

學號：0653440

i. 截圖與步驟描述(安裝的部分將較重要的幾個步驟截圖即可)

安裝 HADOOP

設定 HADOOP 環境變數

```
$sudo gedit ~/.bashrc
```

```
# Hadoop Variable
```

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
```

```
export HADOOP_HOME=/usr/local/hadoop
```

```
export PATH=$PATH:$HADOOP_HOME/bin
```

```
export PATH=$PATH:$HADOOP_HOME/sbin
```

```
export HADOOP_MAPRED_HOME=$HADOOP_HOME
```

```
export HADOOP_COMMON_HOME=$HADOOP_HOME
```

```
export HADOOP_HDFS_HOME=$HADOOP_HOME
```

```
export YARN_HOME=$HADOOP_HOME
```

```
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
```

```
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
```

```
export JAVA_LIBRARY_PATH=$HADOOP_HOME/lib/native:$JAVA_LIBRARY_PATH
```

```
# Hadoop Variable
```

生效

```
$source ~/.bashrc
```

安裝 SCALA

設定 HADOOP 環境變數

```
export SCALA_HOME=/usr/local/scala
```

```
export PATH=$PATH:$SCALA_HOME/bin
```

安裝 Spark 2.0.2

設定 HADOOP 環境變數

```
export SPARK_HOME=/usr/local/spark
```

```
export PATH=$PATH:$SPARK_HOME/bin
```

\$pyspark

```

hduser@hduser-VirtualBox: ~
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel).
17/11/28 16:10:10 WARN NativeCodeLoader: Unable to load native-hadoop library for
your platform... using builtin-java classes where applicable
17/11/28 16:10:11 WARN Utils: Your hostname, hduser-VirtualBox resolves to a local
back address: 127.0.1.1; using 10.0.2.15 instead (on interface eth0)
17/11/28 16:10:11 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
17/11/28 16:10:12 WARN Utils: Service 'SparkUI' could not bind on port 4040. Atte
mpting port 4041.
Welcome to

      _/ _ \| | | | _/_/
     / ___ \| |_| |' __ \|
    / ___ \| __| | | | | |
   /___ \| |_| | | | | | |
  /___ \| \___|_|_|_| \___|
version 2.0.2

Using Python version 2.7.13 (default, Dec 20 2016 23:09:15)
SparkSession available as 'spark'.

In [1]:

```

ii. 作業目標對應之結果

1. 前處理

```

import pandas as pd
#from sklearn import tree
import graphviz
#from sklearn.model_selection import train_test_split
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.tree import DecisionTree
from pyspark.mllib.util import MLUtils
from pyspark.mllib.evaluation import MulticlassMetrics

df = pd.read_csv("file:/home/hduser/pythonwork/dmhw1208/data/character-deaths.csv")#讀取資料
df=df.fillna(0)#把空值以0替代
df.loc[df['Death Year'] > 0, 'Death Year'] = 1.0
df=df.drop('Book of Death',axis = 1)
df=df.drop('Death Chapter',axis = 1)
df = pd.get_dummies(df, columns=['Allegiances'])#將Allegiances底下的家族轉成dummy的特徵
df = df.drop('Name',axis = 1)

```

2. 格式轉換

```
x = df.drop('Death Year', axis=1)
y = df['Death Year']

from sklearn.datasets import dump_svmlight_file
dump_svmlight_file(x, y, 'svm-output.libsvm') # where is your y?
from sklearn.datasets import load_svmlight_file
```

3. 製作 model

```
from pyspark.mllib.tree import DecisionTree, DecisionTreeModel
from pyspark.mllib.util import MLUtils
from pyspark.mllib.evaluation import BinaryClassificationMetrics
from pyspark.mllib.evaluation import MulticlassMetrics
from pyspark.mllib.evaluation import MultilabelMetrics

data = MLUtils.loadLibSVMFile(sc,"svm-output.libsvm")
(trainingData, testData) = data.randomSplit([0.75, 0.25])
model = DecisionTree.trainClassifier(trainingData, numClasses=2, categoricalFeaturesInfo={}, impurity='gini', maxDepth=5, maxBins=32)
```

4. 預測

```
predictions = model.predict(testData.map(lambda x: x.features))
labelsAndPredictions = testData.map(lambda lp: lp.label).zip(predictions)
```

5. 計算 accuracy, recall, precision

```
metrics = MulticlassMetrics(labelsAndPredictions)
precision = metrics.precision(label=1)
recall = metrics.recall(label=1)
Accuracy = metrics.accuracy
print("Precision = %s" % precision)
print("Recall = %s" % recall)
print("Accuracy = %s" % Accuracy)
```

6. 結果

(1) 產出預測結果(僅列出前三十項，第一欄為原始資料，第二欄為預測的結果。)

(1.0, 0.0),	(0.0, 0.0),	(0.0, 0.0),
(0.0, 1.0),	(0.0, 1.0),	(0.0, 0.0),
(1.0, 1.0),	(1.0, 1.0),	(0.0, 0.0),
(0.0, 1.0),	(0.0, 0.0),	(1.0, 0.0),
(0.0, 1.0),	(0.0, 1.0),	(0.0, 0.0),
(1.0, 1.0),	(0.0, 1.0),	(0.0, 0.0),
(1.0, 1.0),	(0.0, 0.0),	(0.0, 1.0),
(0.0, 1.0),	(0.0, 1.0),	(0.0, 0.0),
(0.0, 0.0),	(0.0, 0.0),	(0.0, 0.0),
(0.0, 1.0),	(0.0, 0.0),	(0.0, 1.0)]

(2) 計算 accuracy, recall, precision

Precision = 0.626666666667

Recall = 0.61038961039

Accuracy = 0.725118483412

iii. 討論

- 環境設定：安裝版本若與參考資料不同，則檔案路徑也會不同，在下指令時要特別更改。
- 程式：pandas 前處理後的資料格式為 dataframe，須轉為 mllib 用的 libsvm 格式才能做後續的建樹及預測。