df1

Cmd 1

```
1 import sys.process._
2 "wget -P /tmp http://www.utdallas.edu/~axn112530/cs6350/dflab/car-milage.csv" !!
```

```
--2017-06-26 20:27:22--  http://www.utdallas.edu/~axn112530/cs6350/dflab/car-milage.csv
Resolving www.utdallas.edu (www.utdallas.edu)... 104.16.43.54, 104.16.44.54, 2400:cb00:2048:1::6810:2c36, ...
Connecting to www.utdallas.edu (www.utdallas.edu)|104.16.43.54|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/csv]
Saving to: ‘/tmp/car-milage.csv’

     0K .                                                     128M=0s

2017-06-26 20:27:22 (128 MB/s) - ‘/tmp/car-milage.csv’ saved [1569]

warning: there were 1 feature warning(s); re-run with -feature for details
import sys.process._
res0: String = ""
Command took 17.49 seconds -- by louhy1128@gmail.com at 2017-6-26 15:26:26 on My Cluster
```

Cmd 2

```
1 val filePath = "file:/tmp/car-milage.csv"
```

```
filePath: String = file:/tmp/car-milage.csv
Command took 0.17 seconds -- by louhy1128@gmail.com at 2017-6-26 15:29:52 on My Cluster
```

Cmd 3

```
1 val cars = spark.read.option("header","true").
  option("inferSchema","true").csv(filePath)
```

▸ (2) Spark Jobs

```
cars: org.apache.spark.sql.DataFrame = [mpg: double, displacement: double ... 10 more fields]
Command took 3.75 seconds -- by louhy1128@gmail.com at 2017-6-26 15:29:54 on My Cluster
```

Cmd 4

```
1 display(cars)
```

▸ (1) Spark Jobs

mpg	displacement	hp	torque	CRatio	RARatio	Ca
16.5	350	155	250	8.5	3.08	4
36.5	85.3	80	83	8.5	3.89	2
21.5	171	109	146	8.2	3.22	2

19.7	258	110	195	8	3.08	1
20.3	140	83	109	8.4	3.4	2
17.8	302	129	220	8	3	2
14.39	500	190	360	8.5	2.73	4
14.89	440	215	330	8.2	2.71	4
17.8	350	155	250	8.5	3.08	4

Command took 1.23 seconds -- by louhy1128@gmail.com at 2017-6-26 15:30:17 on My Cluster

Cmd 5

```
1  display(cars.take(10))
```

▸ (1) Spark Jobs

col_0	col_1	col_2	col_3	col_4	col_5
18.9	350	165	260	8	2.56
17	350	170	275	8.5	2.56
20	250	105	185	8.25	2.73
18.25	351	143	255	8	3
20.07	225	95	170	8.4	2.76
11.2	440	215	330	8.2	2.88
22.12	231	110	175	8	2.56
21.47	262	110	200	8.5	2.56
34.7	89.7	70	81	8.2	3.9

Command took 1.29 seconds -- by louhy1128@gmail.com at 2017-6-26 15:30:56 on My Cluster

Cmd 6

```
1  println("Cars has "+cars.count()+" rows")
```

▸ (1) Spark Jobs

```
Cars has 32 rows
```

Command took 0.65 seconds -- by louhy1128@gmail.com at 2017-6-26 15:32:24 on My Cluster

Cmd 7

```
1  cars.printSchema()
```

```
root
 |-- mpg: double (nullable = true)
 |-- displacement: double (nullable = true)
 |-- hp: integer (nullable = true)
 |-- torque: integer (nullable = true)
 |-- CRatio: double (nullable = true)
```

```
 |-- RARatio: double (nullable = true)
 |-- CarbBarrells: integer (nullable = true)
 |-- NoOfSpeed: integer (nullable = true)
 |-- length: double (nullable = true)
 |-- width: double (nullable = true)
 |-- weight: integer (nullable = true)
 |-- automatic: integer (nullable = true)
```

Command took 0.18 seconds -- by louhy1128@gmail.com at 2017-6-26 15:33:06 on My Cluster

Cmd 8

```
1 cars.describe("mpg","hp","weight","automatic").show()
```

▸ (1) Spark Jobs

```
+-------+-----------------+-----------------+---------------+------------------
-+
|summary|              mpg|               hp|         weight|          automati
c|
+-------+-----------------+-----------------+---------------+------------------
-+
|  count|               32|               32|             32|                 3
2|
|   mean|        20.223125|          136.875|      3586.6875|            0.7187
5|
| stddev|6.318289089312789|44.98082028541039|947.943187269323|0.4568034093991743
5|
|    min|             11.2|               70|           1905|
 0|
|    max|             36.5|              223|           5430|
 1|
+-------+-----------------+-----------------+---------------+------------------
-+
```

Command took 0.99 seconds -- by louhy1128@gmail.com at 2017-6-26 15:34:14 on My Cluster

Cmd 9

```
1 cars.groupBy("automatic").avg("mpg","torque").show()
```

▸ (5) Spark Jobs

```
+---------+------------------+-----------------+
|automatic|          avg(mpg)|      avg(torque)|
+---------+------------------+-----------------+
|        1|17.324782608695646|257.3636363636364|
|        0|27.630000000000006|          109.375|
+---------+------------------+-----------------+
```

Command took 1.94 seconds -- by louhy1128@gmail.com at 2017-6-26 15:36:55 on My Cluster

Cmd 10

```
1 import org.apache.spark.sql.functions.{avg,mean}
2 cars.agg(avg(cars("mpg")), mean(cars("torque")) ).show()
```

▸ (1) Spark Jobs

```
+---------+-----------+
| avg(mpg)|avg(torque)|
+---------+-----------+
|20.223125|      217.9|
+---------+-----------+

import org.apache.spark.sql.functions.{avg, mean}
```

Command took 0.48 seconds -- by louhy1128@gmail.com at 2017-6-26 15:41:35 on My Cluster

Cmd 11

```
1 val cor = cars.stat.corr("hp","weight")
2 println("hp to weight : Correlation = %.4f".format(cor))
```

▸ (1) Spark Jobs

```
hp to weight : Correlation = 0.8834
cor: Double = 0.8834003785623672
```

Command took 0.30 seconds -- by louhy1128@gmail.com at 2017-6-26 15:43:34 on My Cluster

Cmd 12

```
1 import sys.process._
2 "wget -P /tmp
  http://www.utdallas.edu/~axn112530/cs6350/dflab/titanic3_02.csv" !!
3
```

```
--2017-06-26 20:44:18--  http://www.utdallas.edu/~axn112530/cs6350/dflab/titanic
3_02.csv
Resolving www.utdallas.edu (www.utdallas.edu)... 104.16.43.54, 104.16.44.54, 240
0:cb00:2048:1::6810:2b36, ...
Connecting to www.utdallas.edu (www.utdallas.edu)|104.16.43.54|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/csv]
Saving to: ‘/tmp/titanic3_02.csv’

     0K .......... .......... .......... .......... .......... 3.28M
    50K .......... .......... .......... .......... .......... 8.07M
   100K ...                                                    3.34M=0.02s

2017-06-26 20:44:18 (4.61 MB/s) - ‘/tmp/titanic3_02.csv’ saved [105784]

warning: there were 1 feature warning(s); re-run with -feature for details
import sys.process._
res10: String = ""
```

Command took 0.33 seconds -- by louhy1128@gmail.com at 2017-6-26 15:44:18 on My Cluster

Cmd 13

```
1
2 val filePath = "file:/tmp/titanic3_02.csv"
```

```
filePath: String = file:/tmp/titanic3_02.csv
```

Command took 0.09 seconds -- by louhy1128@gmail.com at 2017-6-26 15:46:10 on My Cluster

Cmd 14

```
1 val passengers = spark.read.option("header","true").
  option("inferSchema","true"). csv(filePath)
2
```

▸ (2) Spark Jobs

```
passengers: org.apache.spark.sql.DataFrame = [Pclass: int, Survived: int ... 12
 more fields]
```

Command took 0.76 seconds -- by louhy1128@gmail.com at 2017-6-26 15:46:16 on My Cluster

Cmd 15

```
1 display(passengers)
```

▸ (1) Spark Jobs

Pclass	Survived	Name	Gender	Age	Sib
1	1	Allen, Miss. Elisabeth Walton	female	29	0
1	1	Allison, Master. Hudson Trevor	male	0.9167	1
1	0	Allison, Miss. Helen Loraine	female	2	1
1	0	Allison, Mr. Hudson Joshua Creighton	male	30	1
1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25	1
1	1	Anderson, Mr. Harry	male	48	0
1	1	Andrews, Miss. Kornelia Theodosia	female	63	1
1	0	Andrews, Mr. Thomas Jr	male	39	0
1	1	Appleton, Mrs. Edward Dale (Charlotte Lamson)	female	53	2

Showing the first 1000 rows.

Command took 0.28 seconds -- by louhy1128@gmail.com at 2017-6-26 15:51:02 on My Cluster

Cmd 16

```
1 println("Passengers has "+passengers.count()+" rows")
```

▸ (1) Spark Jobs

```
Passengers has 1309 rows
```

Command took 0.20 seconds -- by louhy1128@gmail.com at 2017-6-26 15:50:45 on My Cluster

Cmd 17

```
1 val passengers1 =
  passengers.select(passengers("Pclass"),passengers("Survived"),passengers("G
  ender"),passengers("Age"),passengers("SibSp"),passengers("Parch"),passenger
  s("Fare"))
2
```

```
passengers1: org.apache.spark.sql.DataFrame = [Pclass: int, Survived: int ... 5
 more fields]
```

Command took 0.16 seconds -- by louhy1128@gmail.com at 2017-6-26 15:47:16 on My Cluster

Cmd 18

```
1 passengers1.schema
2
```

```
res12: org.apache.spark.sql.types.StructType = StructType(StructField(Pclass,Int
egerType,true), StructField(Survived,IntegerType,true), StructField(Gender,Strin
gType,true), StructField(Age,DoubleType,true), StructField(SibSp,IntegerType,tru
e), StructField(Parch,IntegerType,true), StructField(Fare,DoubleType,true))
```

Command took 0.12 seconds -- by louhy1128@gmail.com at 2017-6-26 15:48:12 on My Cluster

Cmd 19

```
1 passengers1.groupBy("Gender").avg().show()
```

▸ (5) Spark Jobs

```
+------+------------------+------------------+------------------+-------------
----+------------------+------------------+
|Gender|       avg(Pclass)|      avg(Survived)|          avg(Age)|        avg(Si
bSp)|        avg(Parch)|         avg(Fare)|
+------+------------------+------------------+------------------+-------------
----+------------------+------------------+
|female|2.1545064377682404| 0.7274678111587983|  28.6870706185567|0.652360515021
4592|0.6330472103004292| 46.19809656652367|
|  male| 2.372479240806643|0.19098457888493475|30.585232978723408|0.413997627520
7592|0.2479240806642942|26.154600831353797|
+------+------------------+------------------+------------------+-------------
----+------------------+------------------+
```

Command took 2.39 seconds -- by louhy1128@gmail.com at 2017-6-26 15:49:17 on My Cluster

Cmd 20

```
1 passengers1.groupBy("Gender").count().show()
```

▸ (5) Spark Jobs

```
+------+-----+
|Gender|count|
+------+-----+
|female|  466|
|  male|  843|
```

```
+------+-----+
```

Command took 0.93 seconds -- by louhy1128@gmail.com at 2017-6-26 15:49:33 on My Cluster

Cmd 21

```
1 passengers1.stat.crosstab("survived","gender").show()
```

▸ (5) Spark Jobs

```
+---------------+------+----+
|survived_gender|female|male|
+---------------+------+----+
|              1|   339| 161|
|              0|   127| 682|
+---------------+------+----+
```

Command took 1.06 seconds -- by louhy1128@gmail.com at 2017-6-26 15:52:35 on My Cluster

Cmd 22

```
1 import sys.process._
2 "wget -P /tmp
  http://www.utdallas.edu/~axn112530/cs6350/dflab/bankruptcy.data.txt" !!
```

```
--2017-06-26 21:09:06--  http://www.utdallas.edu/~axn112530/cs6350/dflab/bankrup
tcy.data.txt
Resolving www.utdallas.edu (www.utdallas.edu)... 104.16.43.54, 104.16.44.54, 240
0:cb00:2048:1::6810:2c36, ...
Connecting to www.utdallas.edu (www.utdallas.edu)|104.16.43.54|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]
Saving to: ‘/tmp/bankruptcy.data.txt’

     0K ...                                                   349M=0s

2017-06-26 21:09:06 (349 MB/s) - ‘/tmp/bankruptcy.data.txt’ saved [3893]

warning: there were 1 feature warning(s); re-run with -feature for details
import sys.process._
res18: String = ""
```

Command took 0.21 seconds -- by louhy1128@gmail.com at 2017-6-26 16:09:06 on My Cluster

Cmd 23

```
1 import org.apache.spark.mllib.evaluation.MulticlassMetrics
2 import org.apache.spark.mllib.classification.{LogisticRegressionWithLBFGS,
  LogisticRegressionModel}
3 import org.apache.spark.mllib.regression.LabeledPoint
4 import org.apache.spark.mllib.linalg.{Vector, Vectors}
```

```
import org.apache.spark.mllib.evaluation.MulticlassMetrics
import org.apache.spark.mllib.classification.{LogisticRegressionWithLBFGS, Logis
ticRegressionModel}
```

```
import org.apache.spark.mllib.regression.LabeledPoint
import org.apache.spark.mllib.linalg.{Vector, Vectors}
```

Command took 0.12 seconds -- by louhy1128@gmail.com at 2017-6-26 16:14:17 on My Cluster

Cmd 24

```
1  val data = sc.textFile("file:/tmp/bankruptcy.data.txt")
```

```
data: org.apache.spark.rdd.RDD[String] = file:/tmp/bankruptcy.data.txt MapPartit
ionsRDD[82] at textFile at <console>:45
```

Command took 0.19 seconds -- by louhy1128@gmail.com at 2017-6-26 16:10:30 on My Cluster

Cmd 25

```
1  data.count()
```

▸ (1) Spark Jobs

```
res22: Long = 250
```

Command took 0.12 seconds -- by louhy1128@gmail.com at 2017-6-26 16:10:34 on My Cluster

Cmd 26

```
1  data.collect().foreach(println)
```

▸ (1) Spark Jobs

```
P,P,A,A,A,P,NB
N,N,A,A,A,N,NB
A,A,A,A,A,A,NB
P,P,P,P,P,P,NB
N,N,P,P,P,N,NB
A,A,P,P,P,A,NB
P,P,A,P,P,P,NB
P,P,P,A,A,P,NB
P,P,A,P,A,P,NB
P,P,A,A,P,P,NB
P,P,P,P,A,P,NB
P,P,P,A,P,P,NB
N,N,A,P,P,N,NB
N,N,P,A,A,N,NB
N,N,A,P,A,N,NB
N,N,A,P,A,N,NB
N,N,A,A,P,N,NB
N,N,P,P,A,N,NB
N,N,P,A,P,N,NB
A,A,A,P,P,A,NB
A,A,P,A,A,A,NB
```

Command took 0.18 seconds -- by louhy1128@gmail.com at 2017-6-26 16:10:36 on My Cluster

Cmd 27

```
def getDoubleValue( input:String ) : Double = {
    var result:Double = 0.0
    if (input == "P")  result = 3.0
    if (input == "A")  result = 2.0
    if (input == "N")  result = 1.0
    if (input == "NB") result = 1.0
    if (input == "B")  result = 0.0
    return result
   }
```

```
getDoubleValue: (input: String)Double
```

Command took 0.14 seconds -- by louhy1128@gmail.com at 2017-6-26 16:09:53 on My Cluster

Cmd 28

```

val parsedData = data.map{line =>
    val parts = line.split(",")
    LabeledPoint(getDoubleValue(parts(6)),
Vectors.dense(parts.slice(0,6).map(x => getDoubleValue(x))))
}
```

```
parsedData: org.apache.spark.rdd.RDD[org.apache.spark.mllib.regression.LabeledPo
int] = MapPartitionsRDD[83] at map at <console>:53
```

Command took 0.27 seconds -- by louhy1128@gmail.com at 2017-6-26 16:14:22 on My Cluster

Cmd 29

```
parsedData.collect().foreach(println)

```

▸ (1) Spark Jobs

```
(1.0,[3.0,3.0,2.0,2.0,2.0,3.0])
(1.0,[1.0,1.0,2.0,2.0,2.0,1.0])
(1.0,[2.0,2.0,2.0,2.0,2.0,2.0])
(1.0,[3.0,3.0,3.0,3.0,3.0,3.0])
(1.0,[1.0,1.0,3.0,3.0,3.0,1.0])
(1.0,[2.0,2.0,3.0,3.0,3.0,2.0])
(1.0,[3.0,3.0,2.0,3.0,3.0,3.0])
(1.0,[3.0,3.0,3.0,2.0,2.0,3.0])
(1.0,[3.0,3.0,2.0,3.0,2.0,3.0])
(1.0,[3.0,3.0,2.0,2.0,3.0,3.0])
(1.0,[3.0,3.0,3.0,3.0,2.0,3.0])
(1.0,[3.0,3.0,3.0,2.0,3.0,3.0])
(1.0,[1.0,1.0,2.0,3.0,3.0,1.0])
(1.0,[1.0,1.0,3.0,2.0,2.0,1.0])
(1.0,[1.0,1.0,2.0,3.0,2.0,1.0])
(1.0,[1.0,1.0,2.0,3.0,2.0,1.0])
(1.0,[1.0,1.0,2.0,2.0,3.0,1.0])
(1.0,[1.0,1.0,3.0,3.0,2.0,1.0])
```

```
(1.0,[1.0,1.0,3.0,2.0,3.0,1.0])
(1.0,[2.0,2.0,2.0,3.0,3.0,2.0])
```

Command took 0.27 seconds -- by louhy1128@gmail.com at 2017-6-26 16:15:09 on My Cluster

Cmd 30

```
1  println(parsedData.take(10).mkString("\n"))
2
```

▸ (1) Spark Jobs

```
(1.0,[3.0,3.0,2.0,2.0,2.0,3.0])
(1.0,[1.0,1.0,2.0,2.0,2.0,1.0])
(1.0,[2.0,2.0,2.0,2.0,2.0,2.0])
(1.0,[3.0,3.0,3.0,3.0,3.0,3.0])
(1.0,[1.0,1.0,3.0,3.0,3.0,1.0])
(1.0,[2.0,2.0,3.0,3.0,3.0,2.0])
(1.0,[3.0,3.0,2.0,3.0,3.0,3.0])
(1.0,[3.0,3.0,3.0,2.0,2.0,3.0])
(1.0,[3.0,3.0,2.0,3.0,2.0,3.0])
(1.0,[3.0,3.0,2.0,2.0,3.0,3.0])
```

Command took 0.35 seconds -- by louhy1128@gmail.com at 2017-6-26 16:15:20 on My Cluster

Cmd 31

```
1  val splits = parsedData.randomSplit(Array(0.6, 0.4), seed = 11L)
2
```

```
splits: Array[org.apache.spark.rdd.RDD[org.apache.spark.mllib.regression.Labeled
Point]] = Array(MapPartitionsRDD[129] at randomSplit at <console>:55, MapPartiti
onsRDD[130] at randomSplit at <console>:55)
```

Command took 0.14 seconds -- by louhy1128@gmail.com at 2017-6-26 16:17:35 on My Cluster

Cmd 32

```
1  val trainingData = splits(0)
```

```
trainingData: org.apache.spark.rdd.RDD[org.apache.spark.mllib.regression.Labeled
Point] = MapPartitionsRDD[129] at randomSplit at <console>:55
```

Command took 0.09 seconds -- by louhy1128@gmail.com at 2017-6-26 16:17:37 on My Cluster

Cmd 33

```
1  val testData = splits(1)
```

```
testData: org.apache.spark.rdd.RDD[org.apache.spark.mllib.regression.LabeledPoin
t] = MapPartitionsRDD[130] at randomSplit at <console>:55
```

Command took 0.09 seconds -- by louhy1128@gmail.com at 2017-6-26 16:17:38 on My Cluster

Cmd 34

```
1  trainingData.take(10).foreach(println)
2
```

▸ (1) Spark Jobs

```
(1.0,[3.0,3.0,2.0,2.0,2.0,3.0])
(1.0,[1.0,1.0,2.0,2.0,2.0,1.0])
```

```
(1.0,[1.0,1.0,3.0,3.0,3.0,1.0])
(1.0,[2.0,2.0,3.0,3.0,3.0,2.0])
(1.0,[3.0,3.0,2.0,3.0,3.0,3.0])
(1.0,[3.0,3.0,3.0,2.0,2.0,3.0])
(1.0,[3.0,3.0,2.0,3.0,2.0,3.0])
(1.0,[1.0,1.0,2.0,3.0,2.0,1.0])
(1.0,[1.0,1.0,2.0,3.0,2.0,1.0])
(1.0,[1.0,1.0,2.0,2.0,3.0,1.0])
```

Command took 0.63 seconds -- by louhy1128@gmail.com at 2017-6-26 16:15:40 on My Cluster

Cmd 35

```
1 val model = new
  LogisticRegressionWithLBFGS().setNumClasses(2).run(trainingData)
2
```

▸ (30) Spark Jobs

```
model: org.apache.spark.mllib.classification.LogisticRegressionModel = org.apach
e.spark.mllib.classification.LogisticRegressionModel: intercept = 0.0, numFeatur
es = 6, numClasses = 2, threshold = 0.5
```

Command took 2.50 seconds -- by louhy1128@gmail.com at 2017-6-26 16:15:58 on My Cluster

Cmd 36

```
1 val parsedData1 = data.map{line =>
2      line.split(",").map(getDoubleValue).mkString(",")
3 }
```

```
parsedData1: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[127] at map at
 <console>:53
```

Command took 0.20 seconds -- by louhy1128@gmail.com at 2017-6-26 16:16:28 on My Cluster

Cmd 37

```
1 val labelAndPreds = testData.map { point =>
2   val prediction = model.predict(point.features)
3   (point.label, prediction)
4 }
5
```

```
labelAndPreds: org.apache.spark.rdd.RDD[(Double, Double)] = MapPartitionsRDD[12
8] at map at <console>:63
```

Command took 0.31 seconds -- by louhy1128@gmail.com at 2017-6-26 16:16:46 on My Cluster

Cmd 38

```
1 labelAndPreds.collect.foreach(println)
```

▸ (1) Spark Jobs

```
(1.0,1.0)
(1.0,1.0)
(1.0,1.0)
(1.0,0.0)
(1.0,1.0)
```

```
(1.0,1.0)
(1.0,1.0)
(1.0,1.0)
(1.0,1.0)
(1.0,1.0)
(1.0,1.0)
(1.0,0.0)
(1.0,1.0)
(1.0,1.0)
(1.0,1.0)
(1.0,1.0)
(1.0,1.0)
(1.0,1.0)
(1.0,1.0)
(1.0,1.0)
```

Command took 0.20 seconds -- by louhy1128@gmail.com at 2017-6-26 16:20:02 on My Cluster

Cmd 39

```
1 val trainErr = labelAndPreds.filter(r => r._1 != r._2).count.toDouble /
  testData.count
```

▸ (2) Spark Jobs

```
trainErr: Double = 0.20408163265306123
```

Command took 0.15 seconds -- by louhy1128@gmail.com at 2017-6-26 16:21:25 on My Cluster

Cmd 40

```
1 import sys.process._
2 "wget -P /tmp http://www.utdallas.edu/~axn112530/cs6350/dflab/car-
  milage.csv" !!
```

```
--2017-06-26 21:23:13--  http://www.utdallas.edu/~axn112530/cs6350/dflab/car-mil
age.csv
Resolving www.utdallas.edu (www.utdallas.edu)... 104.16.44.54, 104.16.43.54, 240
0:cb00:2048:1::6810:2b36, ...
Connecting to www.utdallas.edu (www.utdallas.edu)|104.16.44.54|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/csv]
Saving to: ‘/tmp/car-milage.csv.1’

     0K .                                                     150M=0s

2017-06-26 21:23:13 (150 MB/s) - ‘/tmp/car-milage.csv.1’ saved [1569]

warning: there were 1 feature warning(s); re-run with -feature for details
import sys.process._
res30: String = ""
```

Command took 0.23 seconds -- by louhy1128@gmail.com at 2017-6-26 16:23:13 on My Cluster

Cmd 41

```
1 import org.apache.spark.sql.SparkSession
2 import org.apache.spark.sql.functions.corr
3 import org.apache.spark.ml.regression.LinearRegression
4 import org.apache.spark.ml.feature.VectorAssembler
5 import org.apache.spark.ml.linalg.Vectors
6 import org.apache.spark.ml.evaluation.RegressionEvaluator
```

```
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions.corr
import org.apache.spark.ml.regression.LinearRegression
import org.apache.spark.ml.feature.VectorAssembler
import org.apache.spark.ml.linalg.Vectors
import org.apache.spark.ml.evaluation.RegressionEvaluator
```

Command took 0.17 seconds -- by louhy1128@gmail.com at 2017-6-26 16:23:24 on My Cluster

Cmd 42

```
1 val cars1 = cars.na.drop()
2     val assembler = new VectorAssembler()
3
  assembler.setInputCols(Array("displacement","hp","torque","CRatio","RARatio
  ","CarbBarrells","NoOfSpeed","length","width","weight","automatic"))
4     assembler.setOutputCol("features")
5     val cars2 = assembler.transform(cars1)
```

```
cars1: org.apache.spark.sql.DataFrame = [mpg: double, displacement: double ... 1
0 more fields]
assembler: org.apache.spark.ml.feature.VectorAssembler = vecAssembler_77f9e9ed0d
9f
cars2: org.apache.spark.sql.DataFrame = [mpg: double, displacement: double ... 1
1 more fields]
```

Command took 0.31 seconds -- by louhy1128@gmail.com at 2017-6-26 16:34:07 on My Cluster

Cmd 43

```
1     cars2.show(40)
```

▸ (1) Spark Jobs

```
+-----+------------+---+------+------+-------+------------+---------+------+----
-+------+---------+--------------------+
|  mpg|displacement| hp|torque|CRatio|RARatio|CarbBarrells|NoOfSpeed|length|widt
h|weight|automatic|            features|
+-----+------------+---+------+------+-------+------------+---------+------+----
-+------+---------+--------------------+
| 18.9|       350.0|165|   260|   8.0|   2.56|           4|        3| 200.3| 69.
9|  3910|        1|[350.0,165.0,260....|
| 17.0|       350.0|170|   275|   8.5|   2.56|           4|        3| 199.6| 72.
9|  3860|        1|[350.0,170.0,275....|
| 20.0|       250.0|105|   185|  8.25|   2.73|           1|        3| 196.7| 72.
2|  3510|        1|[250.0,105.0,185....|
```

```
|18.25|       351.0|143|   255|   8.0|    3.0|           2|        3| 199.9| 74.
0|  3890|        1|[351.0,143.0,255....|
|20.07|       225.0| 95|   170|   8.4|   2.76|           1|        3| 194.1| 71.
8|  3365|        0|[225.0,95.0,170.0...|
| 11.2|       440.0|215|   330|   8.2|   2.88|           4|        3| 184.5| 69.
0|  4215|        1|[440.0,215.0,330....|
|22.12|       231.0|110|   175|   8.0|   2.56|           2|        3| 179.3| 65.
4|  3020|        1|[231.0,110.0,175....|
```

Command took 0.24 seconds -- by louhy1128@gmail.com at 2017-6-26 16:34:10 on My Cluster

Cmd 44

```
1 val train = cars2.filter(cars1("weight") <= 4000)
2     val test = cars2.filter(cars1("weight") > 4000)
```

```
train: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [mpg: double, di
splacement: double ... 11 more fields]
test: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [mpg: double, dis
placement: double ... 11 more fields]
```

Command took 0.16 seconds -- by louhy1128@gmail.com at 2017-6-26 16:34:55 on My Cluster

Cmd 45

```
1 test.show()
2     println("Train = "+train.count()+" Test = "+test.count())
```

▸ (3) Spark Jobs

```
+-----+------------+---+------+------+-------+------------+---------+------+----
-+------+---------+--------------------+
|  mpg|displacement| hp|torque|CRatio|RARatio|CarbBarrells|NoOfSpeed|length|widt
h|weight|automatic|            features|
+-----+------------+---+------+------+-------+------------+---------+------+----
-+------+---------+--------------------+
| 11.2|       440.0|215|   330|   8.2|   2.88|           4|        3| 184.5| 69.
0|  4215|        1|[440.0,215.0,330....|
|14.39|       500.0|190|   360|   8.5|   2.73|           4|        3| 224.1| 79.
8|  5290|        1|[500.0,190.0,360....|
|14.89|       440.0|215|   330|   8.2|   2.71|           4|        3| 231.0| 79.
7|  5185|        1|[440.0,215.0,330....|
|21.47|       360.0|180|   290|   8.4|   2.45|           2|        3| 214.2| 76.
3|  4250|        1|[360.0,180.0,290....|
|13.27|       460.0|223|   366|   8.0|    3.0|           4|        3| 228.0| 79.
8|  5430|        1|[460.0,223.0,366....|
|19.73|       318.0|140|   255|   8.5|   2.71|           2|        3| 215.3| 76.
3|  4370|        1|[318.0,140.0,255....|
| 13.9|       351.0|148|   243|   8.0|   3.25|           2|        3| 215.5| 78.
5|  4540|        1|[351.0,148.0,243....|
|13.27|       351.0|148|   243|   8.0|   3.26|           2|        3| 216.1| 78.
5|  4715|        1|[351.0,148.0,243....|
|13.77|       360.0|195|   295|  8.25|   3.15|           4|        3| 209.3| 77.
4|  4215|        1|[360.0,195.0,295....|
```

```
+-----+------------+---+------+------+-------+------------+---------+------+----
-+------+---------+--------------------+

Train = 21 Test = 9
```

Command took 0.59 seconds -- by louhy1128@gmail.com at 2017-6-26 16:35:28 on My Cluster

Cmd 46

```
1  val algLR = new LinearRegression()
2          algLR.setMaxIter(100)
3          algLR.setRegParam(0.3)
4          algLR.setElasticNetParam(0.8)
5          algLR.setLabelCol("mpg")
```

```
algLR: org.apache.spark.ml.regression.LinearRegression = linReg_007d8c99de59
res34: org.apache.spark.ml.regression.LinearRegression = linReg_007d8c99de59
```

Command took 0.21 seconds -- by louhy1128@gmail.com at 2017-6-26 16:36:08 on My Cluster

Cmd 47

```
1  val mdlLR = algLR.fit(train)
```

▸ (5) Spark Jobs

```
mdlLR: org.apache.spark.ml.regression.LinearRegressionModel = linReg_007d8c99de5
9
```

Command took 0.99 seconds -- by louhy1128@gmail.com at 2017-6-26 16:36:22 on My Cluster

Cmd 48

```
1  println(s"Coefficients: ${mdlLR.coefficients} Intercept:
   ${mdlLR.intercept}")
```

```
Coefficients: [0.0,-0.007017141998245329,0.0,0.0,2.9424064428541645,0.0,-0.86281
57004496956,0.0,-0.26534908526629886,-0.004729717739392178,0.0] Intercept: 50.05
862655261031
```

Command took 0.16 seconds -- by louhy1128@gmail.com at 2017-6-26 16:36:55 on My Cluster

Cmd 49

```
1  val trSummary = mdlLR.summary
2  println(s"numIterations: ${trSummary.totalIterations}")
3      println(s"Iteration Summary History:
   ${trSummary.objectiveHistory.toList}")
4      trSummary.residuals.show()
5      println(s"RMSE: ${trSummary.rootMeanSquaredError}")
6      println(s"r2: ${trSummary.r2}")
```

▸ (1) Spark Jobs

```
numIterations: 101
Iteration Summary History: List(0.4999999999999991, 0.39843747544896296, 0.14822
95744826868, 0.14005807692077016, 0.1446968308765583, 0.12669217682960035, 0.125
8630122249318, 0.12511340526765194, 0.12322427437191928, 0.12276798916627535, 0.
12223083208160618, 0.12198963359328388, 0.1217980834932178, 0.12150878758627376,
0.1213259562375511, 0.12091301876948132, 0.12028626315701765, 0.1200516927821073
```

```
8, 0.11977289314390564, 0.1194232265949696, 0.11834850015514281, 0.1181617248486
041, 0.11798849972869874, 0.1177544183274557, 0.11764132783918399, 0.11761053226
921289, 0.1175372291362528, 0.11740134195406475, 0.11730833333681416, 0.11717018
963534281, 0.11717255191139506, 0.11715818488156707, 0.11713935699462788, 0.1171
267413739083, 0.11711722997357507, 0.11710097506014061, 0.11710017828208376, 0.1
1709947582555068, 0.11709917904474389, 0.11709768838483264, 0.1170929119008887,
 0.11708390851967115, 0.11707212479832396, 0.1170662760037003, 0.117061271662565
19, 0.1170522477334453, 0.11704956881759424, 0.11704258835326761, 0.117034256320
00736, 0.11703156832900255, 0.11702939448133705, 0.1170287537028389, 0.117028483
37455823, 0.11702624620585331, 0.11702621917586575, 0.11702618327298192, 0.11702
613294793285, 0.11702611993473151, 0.11702611478925005, 0.11702610986218365, 0.1
170260926507051, 0.11702609220245044, 0.11702609164365266, 0.11702609140604611,
 0.11702609131291232, 0.11702609126614305, 0.11702609123924826, 0.11702609122479
349, 0.11702609121648369, 0.11702609118295512, 0.11702609115548301, 0.1170260911
```

Command took 0.53 seconds -- by louhy1128@gmail.com at 2017-6-26 16:37:34 on My Cluster

Cmd 50

```
println(s"numIterations: ${trSummary.totalIterations}")
    println(s"Iteration Summary History:
${trSummary.objectiveHistory.toList}")
    trSummary.residuals.show()
    println(s"RMSE: ${trSummary.rootMeanSquaredError}")
    println(s"r2: ${trSummary.r2}")
```

▸ (1) Spark Jobs

```
numIterations: 101
Iteration Summary History: List(0.4999999999999991, 0.39843747544896296, 0.14822
95744826868, 0.14005807692077016, 0.1446968308765583, 0.12669217682960035, 0.125
8630122249318, 0.12511340526765194, 0.12322427437191928, 0.12276798916627535, 0.
12223083208160618, 0.12198963359328388, 0.1217980834932178, 0.12150878758627376,
0.1213259562375511, 0.12091301876948132, 0.12028626315701765, 0.1200516927821073
8, 0.11977289314390564, 0.1194232265949696, 0.11834850015514281, 0.1181617248486
041, 0.11798849972869874, 0.1177544183274557, 0.11764132783918399, 0.11761053226
921289, 0.1175372291362528, 0.11740134195406475, 0.11730833333681416, 0.11717018
963534281, 0.11717255191139506, 0.11715818488156707, 0.11713935699462788, 0.1171
267413739083, 0.11711722997357507, 0.11710097506014061, 0.11710017828208376, 0.1
1709947582555068, 0.11709917904474389, 0.11709768838483264, 0.1170929119008887,
 0.11708390851967115, 0.11707212479832396, 0.1170662760037003, 0.117061271662565
19, 0.1170522477334453, 0.11704956881759424, 0.11704258835326761, 0.117034256320
00736, 0.11703156832900255, 0.11702939448133705, 0.1170287537028389, 0.117028483
37455823, 0.11702624620585331, 0.11702621917586575, 0.11702618327298192, 0.11702
613294793285, 0.11702611993473151, 0.11702611478925005, 0.11702610986218365, 0.1
170260926507051, 0.11702609220245044, 0.11702609164365266, 0.11702609140604611,
 0.11702609131291232, 0.11702609126614305, 0.11702609123924826, 0.11702609122479
349, 0.11702609121648369, 0.11702609118295512, 0.11702609115548301, 0.1170260911
4831418, 0.11702609112982439, 0.11702609111569616, 0.11702609110533795, 0.117026
```

Command took 0.35 seconds -- by louhy1128@gmail.com at 2017-6-26 16:42:16 on My Cluster

Cmd 51

```
1  val predictions = mdlLR.transform(test)
2      predictions.show()
```

▸ (1) Spark Jobs

```
+-----+------------+---+------+------+-------+------------+---------+------+----
-+------+---------+--------------------+------------------+
|  mpg|displacement| hp|torque|CRatio|RARatio|CarbBarrells|NoOfSpeed|length|widt
h|weight|automatic|            features|        prediction|
+-----+------------+---+------+------+-------+------------+---------+------+----
-+------+---------+--------------------+------------------+
| 11.2|       440.0|215|   330|   8.2|   2.88|           4|        3| 184.5| 69.
0|  4215|        1|[440.0,215.0,330....|16.190777322145827|
|14.39|       500.0|190|   360|   8.5|   2.73|           4|        3| 224.1| 79.
8|  5290|        1|[500.0,190.0,360....| 7.974628214951217|
|14.89|       440.0|215|   330|   8.2|   2.71|           4|        3| 231.0| 79.
7|  5185|        1|[440.0,215.0,330....| 8.263506807300807|
|21.47|       360.0|180|   290|   8.4|   2.45|           2|        3| 214.2| 76.
3|  4250|        1|[360.0,180.0,290....|13.068554078334408|
|13.27|       460.0|223|   366|   8.0|    3.0|           4|        3| 228.0| 79.
8|  5430|        1|[460.0,223.0,366....| 7.875351785064844|
|19.73|       318.0|140|   255|   8.5|   2.71|           2|        3| 215.3| 76.
3|  4370|        1|[318.0,140.0,255....|13.546699304679244|
| 13.9|       351.0|148|   243|   8.0|   3.25|           2|        3| 215.5| 78.
5|  4540|        1|[351.0,148.0,243....|13.691641644552007|
|13.27|       351.0|148|   243|   8.0|   3.26|           2|        3| 216.1| 78.
5|  4715|        1|[351.0,148.0,243....|12.893365104586913|
|13.77|       360.0|195|   295|  8.25|   3.15|           4|        3| 209.3| 77.
4|  4215|        1|[360.0,195.0,295....|14.896637585444445|
+-----+------------+---+------+------+-------+------------+---------+------+----
-+------+---------+--------------------+------------------+

predictions: org.apache.spark.sql.DataFrame = [mpg: double, displacement: double
... 12 more fields]
```

Command took 0.51 seconds -- by louhy1128@gmail.com at 2017-6-26 16:42:40 on My Cluster

Cmd 52

```
1      val evaluator = new RegressionEvaluator()
2          evaluator.setLabelCol("mpg")
3          val rmse = evaluator.evaluate(predictions)
4          println("Root Mean Squared Error = "+"%6.3f".format(rmse))
```

▸ (1) Spark Jobs

```
Root Mean Squared Error =  5.264
evaluator: org.apache.spark.ml.evaluation.RegressionEvaluator = regEval_de0800cf
1a92
rmse: Double = 5.263602222142872
```

Command took 0.29 seconds -- by louhy1128@gmail.com at 2017-6-26 16:43:09 on My Cluster

Cmd 53

```
1          evaluator.setMetricName("mse")
2          val mse = evaluator.evaluate(predictions)
3          println("Mean Squared Error = "+"%6.3f".format(mse))
```

▸ (1) Spark Jobs

```
Mean Squared Error = 27.706
mse: Double = 27.705508352947376
```

Command took 0.37 seconds -- by louhy1128@gmail.com at 2017-6-26 16:43:26 on My Cluster

Cmd 54

```
1 import org.apache.spark.ml.{Pipeline, PipelineModel}
2 import org.apache.spark.ml.classification.LogisticRegression
3 import org.apache.spark.ml.feature.{HashingTF, Tokenizer}
4 import org.apache.spark.ml.linalg.Vector
5 import org.apache.spark.sql.Row
```

```
import org.apache.spark.ml.{Pipeline, PipelineModel}
import org.apache.spark.ml.classification.LogisticRegression
import org.apache.spark.ml.feature.{HashingTF, Tokenizer}
import org.apache.spark.ml.linalg.Vector
import org.apache.spark.sql.Row
```

Command took 0.12 seconds -- by louhy1128@gmail.com at 2017-6-26 16:44:20 on My Cluster

Cmd 55

```
1 val training = spark.createDataFrame(Seq(
2   (0L, "a b c d e spark", 1.0),
3   (1L, "b d", 0.0),
4   (2L, "spark f g h", 1.0),
5   (3L, "hadoop mapreduce", 0.0)
6 )).toDF("id", "text", "label")
```

```
training: org.apache.spark.sql.DataFrame = [id: bigint, text: string ... 1 more
 field]
```

Command took 0.23 seconds -- by louhy1128@gmail.com at 2017-6-26 16:45:02 on My Cluster

Cmd 56

```
1 training.show()
```

```
+---+----------------+-----+
| id|            text|label|
+---+----------------+-----+
|  0| a b c d e spark|  1.0|
|  1|             b d|  0.0|
|  2|     spark f g h|  1.0|
|  3|hadoop mapreduce|  0.0|
+---+----------------+-----+
```

Command took 0.19 seconds -- by louhy1128@gmail.com at 2017-6-26 16:45:20 on My Cluster

Cmd 57

```
1  val tokenizer = new Tokenizer()
2    .setInputCol("text")
3    .setOutputCol("words")
4  val hashingTF = new HashingTF()
5    .setNumFeatures(1000)
6    .setInputCol(tokenizer.getOutputCol)
7    .setOutputCol("features")
8  val lr = new LogisticRegression()
9    .setMaxIter(10)
10   .setRegParam(0.001)
11 val pipeline = new Pipeline()
12   .setStages(Array(tokenizer, hashingTF, lr))
```

```
tokenizer: org.apache.spark.ml.feature.Tokenizer = tok_7c2e86ba3870
hashingTF: org.apache.spark.ml.feature.HashingTF = hashingTF_a8d014f22d9f
lr: org.apache.spark.ml.classification.LogisticRegression = logreg_916565f90583
pipeline: org.apache.spark.ml.Pipeline = pipeline_24387f0c7838
```

Command took 0.30 seconds -- by louhy1128@gmail.com at 2017-6-26 16:44:47 on My Cluster

Cmd 58

```
1  val model = pipeline.fit(training)
```

▸ (12) Spark Jobs

```
model: org.apache.spark.ml.PipelineModel = pipeline_24387f0c7838
```

Command took 0.56 seconds -- by louhy1128@gmail.com at 2017-6-26 16:46:03 on My Cluster

Cmd 59

```
1  model.write.overwrite().save("/tmp/spark-logistic-regression-model")
```

▸ (5) Spark Jobs

Command took 8.63 seconds -- by louhy1128@gmail.com at 2017-6-26 16:46:40 on My Cluster

Cmd 60

```
1  val test = spark.createDataFrame(Seq(
2    (4L, "spark i j k"),
3    (5L, "l m n"),
4    (6L, "spark hadoop spark"),
5    (7L, "apache hadoop")
6  )).toDF("id", "text")
```

```
test: org.apache.spark.sql.DataFrame = [id: bigint, text: string]
```

Command took 0.36 seconds -- by louhy1128@gmail.com at 2017-6-26 16:47:14 on My Cluster

Cmd 61

```
1  model.transform(test)
```

```
res44: org.apache.spark.sql.DataFrame = [id: bigint, text: string ... 5 more fie
lds]
```

Command took 0.23 seconds -- by louhy1128@gmail.com at 2017-6-26 16:47:56 on My Cluster

Cmd 62

```
model.transform(test)
  .select("id", "text", "probability", "prediction")
  .collect()
  .foreach { case Row(id: Long, text: String, prob: Vector, prediction: Double) =>
    println(s"($id, $text) --> prob=$prob, prediction=$prediction")
  }
```

```
(4, spark i j k) --> prob=[0.15964077387874118,0.8403592261212589], prediction=1.0
(5, l m n) --> prob=[0.8378325685476614,0.16216743145233858], prediction=0.0
(6, spark hadoop spark) --> prob=[0.06926633132976263,0.9307336686702373], prediction=1.0
(7, apache hadoop) --> prob=[0.9821575333444208,0.017842466655579155], prediction=0.0
```

Command took 0.78 seconds -- by louhy1128@gmail.com at 2017-6-26 16:48:15 on My Cluster

Cmd 63

```
import org.apache.spark.ml.Pipeline
import org.apache.spark.ml.classification.LogisticRegression
import org.apache.spark.ml.evaluation.BinaryClassificationEvaluator
import org.apache.spark.ml.feature.{HashingTF, Tokenizer}
import org.apache.spark.ml.linalg.Vector
import org.apache.spark.ml.tuning.{CrossValidator, ParamGridBuilder}
import org.apache.spark.sql.Row

```

```
import org.apache.spark.ml.Pipeline
import org.apache.spark.ml.classification.LogisticRegression
import org.apache.spark.ml.evaluation.BinaryClassificationEvaluator
import org.apache.spark.ml.feature.{HashingTF, Tokenizer}
import org.apache.spark.ml.linalg.Vector
import org.apache.spark.ml.tuning.{CrossValidator, ParamGridBuilder}
import org.apache.spark.sql.Row
```

Command took 0.14 seconds -- by louhy1128@gmail.com at 2017-6-26 16:50:37 on My Cluster

Cmd 64

```
val tokenizer = new Tokenizer()
  .setInputCol("text")
  .setOutputCol("words")
val hashingTF = new HashingTF()
  .setInputCol(tokenizer.getOutputCol)
  .setOutputCol("features")
val lr = new LogisticRegression()
  .setMaxIter(10)
val pipeline = new Pipeline()
  .setStages(Array(tokenizer, hashingTF, lr))
```

```
tokenizer: org.apache.spark.ml.feature.Tokenizer = tok_ccf626e9f0e2
hashingTF: org.apache.spark.ml.feature.HashingTF = hashingTF_af7de3b54858
lr: org.apache.spark.ml.classification.LogisticRegression = logreg_7cc0b254e258
pipeline: org.apache.spark.ml.Pipeline = pipeline_6cbd1e43077b
```

Command took 0.26 seconds -- by louhy1128@gmail.com at 2017-6-26 16:50:49 on My Cluster

Cmd 65

```
1 val paramGrid = new ParamGridBuilder()
2   .addGrid(hashingTF.numFeatures, Array(10, 100, 1000))
3   .addGrid(lr.regParam, Array(0.1, 0.01))
4   .build()
```

```
paramGrid: Array[org.apache.spark.ml.param.ParamMap] =
Array({
        hashingTF_af7de3b54858-numFeatures: 10,
        logreg_7cc0b254e258-regParam: 0.1
}, {
        hashingTF_af7de3b54858-numFeatures: 10,
        logreg_7cc0b254e258-regParam: 0.01
}, {
        hashingTF_af7de3b54858-numFeatures: 100,
        logreg_7cc0b254e258-regParam: 0.1
}, {
        hashingTF_af7de3b54858-numFeatures: 100,
        logreg_7cc0b254e258-regParam: 0.01
}, {
        hashingTF_af7de3b54858-numFeatures: 1000,
        logreg_7cc0b254e258-regParam: 0.1
}, {
        hashingTF_af7de3b54858-numFeatures: 1000,
        logreg_7cc0b254e258-regParam: 0.01
})
```

Command took 0.24 seconds -- by louhy1128@gmail.com at 2017-6-26 16:51:04 on My Cluster

Cmd 66

```
1 val cv = new CrossValidator()
2   .setEstimator(pipeline)
3   .setEvaluator(new BinaryClassificationEvaluator)
4   .setEstimatorParamMaps(paramGrid)
5   .setNumFolds(2)  // Use 3+ in practice
```

```
cv: org.apache.spark.ml.tuning.CrossValidator = cv_64bfccab012a
```

Command took 0.32 seconds -- by louhy1128@gmail.com at 2017-6-26 16:51:23 on My Cluster

Cmd 67

```
1 val cvModel = cv.fit(training)
2
```

▸ (49) Spark Jobs

```
cvModel: org.apache.spark.ml.tuning.CrossValidatorModel = cv_64bfccab012a
```

Command took 15.32 seconds -- by louhy1128@gmail.com at 2017-6-26 16:51:36 on My Cluster

Cmd 68

```
1 val test = spark.createDataFrame(Seq(
2   (4L, "spark i j k"),
3   (5L, "l m n"),
4   (6L, "mapreduce spark"),
5   (7L, "apache hadoop")
6 )).toDF("id", "text")
```

```
test: org.apache.spark.sql.DataFrame = [id: bigint, text: string]
```

Command took 0.29 seconds -- by louhy1128@gmail.com at 2017-6-26 16:53:15 on My Cluster

Cmd 69

```
1 cvModel.transform(test)
2   .select("id", "text", "probability", "prediction")
3   .collect()
4   .foreach { case Row(id: Long, text: String, prob: Vector, prediction:
  Double) =>
5     println(s"($id, $text) --> prob=$prob, prediction=$prediction")
6   }
```

```
(4, spark i j k) --> prob=[0.5851763701100572,0.4148236298899428], prediction=0.
0
(5, l m n) --> prob=[0.3013178552938834,0.6986821447061167], prediction=1.0
(6, mapreduce spark) --> prob=[0.7190457812185589,0.28095421878144106], predicti
on=0.0
(7, apache hadoop) --> prob=[0.7291932008053432,0.2708067991946568], prediction=
0.0
```

Command took 0.46 seconds -- by louhy1128@gmail.com at 2017-6-26 16:54:33 on My Cluster

Cmd 70

```
1 cvModel.bestModel
```

```
res47: org.apache.spark.ml.Model[_] = pipeline_6cbd1e43077b
```

Command took 0.19 seconds -- by louhy1128@gmail.com at 2017-6-26 16:55:22 on My Cluster

Cmd 71

```
res48: Array[org.apache.spark.ml.param.Param[_]] = Array()
```

Command took 0.19 seconds -- by louhy1128@gmail.com at 2017-6-26 16:56:38 on My Cluster