.0])|
|          2|       1|     1|Cumings, Mrs. Joh...|female|38.0|    1|    0|        PC 17599|71.2833|       C|                   1.0|         (2,[1],[1.0])|
|          3|       1|     3|Heikkinen, Miss. ...|female|26.0|    0|    0|STON/O2. 3101282|  7.925|       S|                   0.0|         (2,[0],[1.0])|
|          4|       1|     1|Futrelle, Mrs. Ja...|female|35.0|    1|    0|          113803|   53.1|       S|                   0.0|         (2,[0],[1.0])|
|          5|       0|     3|Allen, Mr. Willia...|  male|35.0|    0|    0|          373450|   8.05|       S|                   0.0|         (2,[0],[1.0])|
|          6|       0|     3|    Moran, Mr. James|  male|21.0|    0|    0|          330877| 8.4583|       Q|                   2.0|             (2,[],[])|
|          7|       0|     1|McCarthy, Mr. Tim...|  male|54.0|    0|    0|           17463|51.8625|       S|                   0.0|         (2,[0],[1.0])|
|          8|       0|     3|Palsson, Master. ...|  male| 2.0|    3|    1|          349909| 21.075|       S|                   0.0|         (2,[0],[1.0])|
|          9|       1|     3|Johnson, Mrs. Osc...|female|27.0|    0|    2|          347742|11.1333|       S|                   0.0|         (2,[0],[1.0])|
|         10|       1|     2|Nasser, Mrs. Nich...|female|14.0|    1|    0|          237736|30.0708|       C|                   1.0|         (2,[1],[1.0])|
|         11|       1|     3|Sandstrom, Miss. ...|female| 4.0|    1|    1|         PP 9549|   16.7|       S|                   0.0|         (2,[0],[1.0])|
|         12|       1|     1|Bonnell, Miss. El...|female|58.0|    0|    0|          113783|  26.55|       S|                   0.0|         (2,[0],[1.0])|
|         13|       0|     3|Saundercock, Mr. ...|  male|20.0|    0|    0|       A/5. 2151|   8.05|       S|                   0.0|         (2,[0],[1.0])|
|         14|       0|     3|Andersson, Mr. An...|  male|39.0|    1|    5|          347082| 31.275|       S|                   0.0|         (2,[0],[1.0])|
|         15|       0|     3|Vestrom, Miss. Hu...|female|14.0|    0|    0|          350406| 7.8542|       S|                   0.0|         (2,[0],[1.0])|
|         16|       1|     2|Hewlett, Mrs. (Ma...|female|55.0|    0|    0|          248706|   16.0|       S|                   0.0|         (2,[0],[1.0])|
|         17|       0|     3|Rice, Master. Eugene|  male| 2.0|    4|    1|          382652| 29.125|       Q|                   2.0|             (2,[],[])|
|         18|       1|     2|Williams, Mr. Cha...|  male|21.0|    0|    0|          244373|   13.0|       S|                   0.0|         (2,[0],[1.0])|
|         19|       0|     3|Vander Planke, Mr...|female|31.0|    1|    0|          345763|   18.0|       S|                   0.0|         (2,[0],[1.0])|
|         20|       1|     3|Masselmani, Mrs. ...|female|21.0|    0|    0|            2649|  7.225|       C|                   1.0|         (2,[1],[1.0])|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+--------+----------------------+----------------------+
only showing top 20 rows
```

特徵工程

Spark 的特徵工程函數還有很多，可以從官方文件查詢，不多做贅述。

官方文件檔：https://spark.apache.org/docs/latest/ml-guide.html

- VectorAssembler
- MinMaxScaler (需要先轉向量)
- Normalizer (需要先轉向量)
- StandardScaler (需要先轉向量)
- PCA、DCT
- TF-IDF
- Word2Vec

Spark pipline

Spark pipline 是機器學習一個很方便的套件，可以快速套用我們想做的事情到資料集之中。

```
# Spark pipline 可以先把要做的特徵工程及模型訓練設定好，執行時會自動派發到各個節點運算
from pyspark.ml.feature import StringIndexer, OneHotEncoder
# 先設定要轉的類別、數值欄位
string_features = ['Sex','Embarked']


# 設定要進行的特徵工程內容
work = []
# label-encoding (one-hot的前置作業)
string_indexer = [StringIndexer(inputCol=column, outputCol=column+'_StringIndexer') for column in string_features]
# one-hot-encoding
one_hot_encoder = [OneHotEncoder(inputCols = [column+'_StringIndexer' for column in string_features], \
outputCols = [column+'_OneHotEncoderEstimator' for column in string_features])]

work += string_indexer
work += one_hot_encoder
work
```

```
[StringIndexer_ffdb4c806794,
 StringIndexer_9cb437c83b86,
 OneHotEncoder_fd54b6d5d845]
```

Spark pipline

pipline 也能確保 Traindata 及 Testdata 的特徵工程處理一致。

```
# 把工作內容加入pipline中
from pyspark.ml import Pipeline
pipline = Pipeline(stages=work)


FE = pipline.fit(sdf_train_hadle_missing)
sdf_transformed_train = FE.transform(sdf_train_hadle_missing)
sdf_transformed_test = FE.transform(sdf_test_hadle_missing)


sdf_transformed_train.show()
```

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+--------+----------------+---------------------+--------------------------+------------------------------+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|   Fare|Embarked|Sex_StringIndexer|Embarked_StringIndexer|Sex_OneHotEncoderEstimator|Embarked_OneHotEncoderEstimator|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+--------+----------------+---------------------+--------------------------+------------------------------+
|          1|       0|     3|Braund, Mr. Owen ...|  male|22.0|    1|    0|       A/5 21171|   7.25|       S|             0.0|                  0.0|             (1,[0],[1.0])|                 (2,[0],[1.0])|
|          2|       1|     1|Cumings, Mrs. Joh...|female|38.0|    1|    0|        PC 17599|71.2833|       C|             1.0|                  1.0|                 (1,[],[])|                 (2,[1],[1.0])|
|          3|       1|     3|Heikkinen, Miss. ...|female|26.0|    0|    0|STON/O2. 3101282|  7.925|       S|             1.0|                  0.0|                 (1,[],[])|                 (2,[0],[1.0])|
|          4|       1|     1|Futrelle, Mrs. Ja...|female|35.0|    1|    0|          113803|   53.1|       S|             1.0|                  0.0|                 (1,[],[])|                 (2,[0],[1.0])|
|          5|       0|     3|Allen, Mr. Willia...|  male|35.0|    0|    0|          373450|   8.05|       S|             0.0|                  0.0|             (1,[0],[1.0])|                 (2,[0],[1.0])|
|          6|       0|     3|    Moran, Mr. James|  male|21.0|    0|    0|          330877| 8.4583|       Q|             0.0|                  2.0|             (1,[0],[1.0])|                     (2,[],[])|
|          7|       0|     1|McCarthy, Mr. Tim...|  male|54.0|    0|    0|           17463|51.8625|       S|             0.0|                  0.0|             (1,[0],[1.0])|                 (2,[0],[1.0])|
|          8|       0|     3|Palsson, Master. ...|  male| 2.0|    3|    1|          349909| 21.075|       S|             0.0|                  0.0|             (1,[0],[1.0])|                 (2,[0],[1.0])|
|          9|       1|     3|Johnson, Mrs. Osc...|female|27.0|    0|    2|          347742|11.1333|       S|             1.0|                  0.0|                 (1,[],[])|                 (2,[0],[1.0])|
|         10|       1|     2|Nasser, Mrs. Nich...|female|14.0|    1|    0|          237736|30.0708|       C|             1.0|                  1.0|                 (1,[],[])|                 (2,[1],[1.0])|
|         11|       1|     3|Sandstrom, Miss. ...|female| 4.0|    1|    1|         PP 9549|   16.7|       S|             1.0|                  0.0|                 (1,[],[])|                 (2,[0],[1.0])|
|         12|       1|     1|Bonnell, Miss. El...|female|58.0|    0|    0|          113783|  26.55|       S|             1.0|                  0.0|                 (1,[],[])|                 (2,[0],[1.0])|
|         13|       0|     3|Saundercock, Mr. ...|  male|20.0|    0|    0|       A/5. 2151|   8.05|       S|             0.0|                  0.0|             (1,[0],[1.0])|                 (2,[0],[1.0])|
|         14|       0|     3|Andersson, Mr. An...|  male|39.0|    1|    5|          347082| 31.275|       S|             0.0|                  0.0|             (1,[0],[1.0])|                 (2,[0],[1.0])|
|         15|       0|     3|Vestrom, Miss. Hu...|female|14.0|    0|    0|          350406| 7.8542|       S|             1.0|                  0.0|                 (1,[],[])|                 (2,[0],[1.0])|
|         16|       1|     2|Hewlett, Mrs. (Ma...|female|55.0|    0|    0|          248706|   16.0|       S|             1.0|                  0.0|                 (1,[],[])|                 (2,[0],[1.0])|
|         17|       0|     3|Rice, Master. Eugene|  male| 2.0|    4|    1|          382652| 29.125|       Q|             0.0|                  2.0|             (1,[0],[1.0])|                     (2,[],[])|
|         18|       1|     2|Williams, Mr. Cha...|  male|21.0|    0|    0|          244373|   13.0|       S|             0.0|                  0.0|             (1,[0],[1.0])|                 (2,[0],[1.0])|
|         19|       0|     3|Vander Planke, Mr...|female|31.0|    1|    0|          345763|   18.0|       S|             1.0|                  0.0|                 (1,[],[])|                 (2,[0],[1.0])|
|         20|       1|     3|Masselmani, Mrs. ...|female|21.0|    0|    0|            2649|  7.225|       C|             1.0|                  1.0|                 (1,[],[])|                 (2,[1],[1.0])|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+--------+----------------+---------------------+--------------------------+------------------------------+
only showing top 20 rows
```

Spark 模型訓練

Spark MLlib 的 Feature 進模型之前，都要轉成向量格式，以隨機森林演算法為例：

```
# 透過pipline處理特徵轉向量及RandomForest訓練
from pyspark.ml.feature import  VectorAssembler
from pyspark.ml.classification import RandomForestClassifier
features = ['Pclass','Age','SibSp','Parch','Fare','Sex_OneHotEncoderEstimator','Embarked_OneHotEncoderEstimator']
vector_assembler = VectorAssembler(inputCols=features, outputCol='Features_Vec')
rf = RandomForestClassifier(labelCol = 'Survived',
                            featuresCol = 'Features_Vec',
                            numTrees = 100,
                            maxDepth = 4,
                            maxBins = 1000)
work = vector_assembler,rf
pipeline = Pipeline(stages=work)
```

```
# 模型訓練並套用在測試資料上
model = pipeline.fit(sdf_transformed_train)
sdf_predict = model.transform(sdf_transformed_test)
```

Spark 模型訓練

預測的結果會顯示再最後一個欄位，可以用 show() 觀察預測結果。

```
sdf_predict.show()
```

```
+------------------+------------------+----------+
|     rawPrediction|       probability|prediction|
+------------------+------------------+----------+
|[86.3504679976785...|[0.86350467997678...|       0.0|
|[59.9318147157009...|[0.59931814715700...|       0.0|
|[84.1030751695053...|[0.84103075169505...|       0.0|
|[85.7219097462057...|[0.85721909746205...|       0.0|
|[48.3771055712067...|[0.48377105571206...|       1.0|
|[83.5808828657031...|[0.83580882865703...|       0.0|
|[45.4411286774513...|[0.45441128677451...|       1.0|
|[68.5928932924063...|[0.68592893292406...|       0.0|
|[37.8788414315967...|[0.37878841431596...|       1.0|
|[82.8936650477932...|[0.82893665047793...|       0.0|
|[86.8104019513008...|[0.86810401951300...|       0.0|
|[73.4754272759421...|[0.73475427275942...|       0.0|
|[9.64061950255030...|[0.09640619502550...|       1.0|
|[77.8899886428613...|[0.77889988642861...|       0.0|
|[10.9951854321911...|[0.10995185432191...|       1.0|
|[15.9835012101962...|[0.15983501210196...|       1.0|
|[81.6012417728795...|[0.81601241772879...|       0.0|
|[84.6076486733037...|[0.84607648673303...|       0.0|
|[52.8347098379208...|[0.52834709837920...|       0.0|
|[48.4732318974300...|[0.48473231897430...|       1.0|
+------------------+------------------+----------+
```

MLlib 支援演算法

- Classification
  - Logistic regression
    - Binomial logistic regression
    - Multinomial logistic regression
  - Decision tree classifier
  - Random forest classifier
  - Gradient-boosted tree classifier
  - Multilayer perceptron classifier
  - Linear Support Vector Machine
  - One-vs-Rest classifier (a.k.a. One-vs-All)
  - Naive Bayes
  - Factorization machines classifier
- Regression
  - Linear regression
  - Generalized linear regression
    - Available families
  - Decision tree regression
  - Random forest regression
  - Gradient-boosted tree regression
  - Survival regression
  - Isotonic regression
  - Factorization machines regressor

資料集切割

Spark 資料集切割可以使用 randomSplit()，從訓練資料切出 validation data 協助訓練，以鐵達尼號的資料為例，切出 traindata, validata 可以協助驗證模型成效。

```
# randomSplit()後面帶資料集切割比例
(traindata, validata) = sdf_transformed_train.randomSplit([0.8, 0.2])
```

```
print(sdf_transformed_train.count())
print(traindata.count())
print(validata.count())
```

```
889
724
165
```

模型成效評估

延伸上面的資料集再建一次模型，並使用 validata 驗證模型成效。

```
# 改用logistic regression訓練
from pyspark.ml.feature import  VectorAssembler
from pyspark.ml.classification import LogisticRegression
features = ['Pclass','Age','SibSp','Parch','Fare','Sex_OneHotEncoderEstimator','Embarked_OneHotEncoderEstimator']
vector_assembler = VectorAssembler(inputCols=features, outputCol='Features_Vec')
lr = LogisticRegression(labelCol='Survived', featuresCol = 'Features_Vec')

work = vector_assembler,lr
pipeline = Pipeline(stages=work)


# 模型訓練並套用在validata資料上
lr_model = pipeline.fit(traindata)
lr_predict = lr_model.transform(validata)



# 顯示validata預測結果
lr_predict.show()
```

```
+--------------------+--------------------+----------+
|       rawPrediction|         probability|prediction|
+--------------------+--------------------+----------+
|[-0.3987603195323...|[0.40161022284638...|       1.0|
|[-2.1665577605142...|[0.10279406861928...|       1.0|
|[-0.2095557558205...|[0.44780193851710...|       1.0|
|[-1.0627912111812...|[0.25677641204364...|       1.0|
|[-0.0048273857849...|[0.49879315589742...|       1.0|
|[0.02384619285974...|[0.50596126573264...|       0.0|
|[-0.8551305874932...|[0.29835770842850...|       1.0|
|[1.68912004780837...|[0.84410840254757...|       0.0|
|[1.96138067690071...|[0.87668229567941...|       0.0|
|[0.71866040347377...|[0.67231196059983...|       0.0|
|[-1.4437368520846...|[0.19096734252237...|       1.0|
|[0.77359464890677...|[0.68429797500904...|       0.0|
|[2.88150569064617...|[0.94692458816655...|       0.0|
|[0.95473878068477...|[0.72206718578785...|       0.0|
|[2.03618321056581...|[0.88454404673121...|       0.0|
|[0.26415790110451...|[0.56565811990135...|       0.0|
|[2.14429178855657...|[0.89513416009134...|       0.0|
|[-0.6137429515256...|[0.35120585359817...|       1.0|
|[-1.2194857322619...|[0.22802696480480...|       1.0|
|[2.64925863014711...|[0.93396528187291...|       0.0|
+--------------------+--------------------+----------+
```

模型成效評估

使用幾個常見指標進行評估，可以使用 MulticlassClassificationEvaluator() 處理分類問題。

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
df_eval = lr_predict.select('prediction','Survived')
eval_acc = MulticlassClassificationEvaluator(labelCol="Survived", predictionCol="prediction", metricName="accuracy")
eval_f1 = MulticlassClassificationEvaluator(labelCol="Survived", predictionCol="prediction", metricName="f1")
eval_pre = MulticlassClassificationEvaluator(labelCol="Survived", predictionCol="prediction", metricName="weightedPrecision")
eval_recall = MulticlassClassificationEvaluator(labelCol="Survived", predictionCol="prediction", metricName="weightedRecall")
```

```
print('pyspark accuracy: %.6f' %eval_acc.evaluate(df_eval))
print('pyspark f1-score: %.6f' %eval_f1.evaluate(df_eval))
print('pyspark precision: %.6f' %eval_pre.evaluate(df_eval))
print('pyspark recall: %.6f' %eval_recall.evaluate(df_eval))
```

```
pyspark accuracy: 0.781818
pyspark f1-score: 0.781079
pyspark precision: 0.780577
pyspark recall: 0.781818
```

進階 - Spark 評估的小痛點

Spark 的評估指標對比 sklearn 的 metrics()，average 參數會默認 weighted，對於較資深的資料科學家需要調整成 macro-average、micro-average 支援性較差。

```
from sklearn import metrics
pd_eval = df_eval.toPandas()
sklern_acc = metrics.accuracy_score(pd_eval['Survived'], pd_eval['prediction'])
sklern_f1 = metrics.f1_score(pd_eval['Survived'], pd_eval['prediction'], average='weighted')
sklern_pre = metrics.precision_score(pd_eval['Survived'], pd_eval['prediction'], average='weighted')
sklern_recall = metrics.recall_score(pd_eval['Survived'], pd_eval['prediction'], average='weighted')

print('sklearn accuracy: %.6f' %sklern_acc)
print('sklearn f1-score: %.6f' %sklern_f1)
print('sklearn precision: %.6f' %sklern_pre)
print('sklearn recall: %.6f' %sklern_recall)
```

```
sklearn accuracy: 0.781818
sklearn f1-score: 0.781079
sklearn precision: 0.780577
sklearn recall: 0.781818
```

聽不懂沒關係，Dataframe 轉回 Pandas 操作就好了

演算法套用練習

1

嘗試將演算法換成 Gradient-Boosted Trees (GBTs) 重新訓練，並使用 validata 驗證模型成效，參數帶 input、output 即可。

參考連結：https://spark.apache.org/docs/latest/ml-classification-regression.html#gradient-boosted-tree-classifier

演算法套用練習

1 嘗試將演算法換成 Gradient-Boosted Trees (GBTs) 重新訓練，並使用 validata 驗證模型成效，參數帶 input、output 即可。

```
# 演算法改GBT
from pyspark.ml.feature import  VectorAssembler
from pyspark.ml.classification import GBTClassifier
features = ['Pclass','Age','SibSp','Parch','Fare','Sex_OneHotEncoderEstimator','Embarked_OneHotEncoderEstimator']
vector_assembler = VectorAssembler(inputCols=features, outputCol='Features_Vec')
GBT = GBTClassifier(labelCol='Survived', featuresCol = 'Features_Vec')

work = vector_assembler,GBT
pipeline = Pipeline(stages=work)

# 模型訓練並套用在validata資料上
GBT_model = pipeline.fit(traindata)
GBT_predict = GBT_model.transform(validata)


# 顯示validata預測結果
GBT_predict.show()
```

```
+--------------------+--------------------+----------+
|       rawPrediction|         probability|prediction|
+--------------------+--------------------+----------+
|[0.38616076361649...|[0.68402287260378...|       0.0|
|[-1.5923724739613...|[0.03974385044712...|       1.0|
|[-0.3118950491752...|[0.34891994425304...|       1.0|
|[0.36936544274392...|[0.67671827352181...|       0.0|
|[0.08310359667713...|[0.54145640672995...|       0.0|
|[0.01569345323789...|[0.50784608250675...|       0.0|
|[-0.1009171437247...|[0.44971202776971...|       1.0|
|[1.00311174048642...|[0.88144895657862...|       0.0|
|[0.00153446135287...|[0.50076723007427...|       0.0|
|[0.34968027456272...|[0.66804598243309...|       0.0|
|[-1.2475514878738...|[0.07620219346105...|       1.0|
|[0.04559681831989...|[0.52278262246017...|       0.0|
|[1.05343988080761...|[0.89157005911830...|       0.0|
|[1.02039216588358...|[0.88501310950892...|       0.0|
|[0.96575408382731...|[0.87341625727131...|       0.0|
|[-1.4106300616334...|[0.05618607286315...|       1.0|
|[1.09156386921046...|[0.89872411122696...|       0.0|
|[0.51065364967064...|[0.73522716729618...|       0.0|
|[-1.3793923781191...|[0.05959243322683...|       1.0|
|[1.42441089246886...|[0.94525774738564...|       0.0|
+--------------------+--------------------+----------+
```

演算法套用練習

1 嘗試將演算法換成 Gradient-Boosted Trees (GBTs) 重新訓練，並使用 validata 驗證模型成效，參數帶 input、output 即可。

```
# Spark 模型評估指標
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
df_eval = GBT_predict.select('prediction','Survived')
eval_acc = MulticlassClassificationEvaluator(labelCol="Survived", predictionCol="prediction", metricName="accuracy")
eval_f1 = MulticlassClassificationEvaluator(labelCol="Survived", predictionCol="prediction", metricName="f1")
eval_pre = MulticlassClassificationEvaluator(labelCol="Survived", predictionCol="prediction", metricName="weightedPrecision")
eval_recall = MulticlassClassificationEvaluator(labelCol="Survived", predictionCol="prediction", metricName="weightedRecall")

print('pyspark accuracy: %.6f' %eval_acc.evaluate(df_eval))
print('pyspark f1-score: %.6f' %eval_f1.evaluate(df_eval))
print('pyspark precision: %.6f' %eva_pre.evaluate(df_eval))
print('pyspark recall: %.6f' %eval_recall.evaluate(df_eval))
```

```
pyspark accuracy: 0.836364
pyspark f1-score: 0.834157
pyspark precision: 0.835207
pyspark recall: 0.836364
```

部門常用演算法

XGBoost

xgboost4j-spark 已釋出，但只能自行從 git 安裝且要設定 Java 環境變數，短時間內無懶人包。

LightGBM

微軟特化的 MMLSpark 已釋出，但 bug 炸裂多，github 上的 issue 多到滿出來還沒人回答，短時間內無懶人包。

spark-sklearn

特化版的 sklearn，專門為 spark 設計，可以直接套用 sklearn 的函數，但目前釋出版本 bug 沒有最多，只有更多，建議先不要用。

END