Data Mining HW5 Spark

大前提:

不知道要怎麼做:請看書籍作者的網站 (http://pythonsparkhadoop.blogspot.com/)

> 要安裝的東西有Oracle VirtualBox、Linux (Ubuntu)、JDK、Python、Hadoop、Scala、Spark、Anaconda...族繁不及備載，因此**版號**務必再三確認，不要ubuntu裝22、java裝6、hadoop裝奇奇怪怪的版本，能舊盡量舊。

以下僅提供參考(按照安裝順序)
Oracle VirtualBox: 7.0.4
Linux Ubuntu: 14.04 LTS(long term support)
JDK(Java): 1.7.0
Python: 2.7.11 >> Python 2
Hadoop: 2.6.4
Scala: 2.11.8
Spark: 2.1.3
Anaconda: 2.5.0

```
hduser@master:~$ java --version
Unrecognized option: --version
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
hduser@master:~$ java -version
java version "1.7.0_201"
OpenJDK Runtime Environment (IcedTea 2.6.17) (7u211-2.6.17-0ubuntu0.1)
OpenJDK 64-Bit Server VM (build 24.201-b00, mixed mode)
hduser@master:~$ python --version
Python 2.7.11 :: Anaconda 2.5.0 (64-bit)
hduser@master:~$ spark
The program 'spark' is currently not installed. You can install it by typing:
sudo apt-get install spark
hduser@master:~$ pyspark --version
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.1.3
      /_/

Using Scala version 2.11.8, OpenJDK 64-Bit Server VM, 1.7.0_201
Branch
Compiled by user  on 2018-06-26T16:37:32Z
Revision
Url
Type --help for more information.
hduser@master:~$ hadoop -version
Error: No command named `-version' was found. Perhaps you meant `hadoop version'
hduser@master:~$ hadoop version
Hadoop 2.6.4
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r 5082c73637530b0
b7e115f9625ed7fac69f937e6
Compiled by jenkins on 2016-02-12T09:45Z
Compiled with protoc 2.5.0
From source with checksum 8dee2286ecdbbbc930a6c87b65cbc010
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-2
.6.4.jar
hduser@master:~$ SS
```

如果搞不懂這些東西的關係，以下我快速介紹:
當你資料量很大的時候，會需要分散式運算，Hadoop是一種支援此類情況的工具，我們使用virtualbox建立虛擬機(VM)模擬多台機器分散運算，但hadoop原生是用Java跑，所以需要Scala、Spark、Python來寫，其中有個特殊的資料結構RDD，你需要把原本運算的data轉RDD。

原則上scala語法跟python很像，你在python用的pandas，在pyspark中就是使用類似SQL、pandas的語法，python2與3我想在此次作業的語法差異應該幾乎沒有。

安裝流程提醒

除了書本上的流程外，以下有些小建議可以讓你更順利

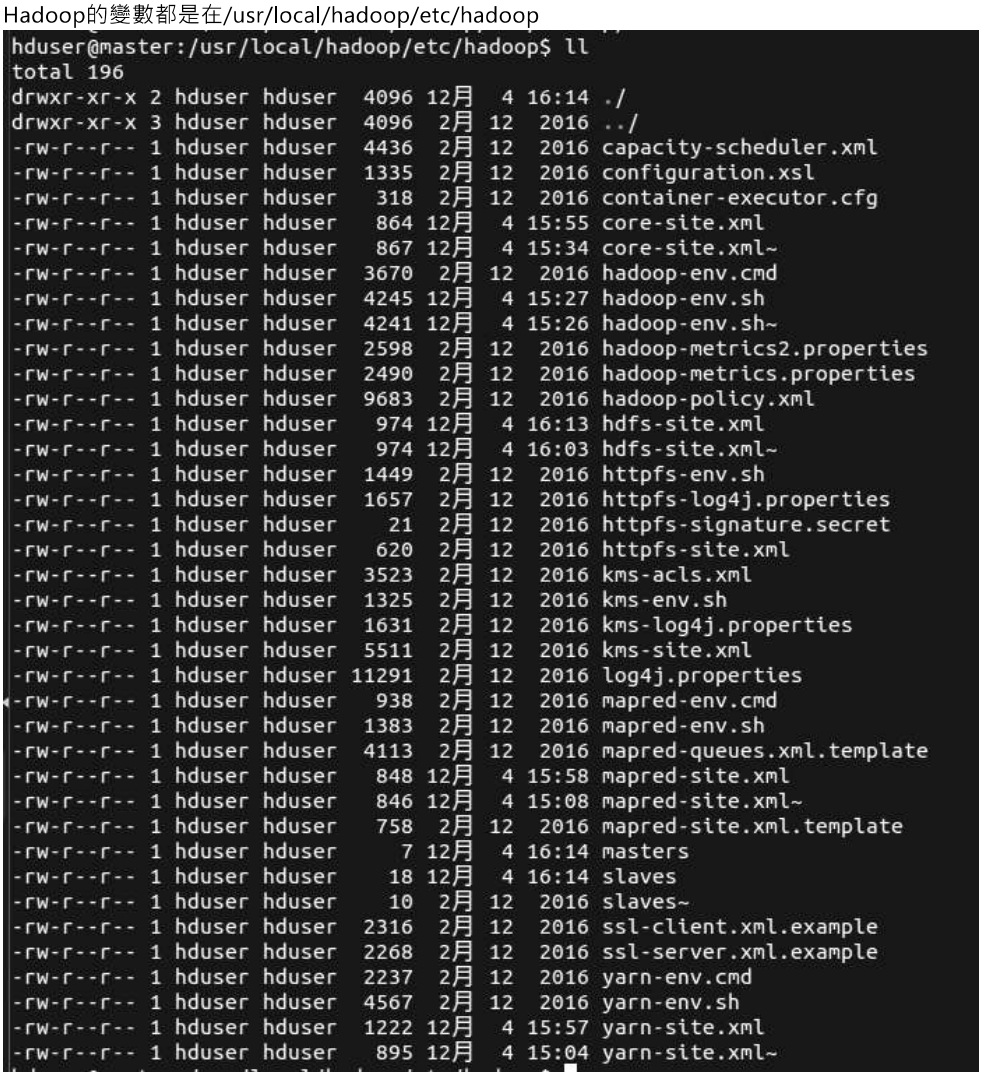
1. 建立hduser的虛擬機是作為template，請不要直接當作slave或master

- 記憶體(RAM)盡量4GB，避免不必要的卡頓，讓操作流暢
- 如果電腦硬體不夠好，請先不要管slave(data1~3)，作業的資料量沒有大到local(single node)跑不動
- 下載ubuntu 14.04的.iso(映像檔)套入VM中並安裝(不用裝中文也可)
- linux指令最好記一下
  - sudo 表示管理者權限執行
  - cd 表示前往目錄底下的某個子目錄
  - ll、ls可以查看當前目錄有哪些玩意兒
  - gedit、vim、vi可以修改檔案，建議都用gedit，想像成是文字編譯器
  - sudo reboot可以重開機
  - source ~/.bashrc可以使剛改完的環境變數立即生效
  - mkdir 是建立資料夾
  - jps可以查看hadoop節點有沒有開起來
  - start、stop-all.sh可以開關hadoop
- linux快捷鍵
  - ctrl+alt+t是叫出terminal(cmd)
  - 在terminal中要複製貼上，需要shift+ctrl+C/V
  - 要終止任何正在運行的程式(如jupyter)，使用ctrl+C

2.環境變數&Hadoop變數

- 電腦的環境變數都是在~/.bashrc，java、python的PATH都在此，對應windows的環境變數

- Hadoop的變數都是在/usr/local/hadoop/etc/hadoop



```
hduser@master:/usr/local/hadoop/etc/hadoop$ ll
total 196
drwxr-xr-x 2 hduser hduser  4096 12月  4 16:14 ./
drwxr-xr-x 3 hduser hduser  4096  2月 12  2016 ../
-rw-r--r-- 1 hduser hduser  4436  2月 12  2016 capacity-scheduler.xml
-rw-r--r-- 1 hduser hduser  1335  2月 12  2016 configuration.xsl
-rw-r--r-- 1 hduser hduser   318  2月 12  2016 container-executor.cfg
-rw-r--r-- 1 hduser hduser   864 12月  4 15:55 core-site.xml
-rw-r--r-- 1 hduser hduser   867 12月  4 15:34 core-site.xml~
-rw-r--r-- 1 hduser hduser  3670  2月 12  2016 hadoop-env.cmd
-rw-r--r-- 1 hduser hduser  4245 12月  4 15:27 hadoop-env.sh
-rw-r--r-- 1 hduser hduser  4241 12月  4 15:26 hadoop-env.sh~
-rw-r--r-- 1 hduser hduser  2598  2月 12  2016 hadoop-metrics2.properties
-rw-r--r-- 1 hduser hduser  2490  2月 12  2016 hadoop-metrics.properties
-rw-r--r-- 1 hduser hduser  9683  2月 12  2016 hadoop-policy.xml
-rw-r--r-- 1 hduser hduser   974 12月  4 16:13 hdfs-site.xml
-rw-r--r-- 1 hduser hduser   974 12月  4 16:03 hdfs-site.xml~
-rw-r--r-- 1 hduser hduser  1449  2月 12  2016 httpfs-env.sh
-rw-r--r-- 1 hduser hduser  1657  2月 12  2016 httpfs-log4j.properties
-rw-r--r-- 1 hduser hduser    21  2月 12  2016 httpfs-signature.secret
-rw-r--r-- 1 hduser hduser   620  2月 12  2016 httpfs-site.xml
-rw-r--r-- 1 hduser hduser  3523  2月 12  2016 kms-acls.xml
-rw-r--r-- 1 hduser hduser  1325  2月 12  2016 kms-env.sh
-rw-r--r-- 1 hduser hduser  1631  2月 12  2016 kms-log4j.properties
-rw-r--r-- 1 hduser hduser  5511  2月 12  2016 kms-site.xml
-rw-r--r-- 1 hduser hduser 11291  2月 12  2016 log4j.properties
-rw-r--r-- 1 hduser hduser   938  2月 12  2016 mapred-env.cmd
-rw-r--r-- 1 hduser hduser  1383  2月 12  2016 mapred-env.sh
-rw-r--r-- 1 hduser hduser  4113  2月 12  2016 mapred-queues.xml.template
-rw-r--r-- 1 hduser hduser   848 12月  4 15:58 mapred-site.xml
-rw-r--r-- 1 hduser hduser   846 12月  4 15:08 mapred-site.xml~
-rw-r--r-- 1 hduser hduser   758  2月 12  2016 mapred-site.xml.template
-rw-r--r-- 1 hduser hduser     7 12月  4 16:14 masters
-rw-r--r-- 1 hduser hduser    18 12月  4 16:14 slaves
-rw-r--r-- 1 hduser hduser    10  2月 12  2016 slaves~
-rw-r--r-- 1 hduser hduser  2316  2月 12  2016 ssl-client.xml.example
-rw-r--r-- 1 hduser hduser  2268  2月 12  2016 ssl-server.xml.example
-rw-r--r-- 1 hduser hduser  2237  2月 12  2016 yarn-env.cmd
-rw-r--r-- 1 hduser hduser  4567  2月 12  2016 yarn-env.sh
-rw-r--r-- 1 hduser hduser  1222 12月  4 15:57 yarn-site.xml
-rw-r--r-- 1 hduser hduser   895 12月  4 15:04 yarn-site.xml~
```

- 要改大部分都是xml檔，要記得configuration、property、name、value的結構一層一層不要少，也不要有無謂的空白，如: x = 1一定要打x=1

```
<configuration>
<property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
</property>
</configuration>
```

3.常用連結

- localhost:9000 是HDFS用的port
- localhost:8088 是檢查node有沒有開起來
- localhost:50070 一樣是檢查工作有沒有在跑(新一點的好像改port了)

> 要細分的話，一個是管工作排程、一個是管運算資源

- localhost:8888 是jupyter notebook

4. 其他

- 在ssh設定中，新版本的ubuntu可能不支援dsa(too old)，可以使用rsa作為替代

- spark裡開啟jupyter有分為local[*]與yarn兩種模式，詳細指令請看此
(http://pythonsparkhadoop.blogspot.com/2016/09/9-ipython-notebook-python-spark.html)

- ```
  In [1]: sc.master
  Out[1]: 'local[*]'
  ```
- ```
  In [4]: from pyspark import SparkContext
          sc =SparkContext.getOrCreate()
          sc.master
  Out[4]: u'yarn'

  In [ ]:
  ```

- 設定ip請務必設為static而非dhcp，你家的位置要固定，郵差才知道信往哪塞，會在virtual box設兩張網卡
  - NAT，讓你可以使用實體主機的網路連到外部
  - 內部網路，需static以方便master slave溝通
  - auto lo保留他，不同版本的ubuntu也都長不一樣
  - interfaces
    ```
    # interfaces(5) file used by ifup(8) and ifdown(8)
    auto lo
    iface lo inet loopback

    #NAT interface
    auto eth0
    iface eth0 inet dhcp

    auto eth1
    iface eth1 inet static
    address         192.168.56.100
    netmask         255.255.255.0
    network         192.168.56.0
    broadcast       192.168.56.255
    ```

5.小結

最後的最後，如果你想知道VM架起來長怎樣

VM (https://drive.google.com/drive/folders/1aP6jI1aHvN1cYuT5sU_xtNSip492Vihd?usp=sharing)(EC王廣和架的)

程式碼 (https://drive.google.com/drive/folders/14UvE6-Cmih5Ic7NMSNNiGzAcH84rjYfL?usp=sharing)(OTA黃瀚程撰寫)

想學習更多linux小知識請找NET楊大神

你下載了VM的iso檔卻開不起來，就要確認virtualbox的版本

程式碼

1. 將資料讀取進來 (可用pandas套件)
   cluster 模式執行(hadoop yarn)，從 HDFS 目錄讀取 csv 檔，讀取下來存成 Dataframe 型別。

```
In [1]: sc.master
```

```
Out[1]: u'yarn'
```

```
In [2]: global Path
        if sc.master[0:5]=="local" :
            Path="file:/home/hduser/Documents/"
        else:
            Path="hdfs://master:9000/user/hduser/"
```

```
In [3]: import pyspark
        from pyspark.sql import SparkSession
```

```
In [4]: sqlContext = SparkSession.builder.getOrCreate()
```

```
In [5]: data = sqlContext.read.format("csv").option("header", "true").load(Path+"movieRating.csv")
```

```
In [6]: data.select("TrainDataID", "UserID", "MovieID", "Rating").show(10)
```

```
+-----------+------+-------+------+
|TrainDataID|UserID|MovieID|Rating|
+-----------+------+-------+------+
|          1|   796|   1193|     5|
|          2|   796|    661|     3|
|          3|   796|    914|     3|
|          4|   796|   3408|     4|
|          5|   796|   2355|     5|
|          6|   796|   1197|     3|
|          7|   796|   1287|     5|
|          8|   796|   2804|     5|
|          9|   796|    919|     4|
|         10|   796|    595|     5|
+-----------+------+-------+------+
only showing top 10 rows
```

2. 亂數後拆成訓練集(80%)與測試集 (20%)
   將 Dataframe 轉為 MLlib 所需的 rdd Rating 結構，之後將資料拆分為訓練集(80%)與測試集(20%)。

```
In [8]: rates_data = data.rdd.map(lambda x: Rating (int(x[1]), int(x[2]), float(x[3])))
```

```
In [9]: # rates_data = rates_data.toDF().dropna().rdd
```

```
In [10]: (train, test) = rates_data.randomSplit([0.8, 0.2])
```

3. 建立矩陣分解模型
   將 Rating 訓練集丟到矩陣分解模型 `pyspark.mllib.recommendation.ALS` ，rank設為20，其餘設定使用預設值。

```
In [11]: model = ALS.train(train, 20, nonnegative=False)
```

4. 產出預測結果
   將測試集拆分為 (user, product) pair 及 rating 結構，代入 `predictAll()` 方法產生預測結果。

```
In [12]: test_x = test.map(lambda x: (x[0], x[1]))
         test_y = test.map(lambda x: x[2])
```

```
In [13]: result = model.predictAll(test_x)
```

5. MAE

將測試集實際值與預測結果 join ，代入 `pyspark.mllib.evaluation.RegressionMetrics()` ，可直接得到 MAE 值。

```
In [14]: result = result.map(lambda r: ((r.user, r.product), r.rating))
         ratesAndPreds = test.map(lambda r: ((r.user, r.product), r.rating)).join(result)
         predictAndTrue = ratesAndPreds.map(lambda r: r[1])
```

```
In [15]: regressionMetrics = RegressionMetrics(predictAndTrue)
```

```
In [16]: print(regressionMetrics.meanAbsoluteError)
```

```
0.73259759353
```