LOREM IPSUM

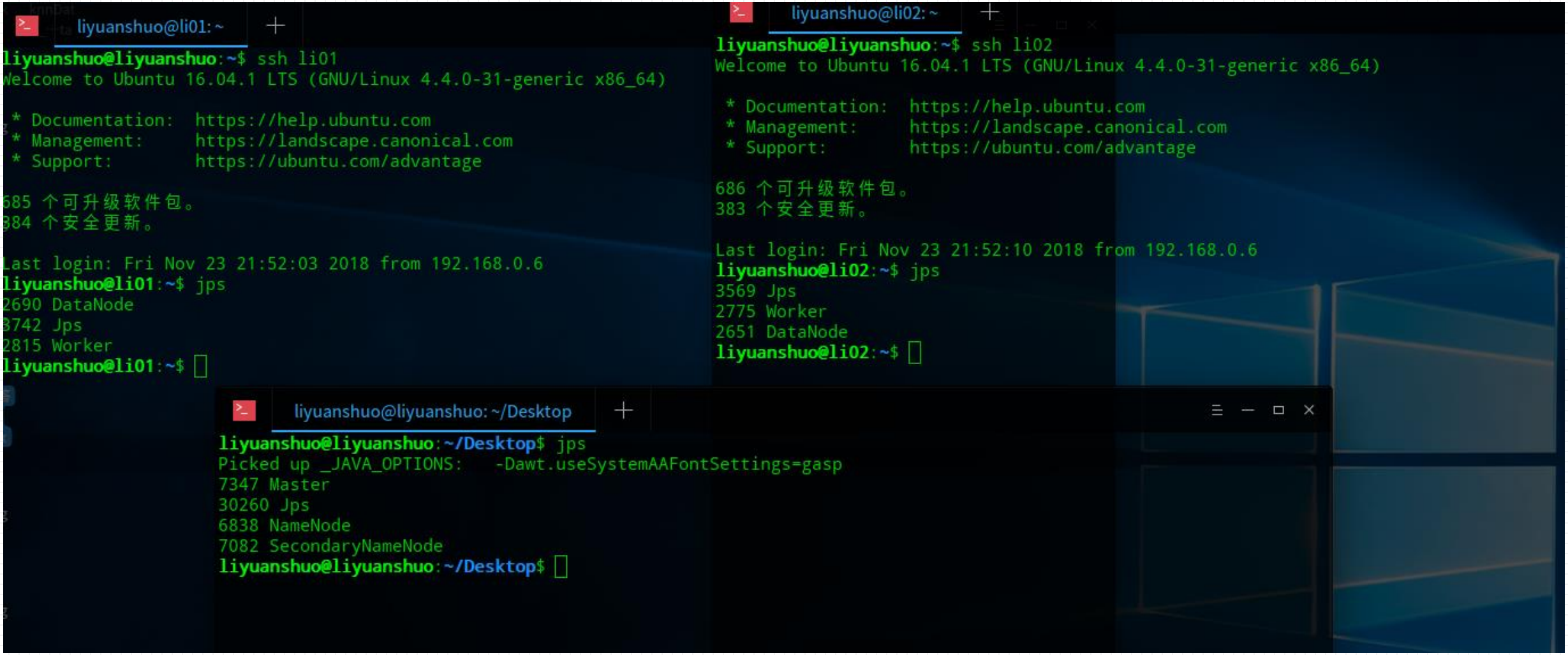
大数据·Hadoop·Spark

傅友、曾友雯、李傲

CONTENTS

01 大数据平台部署

1 完全分布式HADOOP & SPARK部署



1 完全分布式HADOOP & *SPARK部署*

```
liyuanshuo@liyuanshuo:~/Desktop$ jps
Picked up _JAVA_OPTIONS:   -Dawt.useSystemAAFontSettings=gasp
7347 Master
30260 Jps
6838 NameNode
7082 SecondaryNameNode
liyuanshuo@liyuanshuo:~/Desktop$ pyspark
Picked up _JAVA_OPTIONS:   -Dawt.useSystemAAFontSettings=gasp
Python 3.6.5 (default, May 11 2018, 13:30:17)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Picked up _JAVA_OPTIONS:   -Dawt.useSystemAAFontSettings=gasp
Picked up _JAVA_OPTIONS:   -Dawt.useSystemAAFontSettings=gasp
2018-11-24 14:32:45 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... us
ing builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.3.0
      /_/

Using Python version 3.6.5 (default, May 11 2018 13:30:17)
SparkSession available as 'spark'.
>>>
```

1 完全分布式HADOOP & SPARK部署

Configured Capacity:	20.42 GB
DFS Used:	2.45 GB (12.01%)
Non DFS Used:	11.2 GB
DFS Remaining:	5.68 GB (27.82%)
Block Pool Used:	2.45 GB (12.01%)
DataNodes usages% (Min/Median/Max/stdDev):	0.68% / 15.82% / 15.82% / 2.07%

Spark Master at spark://liyuanshuo:7077

2.3.0

URL: spark://liyuanshuo:7077
REST URL: spark://liyuanshuo:6066 *(cluster mode)*
Alive Workers: 2
Cores in use: 4 Total, 0 Used
Memory in use: 4.0 GB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

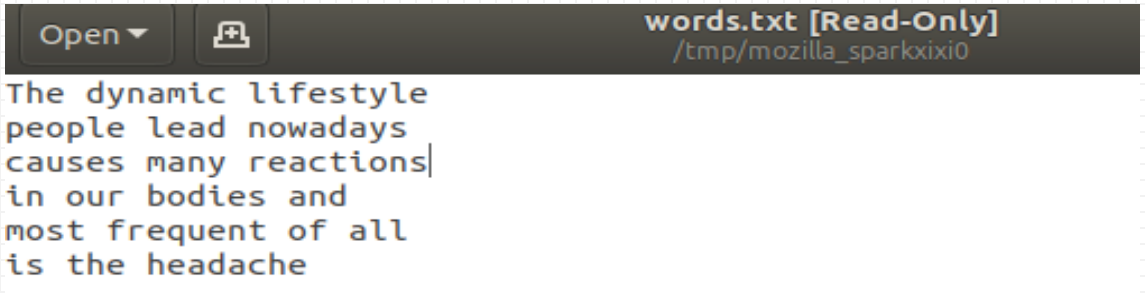
Workers (2)

Worker Id	Address	State	Cores	Memory
worker-20181123215218-192.168.0.7-33019	192.168.0.7:33019	ALIVE	2 (0 Used)	2.0 GB (0.0 B Used)
worker-20181123215218-192.168.0.9-40711	192.168.0.9:40711	ALIVE	2 (0 Used)	2.0 GB (0.0 B Used)

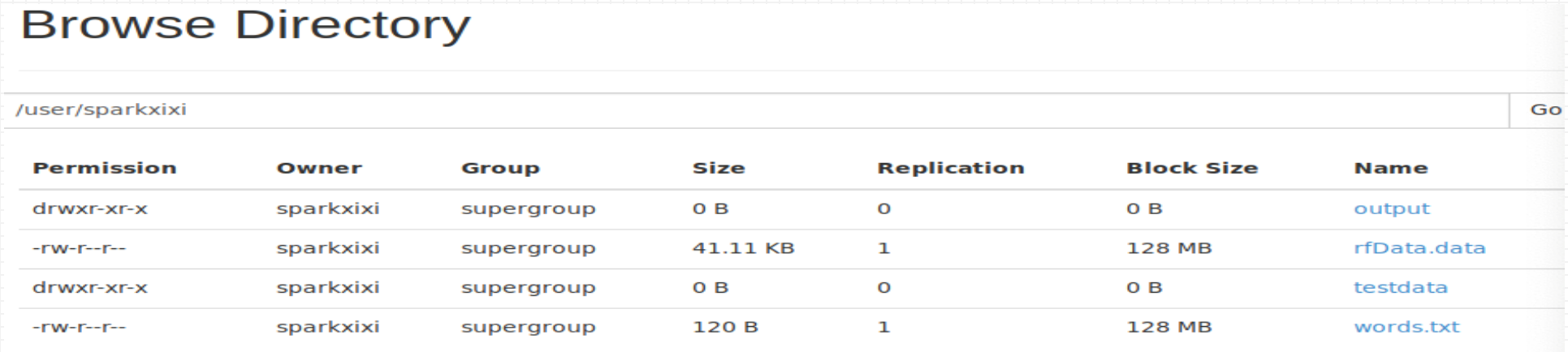
02 Spark & Hadoop案列 分析与实现

1 Wordcount小样例

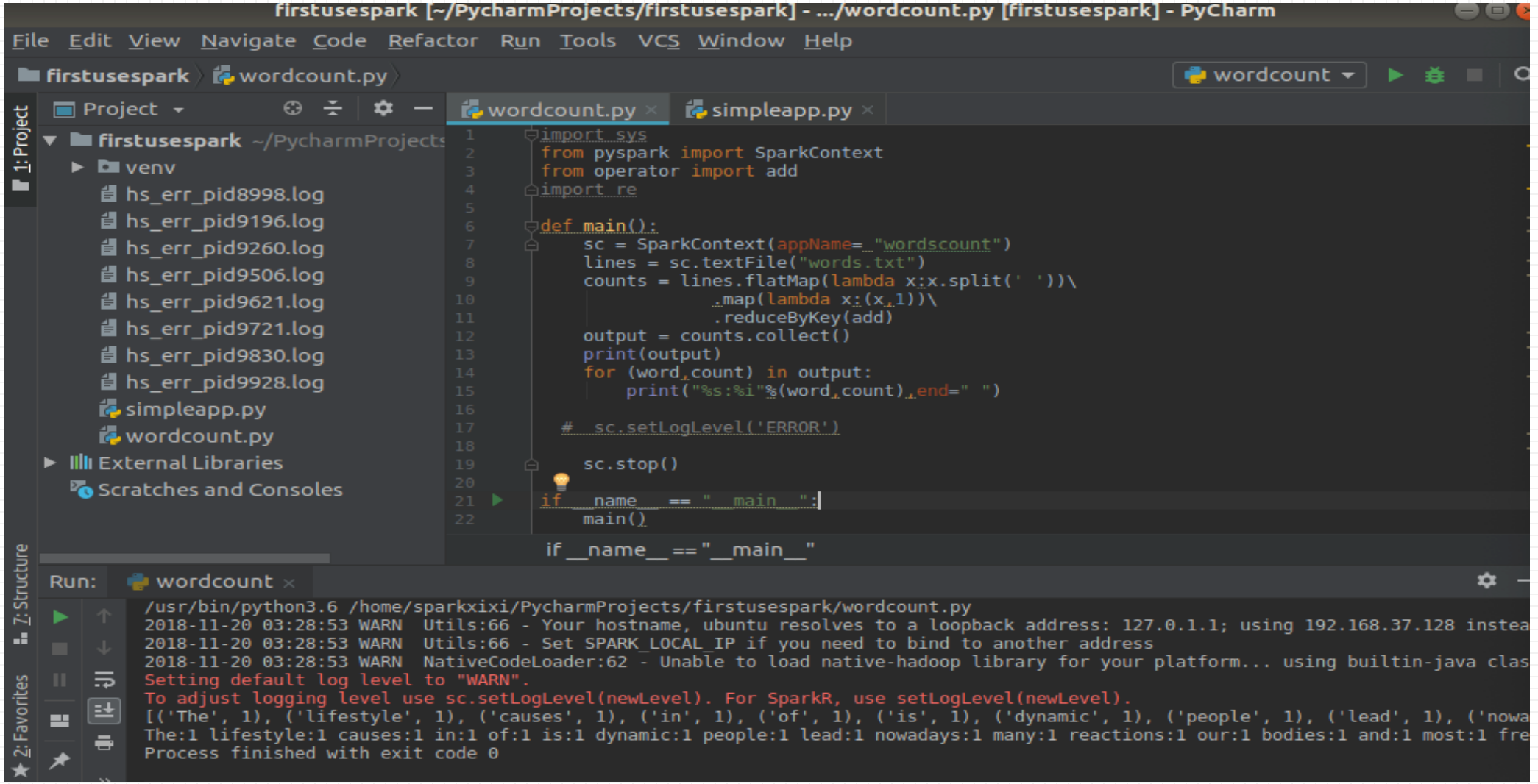
数据集：



hadoop fs -put words.txt hdfs://localhost:9000/user/sparkxixi



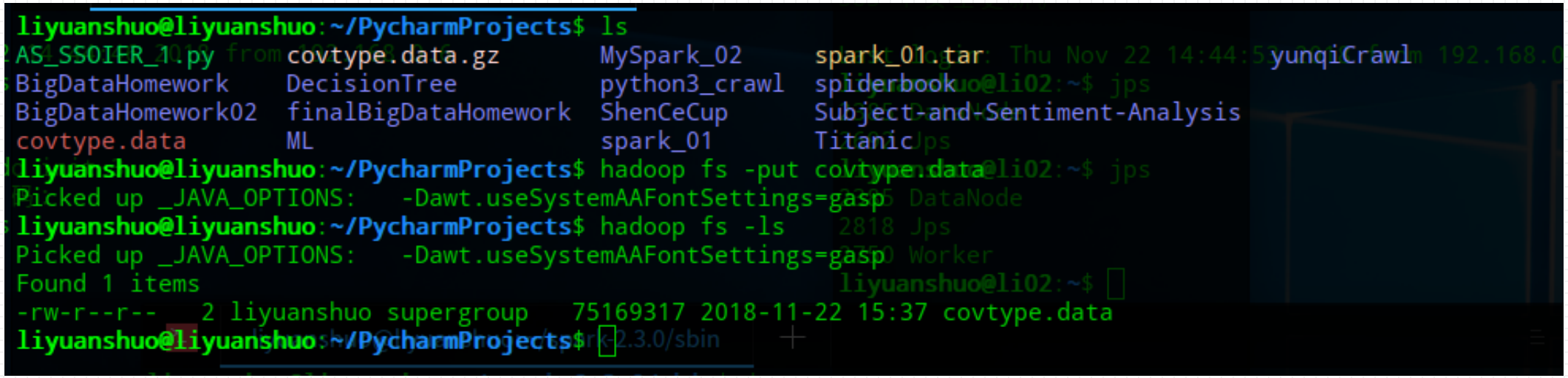
1 Wordcount小样例



2 森林植被分类预测

数据集：使用UCI的数据集 解压后大小为71M（官方数据描述很清楚）
下载地址：**https://archive.ics.uci.edu/ml/machine-learning-databases/covtype/**
数据集记录了美国科罗拉多州不同地块森林植被特征：海拔、坡度、到水源的距离、遮阳情况和土壤类型，并且随同给出了地块的已知森林植被类型。有581012个样本。
字段描述：54个属性字段，一个类别字段环境属性10个：数值属性，描述海拔、方位、斜角等环境信息；野生区域属性4个：二元属性，标定区域类型；土壤类型属性40个：二元属性，标记土壤类型；森林植被类型：1~7，标记七种不同类型

数据集部署到hadoop+spark平台上：



2 森林植被分类预测

查看数据集：

从HDFS文件系统中读取查看数据集的大致信息

```
/usr/bin/python3.6 /home/liyuanshuo/PycharmProjects/finalBigDataHomework/fileInfo.py
Picked up _JAVA_OPTIONS:   -Dawt.useSystemAAFontSettings=gasp
Picked up _JAVA_OPTIONS:   -Dawt.useSystemAAFontSettings=gasp
2018-11-22 15:29:19 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
*********************************************查看数据集的前两个*********************************************
2018-11-22 15:29:24 WARN  Utils:66 - Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.debug.maxToStringFields'
  in SparkEnv.conf.
[Row(_c0='2596', _c1='51', _c2='3', _c3='258', _c4='0', _c5='510', _c6='221', _c7='232', _c8='148', _c9='6279', _c10='1', _c11='0', _c12='0', _c13='0', _c14='0', _c15='0', _c16='0',
  _c17='0', _c18='0', _c19='0', _c20='0', _c21='0', _c22='0', _c23='0', _c24='0', _c25='0', _c26='0', _c27='0', _c28='0', _c29='0', _c30='0', _c31='0', _c32='0', _c33='0', _c34='0',
  _c35='0', _c36='0', _c37='0', _c38='0', _c39='0', _c40='0', _c41='0', _c42='1', _c43='0', _c44='0', _c45='0', _c46='0', _c47='0', _c48='0', _c49='0', _c50='0', _c51='0', _c52='0',
  _c53='0', _c54='5'), Row(_c0='2590', _c1='56', _c2='2', _c3='212', _c4='-6', _c5='390', _c6='220', _c7='235', _c8='151', _c9='6225', _c10='1', _c11='0', _c12='0', _c13='0',
  _c14='0', _c15='0', _c16='0', _c17='0', _c18='0', _c19='0', _c20='0', _c21='0', _c22='0', _c23='0', _c24='0', _c25='0', _c26='0', _c27='0', _c28='0', _c29='0', _c30='0', _c31='0',
  _c32='0', _c33='0', _c34='0', _c35='0', _c36='0', _c37='0', _c38='0', _c39='0', _c40='0', _c41='0', _c42='1', _c43='0', _c44='0', _c45='0', _c46='0', _c47='0', _c48='0', _c49='0',
  _c50='0', _c51='0', _c52='0', _c53='0', _c54='5')]
*********************************************数据集的大小*********************************************
[Stage 2:=============================================>             (3 + 1) / 4]581012
*********************************************数据集的描述*********************************************
DataFrame[summary: string, _c0: string, _c1: string, _c2: string, _c3: string, _c4: string, _c5: string, _c6: string, _c7: string, _c8: string, _c9: string, _c10: string, _c11:
  string, _c12: string, _c13: string, _c14: string, _c15: string, _c16: string, _c17: string, _c18: string, _c19: string, _c20: string, _c21: string, _c22: string, _c23: string,
  _c24: string, _c25: string, _c26: string, _c27: string, _c28: string, _c29: string, _c30: string, _c31: string, _c32: string, _c33: string, _c34: string, _c35: string, _c36:
  string, _c37: string, _c38: string, _c39: string, _c40: string, _c41: string, _c42: string, _c43: string, _c44: string, _c45: string, _c46: string, _c47: string, _c48: string,
  _c49: string, _c50: string, _c51: string, _c52: string, _c53: string, _c54: string]
************************************************************************************************************

Process finished with exit code 0
```

2 森林植被分类预测

数据格式转换数据预处理：

Pyspark要求数据格式为LabeledPoint的RDD形式的，原始数据格式不符合，进行一下格式转换
在调用Pyspark提供的机器学习库的时候，需要对数据进行预处理，把数据范围进行相应的调整
每个算法都需要进行一次，这里只展示的是贝叶斯算法的数据预处理的输出

```
/usr/bin/python3.6 /home/liyuanshuo/PycharmProjects/finalBigDataHomework/dataTransform.py
Picked up _JAVA_OPTIONS:   -Dawt.useSystemAAFontSettings=gasp
Picked up _JAVA_OPTIONS:   -Dawt.useSystemAAFontSettings=gasp
2018-11-22 16:56:04 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
*******************************格式转换和数据预处理之后的数据*******************************
(581012, 55)
[[5.         1.1055     0.42382271 ... 0.         0.         0.        ]
 [5.         1.0965     0.46537396 ... 0.         0.         0.        ]
 [2.         1.4175     1.15512465 ... 0.         0.         0.        ]
 ...
 [3.         0.7905     1.32132964 ... 0.         0.         0.        ]
 [3.         0.7875     1.41274238 ... 0.         0.         0.        ]
 [3.         0.786      1.37119114 ... 0.         0.         0.        ]]
*********************************************************************
Picked up _JAVA_OPTIONS:   -Dawt.useSystemAAFontSettings=gasp

Process finished with exit code 0
```

可以参考网上的数据预处理，这里不存在缺失值，直接对数据范围进行转化

2 森林植被分类预测

数据格式转换数据预处理：

/liyuanshuo Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	liyuanshuo	supergroup	3.72 KB	2018/11/14 下午7:22:23	2	16 MB	README.md
-rw-r--r--	liyuanshuo	supergroup	71.69 MB	2018/11/22 下午4:40:00	2	16 MB	covtype.data
drwxr-xr-x	liyuanshuo	supergroup	0 B	2018/11/14 下午8:19:36	0	0 B	csvs_per_year
-rw-r--r--	liyuanshuo	supergroup	60.95 MB	2018/11/22 下午4:57:10	2	16 MB	data.data
-rw-r--r--	liyuanshuo	supergroup	156.46 MB	2018/11/22 下午5:59:39	2	16 MB	decisionTreeData.data
-rw-r--r--	liyuanshuo	supergroup	207.64 MB	2018/11/23 下午8:09:49	2	16 MB	knnData.data
-rw-r--r--	liyuanshuo	supergroup	3.54 MB	2018/11/23 下午10:06:47	2	16 MB	knnData_10000.data
-rw-r--r--	liyuanshuo	supergroup	1.77 MB	2018/11/23 下午10:13:46	2	16 MB	knnData_5000.data
-rw-r--r--	liyuanshuo	supergroup	17.71 MB	2018/11/23 下午9:52:29	2	16 MB	knnData_50000.data
-rw-r--r--	liyuanshuo	supergroup	3.72 KB	2018/11/14 下午7:16:51	2	16 MB	mySpark
-rw-r--r--	liyuanshuo	supergroup	156.46 MB	2018/11/22 下午9:47:46	2	16 MB	randomForestData.data

2 森林植被分类预测

问题分析：

需要预测的结果：森林植被类型：1~7，标记七种不同类型

得出：这是一个分类问题

选择算法：

1、朴素贝叶斯分类：http://spark.apache.org/docs/latest/mllib-naive-bayes.html

2、决策树：http://spark.apache.org/docs/latest/mllib-decision-tree.html

3、决策树升级->随机森林：

http://spark.apache.org/docs/latest/mllib-ensembles.html#random-forests

4、K近邻-官方没有现成的库，但是算法原理比较简单，且有其他版本的实现可供参考

考虑自己实现KNN算法

https://github.com/evancasey/spark-knn-recommender

https://github.com/IdearHui/sparkML_project

前三种算法参考样例实现，第四种算法参考民间代码复现

2 森林植被分类预测

朴素贝叶斯:
数据预处理代码:

```
data[:, i] = (data[:, i]-data[:, i].min())/(data[:, i].max()-data[:, i].min()+1)*3
```

训练集设置60%， 测试集设置40%
分类预测代码:

```
# 获得训练模型,第一个参数为数据，第二个参数为平滑参数lamda，默认为1，可改
model = NaiveBayes.train(training, lamda: 1.0)
```

```
2018-11-22 17:42:15 INFO  TaskSetManager:54 - Finished task 0.0 in stage 5.0 (TID 14) in 2853 ms on localhost (executor driver) (4/4)
2018-11-22 17:42:15 INFO  TaskSchedulerImpl:54 - Removed TaskSet 5.0, whose tasks have all completed, from pool
2018-11-22 17:42:15 INFO  DAGScheduler:54 - ResultStage 5 (count at /home/liyuanshuo/PycharmProjects/finalBigDataHomework/navieBayes.py:25) finished in 2.863 s
2018-11-22 17:42:15 INFO  DAGScheduler:54 - Job 4 finished: count at /home/liyuanshuo/PycharmProjects/finalBigDataHomework/navieBayes.py:25, took 2.866287 s
*****************************打印朴素贝叶斯的准确率：（0.6 / 0.4）*****************************
0.640302966293491
************************************************************
2018-11-22 17:42:15 INFO  SparkContext:54 - Invoking stop() from shutdown hook
2018-11-22 17:42:15 INFO  AbstractConnector:318 - Stopped Spark@68c3d986{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
2018-11-22 17:42:15 INFO  SparkUI:54 - Stopped Spark web UI at http://liyuanshuo:4040
2018-11-22 17:42:15 INFO  MapOutputTrackerMasterEndpoint:54 - MapOutputTrackerMasterEndpoint stopped!
```

2 森林植被分类预测

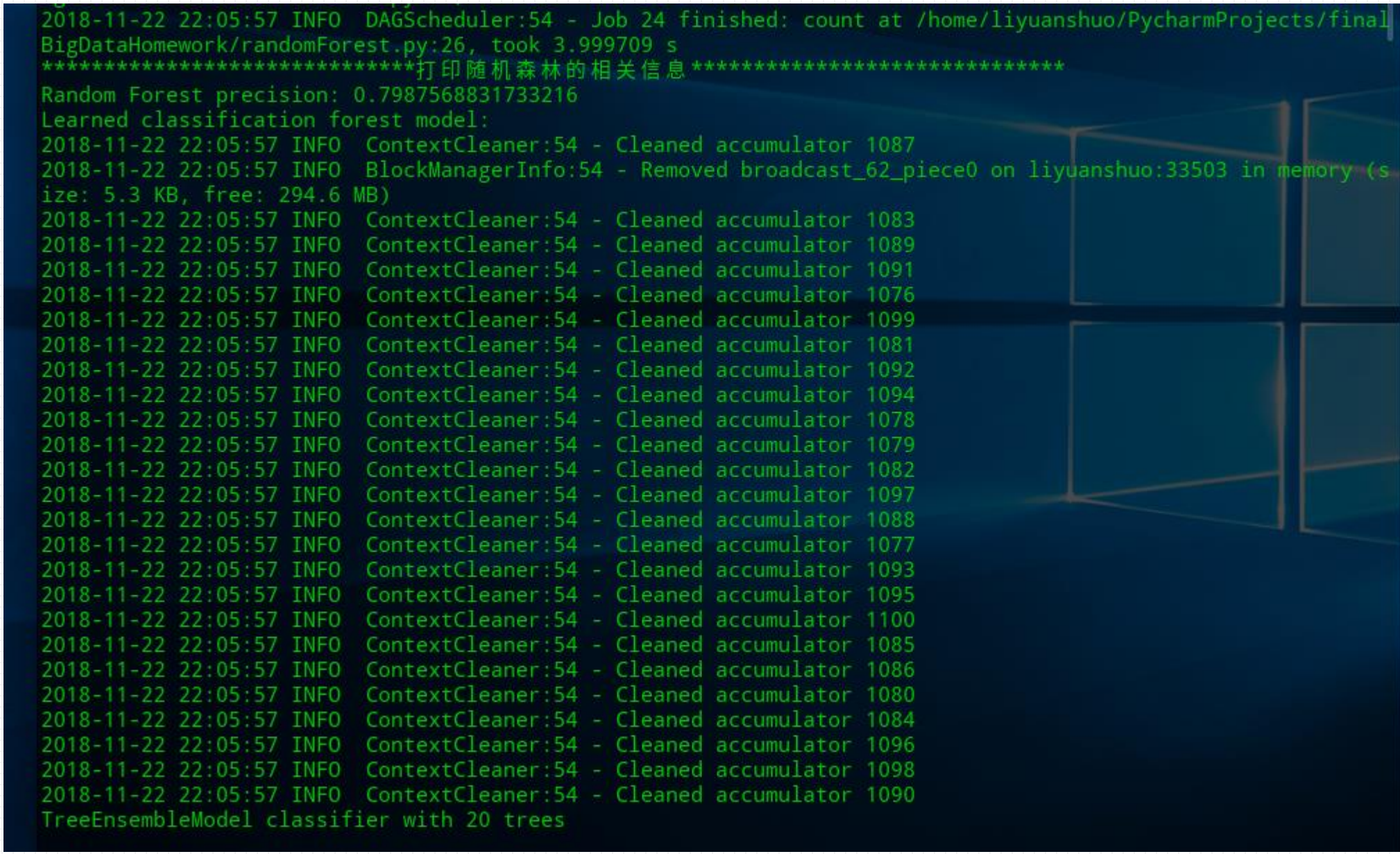
决策树：
70%的训练集，30%的测试集

```
# 训练模型
# 空categoricalFeaturesInfo表示所有特征是连续的。代码直接参照官方的样例就可以，只不过部分参数需要自己设置一下
#  Empty categoricalFeaturesInfo indicates all features are continuous.
#  参数说明
#  numClasses:分类数，需比实际类别数量大，这里设置为8；
#  categoricalFeaturesInfo:特征类别信息，为空，意为所有特征为连续型变量；
#  impurity:信息纯度度量，进行分类时可选择熵或基尼，这里设置为基尼；
#  maxDepth:决策树最大深度，这里设为15；
#  maxBins:特征分裂时的最大划分数量,这里设为32。
model = DecisionTree.trainClassifier(trainingData, numClasses=8, categoricalFeaturesInfo={},
                                     impurity='gini', maxDepth=15, maxBins=32)
```

```
2018-11-22 21:39:42 INFO  PythonRunner:54 - Times: total = 1243, boot = -3542, init = 3543, finish = 1242
2018-11-22 21:39:42 INFO  Executor:54 - Finished task 8.0 in stage 37.0 (TID 360). 1354 bytes result sent to driver
2018-11-22 21:39:42 INFO  TaskSetManager:54 - Finished task 8.0 in stage 37.0 (TID 360) in 1252 ms on localhost (executor driver) (10/10)
2018-11-22 21:39:42 INFO  TaskSchedulerImpl:54 - Removed TaskSet 37.0, whose tasks have all completed, from pool
2018-11-22 21:39:42 INFO  DAGScheduler:54 - ResultStage 37 (count at /home/liyuanshuo/PycharmProjects/finalBigDataHomework/decisionTree.py:26) finished in 3.872 s
2018-11-22 21:39:42 INFO  DAGScheduler:54 - Job 21 finished: count at /home/liyuanshuo/PycharmProjects/finalBigDataHomework/decisionTree.py:26, took 3.875128 s
*****************************打印决策树的准确率：（0.7 / 0.3）*****************************
Decision Tree precision: 0.8535617476798677
************************************************************
2018-11-22 21:39:42 INFO  SparkContext:54 - Invoking stop() from shutdown hook
2018-11-22 21:39:42 INFO  AbstractConnector:318 - Stopped Spark@eabb2c6{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
2018-11-22 21:39:42 INFO  SparkUI:54 - Stopped Spark web UI at http://liyuanshuo:4040
2018-11-22 21:39:42 INFO  MapOutputTrackerMasterEndpoint:54 - MapOutputTrackerMasterEndpoint stopped!
2018-11-22 21:39:42 INFO  MemoryStore:54 - MemoryStore cleared
```

2 森林植被分类预测

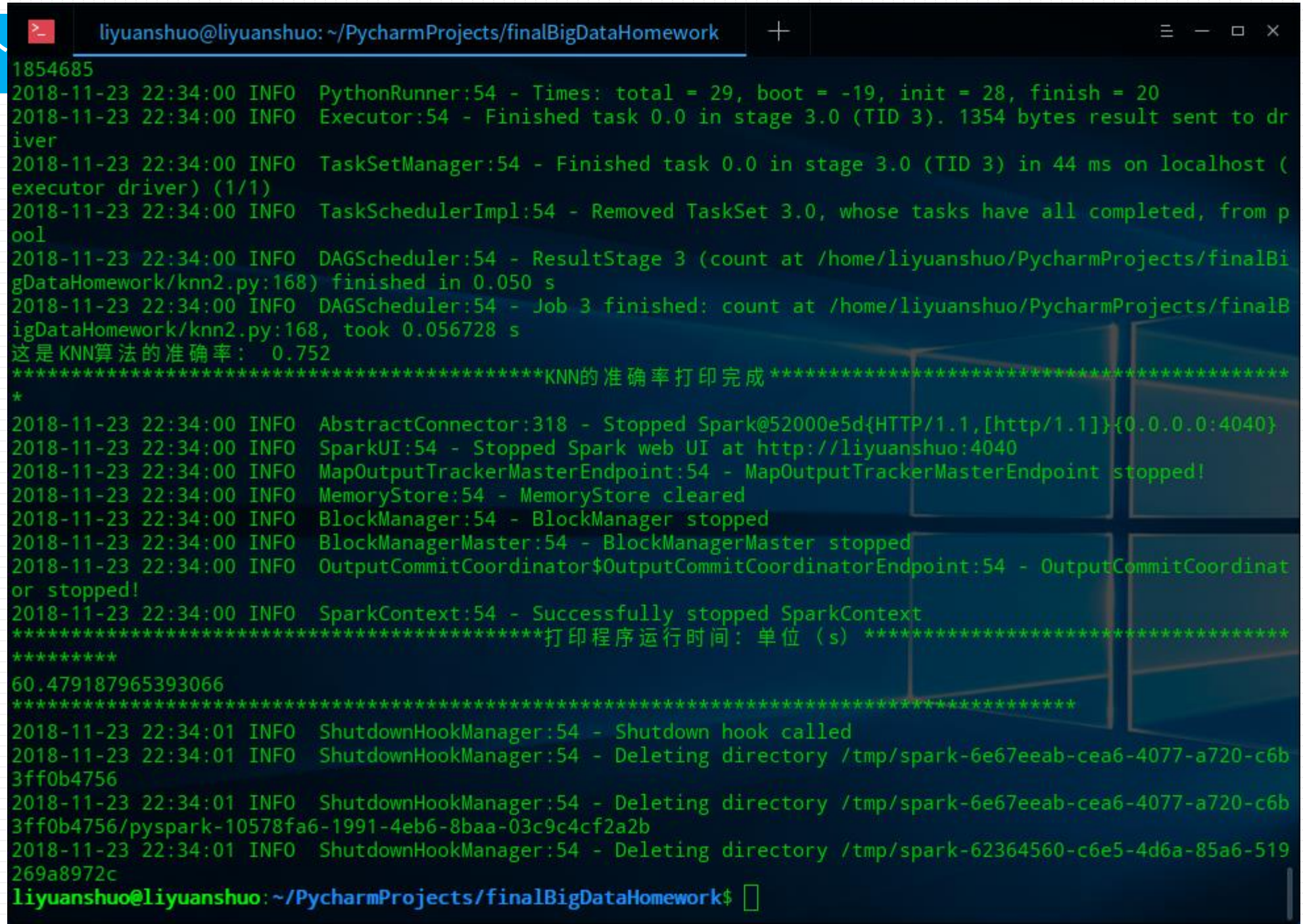
随机森林：
参数说明：
numClasses:分类数，需比实际类别数量大，这里设置为8；
categoricalFeaturesInfo:特征类别信息，为空，意为所有特征为连续型变量；
numTrees:森林中树的数量，这里设为20；
featureSubsetStrategy:特征子集采样策略，auto表示算法自主选取；
impurity:信息纯度度量，进行分类时可选择熵或基尼，这里设置为基尼；
maxDepth:决策树最大深度，这里设为18；
maxBins:特征分裂时的最大划分数量,这里设为32。



2 森林植被分

K近邻：
7/3的数据比例

参数说明：
k为6，距离权重均为1，样本间距离使用欧几里得距离



2 森林植被分类预测

Talk is cheap show me the code

数据预处理代码：
https://github.com/liyuanshuo/finalBigDataHomework/blob/master/dataTransform.py
朴素贝叶斯代码：
https://github.com/liyuanshuo/finalBigDataHomework/blob/master/navieBayes.py
决策树代码：
https://github.com/liyuanshuo/finalBigDataHomework/blob/master/decisionTree.py
随机森林代码：
https://github.com/liyuanshuo/finalBigDataHomework/blob/master/randomForest.py
KNN代码：
https://github.com/liyuanshuo/finalBigDataHomework/blob/master/knn.py
https://github.com/liyuanshuo/finalBigDataHomework/blob/master/knn2.py
整个项目代码：
https://github.com/**liyuanshuo/finalBigDataHomework**

3 利用Mahout进行K-means聚类分析

Mahout简介

Mahout 是一个很强大的**数据挖掘工具**，是一个分布式机器学习算法的集合，包括：被称为Taste的分布式协同过滤的实现、分类、聚类等。Mahout最大的优点就是基于**hadoop实现**，把很多以前运行于单机上的算法，转化为了**MapReduce模式**，这样大大提升了算法可处理的数据量和处理性能。

3 利用Mahout进行K-means聚类分析

Mahout安装配置：

1. 下载 Mahout 发布版
2. 解压安装 Mahout 将压缩文件解压到 /home/hadoop/
3. 配置环境变量

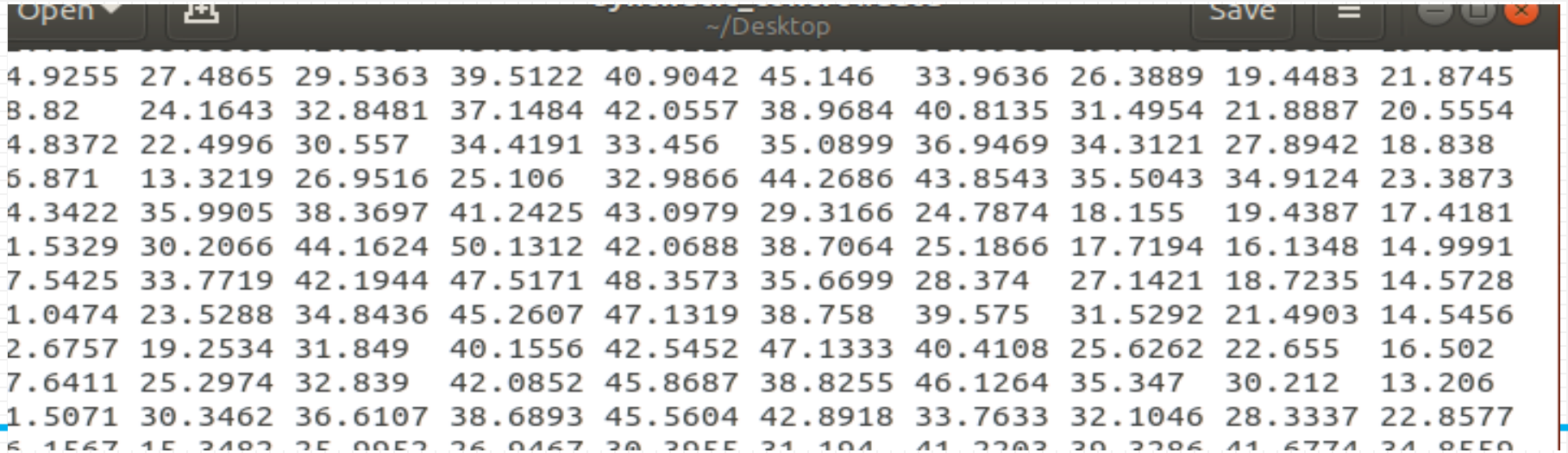

```
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$SPARK_HOME/bin:$PATH:$SCALA_HOME/bin:$PATH:$MAHOUT_HOME/conf:$MAHOUT_HOME/bin:$PATH
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib:$HADOOP_COMMON_LIB_NATIVE_DIR"

export SPARK_HOME=/usr/local/spark
export PYTHONPATH=$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.7-src.zip:$PYTHONPATH
export PYSPARK_PYTHON=/usr/bin/python3.6
export PYSPARK_DRIVER_PYTHON=/usr/bin/python3.6
export SCALA_HOME=/usr/share/scala-2.11

export MAHOUT_HOME=/usr/local/hadoop/apache-mahout-distribution-0.11.0
export MAHOUT_CONF_DIR=$MAHOUT_HOME/conf
```

3 利用Mahout进行K-means聚类分析

数据集：

http://archive.ics.uci.edu/ml/databases/synthetic_control/synthetic_control.data

数据包括600个数据点，每个数据点有60个属性。
数据格式
数据存储在ASCII文件中，600行，60列，每行一个图表。 类别信息如下：
1-100正常；101-200循环；201-300增长趋势；301-400减少趋势；401-500向上移动；501-600向下移动



3 利用Mahout进行K-means聚类分析

1. 上传数据到HDFS

hadoop fs -put synthetic_control.data hdfs://localhost:9000/user/sparkxixi/testdata

2. 运行 Mahout 中的 kmeans 聚类算法，执行命令

hadoop jar mahout-examples-0.11.0-job.jar org.apache.mahout.clustering.syntheticcontrol.kmeans.Job

3 利用Mahout进行K-means聚类分析

```
18/11/20 04:03:06 INFO mapreduce.Job:  map 0% reduce 0%
18/11/20 04:03:11 INFO mapreduce.Job:  map 100% reduce 0%
18/11/20 04:03:11 INFO mapreduce.Job: Job job_1542705685059_0012 completed successfully
18/11/20 04:03:11 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=107753
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=343371
                HDFS: Number of bytes written=363295
                HDFS: Number of read operations=13
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=2429
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=2429
                Total vcore-milliseconds taken by all map tasks=2429
                Total megabyte-milliseconds taken by all map tasks=2487296
        Map-Reduce Framework
                Map input records=600
                Map output records=600
                Input split bytes=126
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=47
                CPU time spent (ms)=1020
                Physical memory (bytes) snapshot=168841216
```

3 利用Mahout进行K-means聚类分析

聚类结果可在output cluster-10-final里查看：

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	sparkxixi	supergroup	194 B	1	128 MB	_policy
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	clusteredPoints
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	clusters-0
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	clusters-1
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	clusters-10-final
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	clusters-2
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	clusters-3
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	clusters-4
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	clusters-5
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	clusters-6
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	clusters-7
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	clusters-8
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	clusters-9
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	data
drwxr-xr-x	sparkxixi	supergroup	0 B	0	0 B	random-seeds

03 踩过的坑

1. 关于在pycharm中spark连接python的错误
   如图所示：

```
rotocol.Py4JJavaError: An error occurred while calling z:org.apache.spark.api.python.PythonRDD.collectAndServe.
apache.spark.SparkException: Job aborted due to stage failure: Task 0 in stage 0.0 failed 1 times, most recent failu
 java.base/java.lang.ProcessBuilder.start(ProcessBuilder.java:1128)
```

   解决办法：在Run/Edit Configurations/Environment variables中添加环境变量
   如图所示：

Name	Value
PYTHONUNBUFFERED	1
SPARK_HOME	/usr/local/spark
PYTHONPATH	/usr/local/spark/python
PYSPARK_PYTHON	/usr/bin/python3.6

2. KNN算法运行时内存爆满：
   不要使用那么多数据，Spark是基于内存的

3. 每次数据处理完之后一定要打开看一看，确保预处理之后的数据时正确的
   第一次处理结果对，第二次数据预处理结果未必对

4. Pycharm设置环境变量后可以本地连接spark进行简单的调试，
   这样方便查找代码错误出在哪里

04 反思和总结 收获

收获：

1、了解了Hadoop生态环境以及应用场景
2、小组讨论可以活跃思维，可以向组内其他成员学习不同的看法以及理解
3、PySpark的编程案例，以及ML库的接口调用
4、基础机器学习算法的原理，以及算法准确度的度量
5、了解了Mahout等第三方数据挖掘工具
6、增强了动手能力，以及写Bug的能力

不足之处：

1、对于HADOOP系统使用不够深刻，平时只会运用简单的HDFS分布式文件系统
2、对与MAP-REDUCE的编程模型理解不够，许多代码只能借鉴。
3、Spark的理解不够深入，只实现了最简单的KNN算法，其他算法都是调库
4、菜是原罪
5、菜是原罪
6、菜是原罪


Thanks