7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

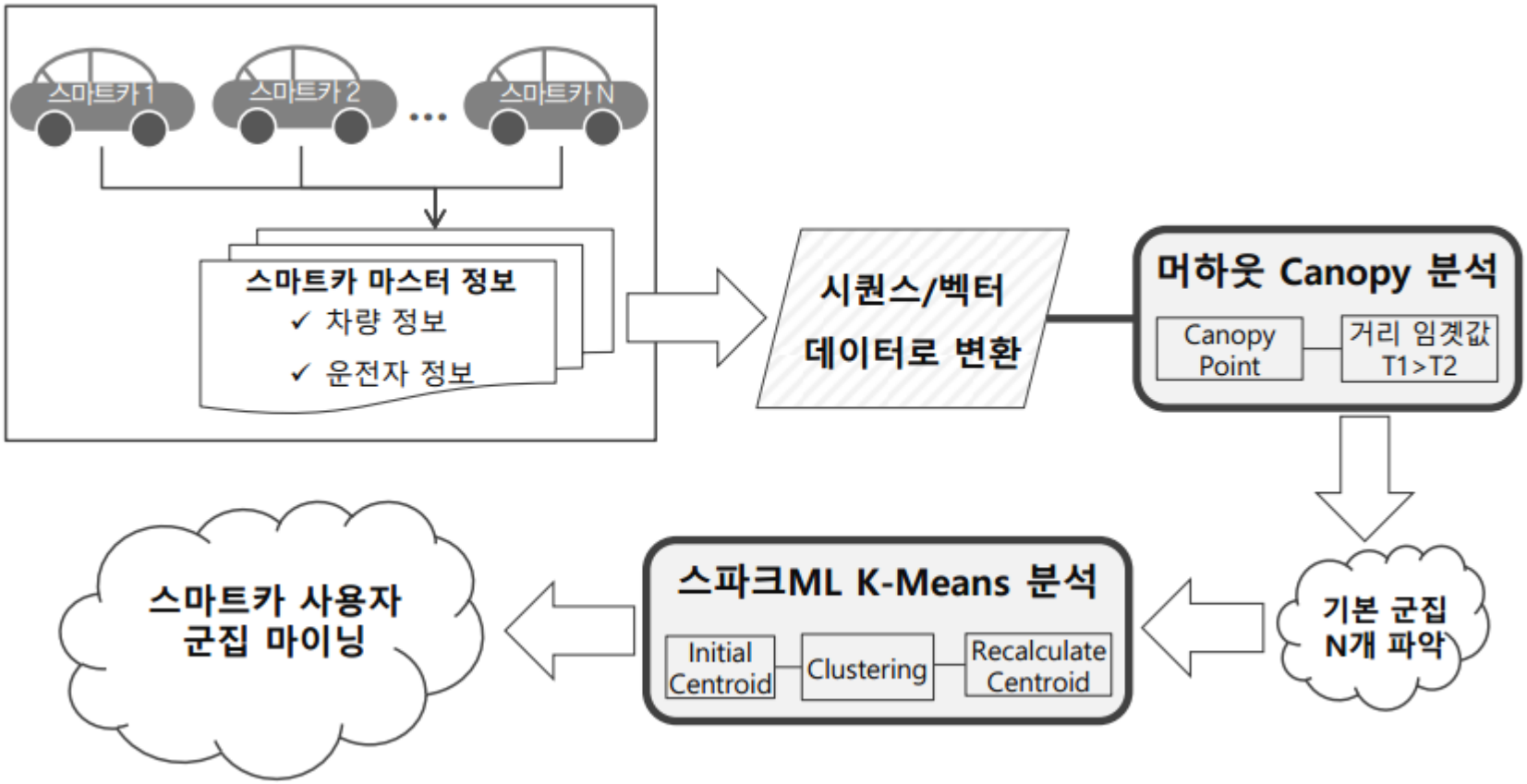
스마트카 고객정보 군집 분석1

데이터 마이닝 중 3번째인 군집 분석을 진행한다. 사용될 데이터셋은 "스마트카 고객 마스터 정보"로 하이브의 External 영역에 SmartCar_Master 테이블이다. 총 2,600명의 스마트카 사용 고객 정보가 적재돼 있다.

	car_number	car_capacity	car_model	owner_sex	owner_age	owner_marriage	owner_job	owner_region
0	A0001	2000	2008	여	68	미혼	공무원	서울
1	A0002	1500	2007	남	66	기혼	회사원	인천
2	A0003	3000	2009	남	23	미혼	무직	인천
3	A0004	1500	2000	여	57	미혼	공무원	인천
4	A0005	1200	2004	여	39	미혼	공무원	경남
5	A0006	2500	2000	남	49	기혼	회사원	대전
6	A0007	2500	2005	남	56	기혼	무직	대전
7	A0008	3000	2010	남	17	미혼	공무원	충남

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1



7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

01. 휴의 하이브 에디터로 "스마트카 마스터 정보" 데이터셋을 조회해서 로컬 디스크에 저장한다. 휴의 하이브 에디터에서 다음의 QL을 실행한다. C://예제소스/bigdata2nd-master/CH07/HiveQL/그림-7.82.hql 경로에 실행할 하이브 QL이 있으니 복사해서 활용하도록 한다.

```
insert overwrite local directory '/home/pilot-pjt/mahout-data/clustering/input'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ' '
select
  car_number,
  case
      when (car_capacity < 2000) then '소형'
      when (car_capacity < 3000) then '중형'
      when (car_capacity < 4000) then '대형'
  end as car_capacity,
  case
      when ((2016-car_year) <= 4)  then 'NEW'
      when ((2016-car_year) <= 8)  then 'NORMAL'
      else 'OLD'
  end as car_year ,
  car_model,
  sex as owner_sex,
  floor (cast(age as int) * 0.1 ) * 10 as owner_age,
  marriage as owner_marriage,
  job as owner_job,
  region as owner_region
from smartcar_master
```

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

02. 군집분석을 하기 위한 "스마트카 사용자 마스터" 데이터셋이 정상적으로 만들어졌는지 확인한다. Server02에 SSH로 접속하고 다음 명령을 실행해 내용을 확인한다.

```
$ more /home/pilot-pjt/mahout-data/clustering/input/*
```

```
[root@server02 input]# more /home/pilot-pjt/mahout-data/clustering/input/*
A0001 중형 NORMAL D 여 60 미혼 공무원 서울
A0002 소형 OLD C 남 60 기혼 회사원 인천
A0003 대형 NORMAL E 남 20 미혼 무직 인천
A0004 소형 OLD H 여 50 미혼 공무원 인천
A0005 소형 OLD H 여 30 미혼 공무원 경남
A0006 중형 OLD G 남 40 기혼 회사원 대전
A0007 중형 OLD F 남 50 기혼 무직 대전
A0008 대형 NORMAL G 남 10 미혼 공무원 충남
A0009 소형 OLD B 여 60 기혼 회사원 경남
A0010 대형 NORMAL G 여 20 미혼 회사원 세종
A0011 대형 OLD E 여 60 기혼 학생 울산
A0012 대형 OLD D 남 40 기혼 자영업 대전
A0013 중형 OLD B 여 50 미혼 회사원 대전
A0014 소형 OLD B 여 20 기혼 학생 서울
A0015 대형 NORMAL H 여 60 미혼 무직 경기
```

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

03. 머하웃의 Canopy 분석의 입력 데이터로 사용하기 위해 HDFS 상에 /pilot-pjt/mahout/clustering/input/ 경로를 생성하고 앞서 생성한 "스마트카 사용자 마스터" 데이터인 "000000_0" 파일의 이름을 "smartcar_master.txt"로 변경해 HDFS에 저장한다.

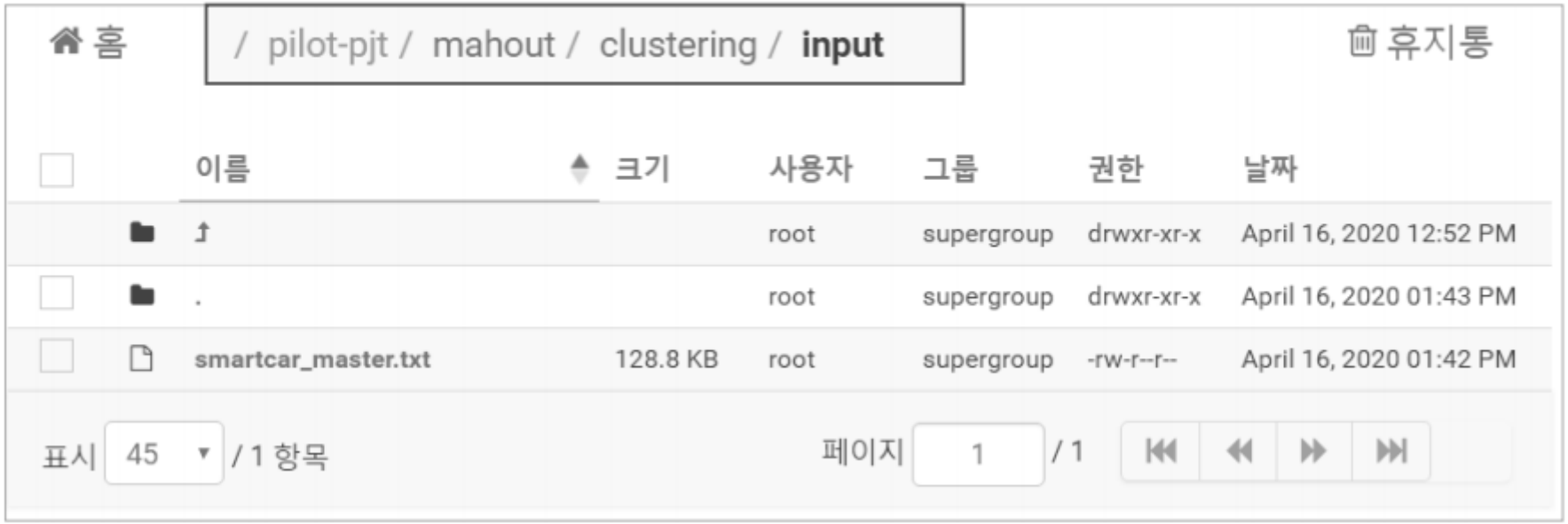
```
$ hdfs dfs -mkdir -p /pilot-pjt/mahout/clustering/input
$ cd /home/pilot-pjt/mahout-data/clustering/input
$ mv 000000_0 smartcar_master.txt

$ hdfs dfs -put smartcar_master.txt /pilot-pjt/mahout/clustering/input
```

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

04. 고객 마스터를 군집분석하기 위한 데이터가 HDFS에 정상적으로 적재됐는지 휴의 파일 브라우저로 확인해 보자.



7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

05. 머하웃의 Canopy 분석을 하기 위해서는 원천 파일이 시퀀스 파일이어야 한다. HDFS에 적재한 고객 마스터 데이터인 "smartcar_master.txt" 파일은 텍스트 파일 형식이므로 시퀀스 파일로 변환한다. 시퀀스 파일은 키/값 형식의 바이너리 데이터셋으로 분산 환경에서 성능과 용량 면에서 효율성을 높인 데이터 포맷이다. 이번 군집 분석에서는 시퀀스 파일의 키를 차량 번호로 하고, 나머지 사용자 마스터(차량연도, 차량 용량, 모델, 나이, 연령 등)를 값으로 구성하기 위해 간단한 시퀀스 파일 변환 프로그램인 "com.wikibook.bigdata.smartcar.mahout.TextToSequence"를 이용한다. 소스 프로그램은 C://예제소스/bigdata2nd-master/workplace/bigdata.smartcar.mahout에 있으니 참고한다. TextToSequence 프로그램을 실행하기 위해 사전에 빌드해 놓은 bigdata.smartcar.mahout-1.0.jar 파일을 Server02의 /home/pilot-pjt/mahout-data에 업로드한다.

- FTP 클라이언트인 파일질라를 실행해 Server02에 접속
- 머하웃 작업 경로: /home/pilot-pjt/mahout-data
- C://예제소스/bigdata2nd-master/CH07/bigdata.smartcar.mahout-1.0.jar 파일을 /home/pilot-pjt/mahout-data에 업로드

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

06. 텍스트 형식의 "스마트카 사용자 마스터" 파일을 시퀀스 파일로 변환한다. 변환 대상은 앞서 HDFS에 저장해둔 /pilot-pjt/mahout/clustering/input/smartcar_master.txt 파일이고, 변환 결과는 HDFS의 /pilot-pjt/mahout/clustering/output/seq에 생성된다.

```
$ hadoop jar /home/pilot-pjt/mahout-data/bigdata.smartcar.mahout-1.0.jar com.wikibook.bigdata.smartcar.mahout.TextToSequence /pilot-pjt/mahout/clustering/input/smartcar_master.txt /pilot-pjt/mahout/clustering/output/seq
```

```
Map-Reduce Framework
        Map input records=2600
        Map output records=2600
        Input split bytes=139
        Spilled Records=0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=45
        CPU time spent (ms)=1790
        Physical memory (bytes) snapshot=165478400
        Virtual memory (bytes) snapshot=1555099648
        Total committed heap usage (bytes)=116916224
File Input Format Counters
        Bytes Read=119122
File Output Format Counters
        Bytes Written=141360
```

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

07. 변환된 시퀀스 파일을 휴의 파일 브라우저를 통해 확인해 보자. 해당 경로에 part-m-00000이라는 이름의 시퀀스 파일이 생성됐을 것이다. 파일을 클릭해도 바이너리 형식으로 내용은 확인할 수 없다.

/ pilot-pjt / mahout / clustering / output / seq / **part-m-00000**

```
0000000: 53 45 51 06 19 6f 72 67 2e 61 70 61 63 68 65 2e  SEQ..org.apache.
0000010: 68 61 64 6f 6f 70 2e 69 6f 2e 54 65 78 74 19 6f  hadoop.io.Text.o
0000020: 72 67 2e 61 70 61 63 68 65 2e 68 61 64 6f 6f 70  rg.apache.hadoop
0000030: 2e 69 6f 2e 54 65 78 74 00 00 00 00 00 00 6c 40  .io.Text......l@
0000040: 8f ee 8b 41 95 73 8c 07 b9 b2 e4 bf 20 3a 00 00  ...A.s...... :..
0000050: 00 2f 00 00 00 06 05 41 30 30 30 31 28 32 30 30  ./.....A0001(200
0000060: 30 20 32 30 30 38 20 ec 97 ac 20 36 38 20 eb af  0 2008 ... 68 ..
0000070: b8 ed 98 bc 20 ea b3 b5 eb ac b4 ec 9b 90 20 ec  .... ......... .
0000080: 84 9c ec 9a b8 00 00 00 2f 00 00 00 06 05 41 30  ......../.....A0
0000090: 30 30 32 28 31 35 30 30 20 32 30 30 37 20 eb 82  002(1500 2007 ..
00000a0: a8 20 36 36 20 ea b8 b0 ed 98 bc 20 ed 9a 8c ec  . 66 ...... ....
00000b0: 82 ac ec 9b 90 20 ec 9d b8 ec b2 9c 00 00 00 2c  ..... .........,
```

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

참고로 시퀀스 파일의 내용을 확인하기 위해 다음의 HDFS 명령을 이용할 수 있다.

```
$ hdfs dfs -text /pilot-pjt/mahout/clustering/output/seq/part-m-00000
```

```
Z0080   중형 NORMAL A 여 50 미혼 자영업 경북
Z0081   중형 OLD A 여 50 미혼 학생 대구
Z0082   중형 OLD G 여 10 미혼 자영업 세종
Z0083   소형 NEW A 여 20 기혼 학생 서울
Z0084   소형 NEW A 남 60 미혼 무직 충북
Z0085   중형 NEW C 여 20 기혼 학생 광주
Z0086   대형 OLD F 남 30 기혼 공무원 제주
Z0087   소형 NEW F 여 60 기혼 무직 대전
Z0088   중형 OLD C 남 20 기혼 회사원 서울
Z0089   소형 NORMAL C 여 60 미혼 회사원 인천
Z0090   소형 NORMAL H 여 10 기혼 무직 제주
Z0091   중형 NORMAL E 남 40 기혼 자영업 제주
Z0092   대형 OLD G 남 10 기혼 무직 부산
Z0093   소형 OLD F 남 40 미혼 학생 전남
Z0094   소형 NORMAL F 여 30 기혼 학생 충남
Z0095   대형 NEW G 남 10 기혼 무직 충남
Z0096   대형 NEW C 여 60 기혼 회사원 경남
Z0097   대형 OLD E 여 60 미혼 회사원 전남
Z0098   소형 NORMAL A 여 30 기혼 회사원 세종
Z0099   중형 NORMAL C 남 20 기혼 무직 강원
Z0100   소형 OLD G 여 20 기혼 회사원 경기
```

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

08. 시퀀스 파일로 변환된 스마트카 마스터 데이터를 확인했으면 해당 시퀀스 파일을 로우별(차량번호)로 n-gram 기반의 TF(Term Frequency) 가중치가 반영된 벡터 데이터로 변환한다. n-gram의 벡터 모델은 단어의 분류와 빈도 수를 측정하는 알고리즘 정도로 이해하자. 여기서는 차량번호별 각 항목의 단어를 분리해 벡터화하기 위해 사용하겠다. 다음 명령을 실행해 스마트카 마스터 데이터를 다차원의 공간 벡터로 변환해 HDFS의 /pilot-pjt/mahout/clustering/output/vec에 생성한다.

```
$ mahout seq2sparse -i /pilot-pjt/mahout/clustering/output/seq -o /pilot-pjt/mahout/
clustering/output/vec -wt tf -s 5 -md 3 -ng 2 -x 85 --namedVector
```

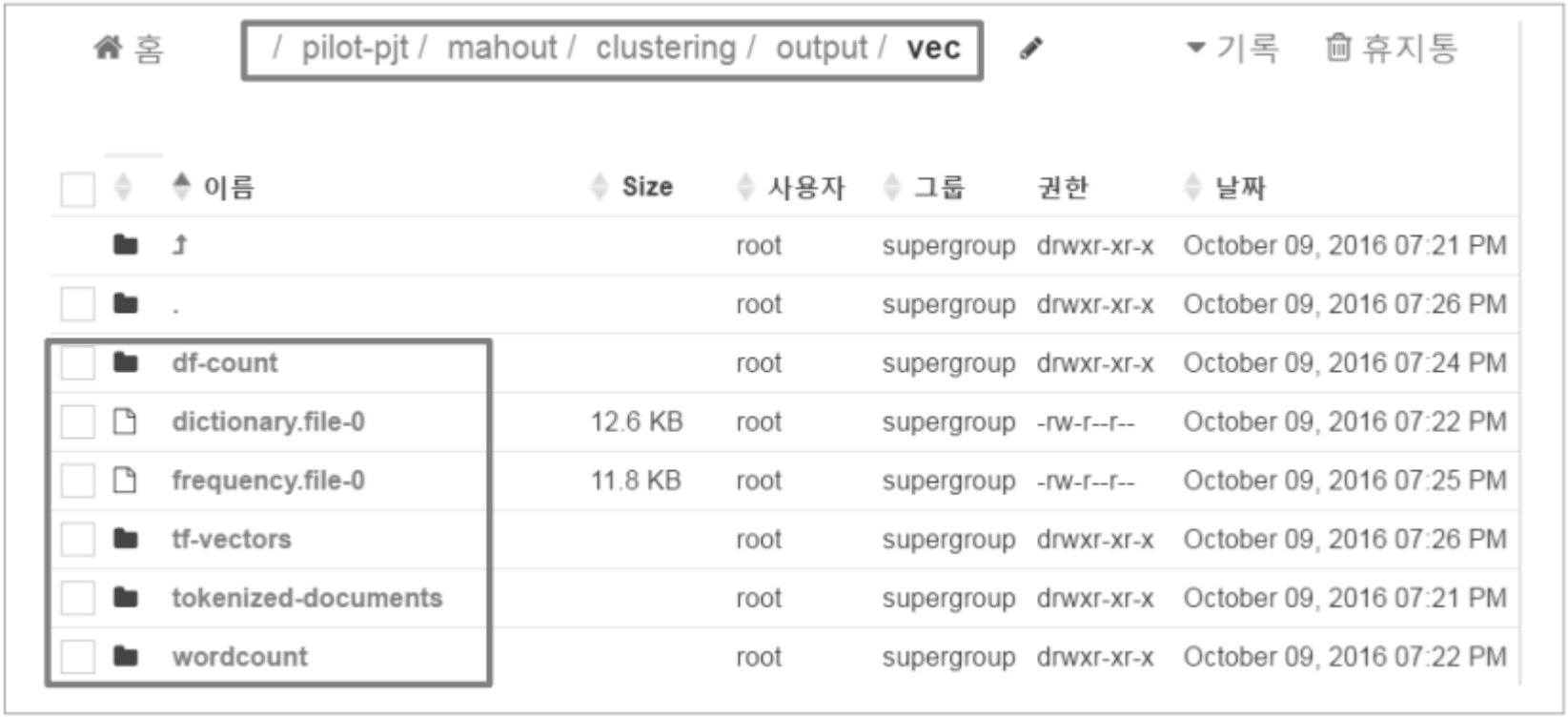
적용된 옵션에 대한 설명이다.

- **wt:** 단어 빈도 가중치 방식
- **md:** 최소 문서 출현 횟수
- **ng:** ngrams 최댓값
- **namedVector:** 네임벡터 데이터 생성

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

맵리듀스 잡이 여러 차례 실행되고 벡터 데이터가 정상적으로 만들어지면 휴의 파일 브라우저를 통해 pilot-pjt/mahout/clustering/output/vec 경로에 df-count, df-vectors 등의 디렉터리와 파일이 생성돼 있음을 확인할 수 있다.



이름	Size	사용자	그룹	권한	날짜
↑		root	supergroup	drwxr-xr-x	October 09, 2016 07:21 PM
.		root	supergroup	drwxr-xr-x	October 09, 2016 07:26 PM
df-count		root	supergroup	drwxr-xr-x	October 09, 2016 07:24 PM
dictionary.file-0	12.6 KB	root	supergroup	-rw-r--r--	October 09, 2016 07:22 PM
frequency.file-0	11.8 KB	root	supergroup	-rw-r--r--	October 09, 2016 07:25 PM
tf-vectors		root	supergroup	drwxr-xr-x	October 09, 2016 07:26 PM
tokenized-documents		root	supergroup	drwxr-xr-x	October 09, 2016 07:21 PM
wordcount		root	supergroup	drwxr-xr-x	October 09, 2016 07:22 PM

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

09. Canopy 군집분석으로 최적의 군집 개수를 파악하기 위해서는 센트로이드로부터 거리를 나타내는 t1, t2 옵션을 바꿔가며 반복적인 군집분석을 수행해야 한다. 다음과 같은 명령으로 첫 번째 Canopy 군집분석을 실행해 본다.

```
$ mahout canopy -i /pilot-pjt/mahout/clustering/output/vec/tf-vectors/ -o /
pilot-pjt/mahout/clustering/canopy/out -dm org.apache.mahout.common.distance.
SquaredEuclideanDistanceMeasure -t1 50 -t2 45 -ow
```

적용된 옵션에 대한 설명이다.

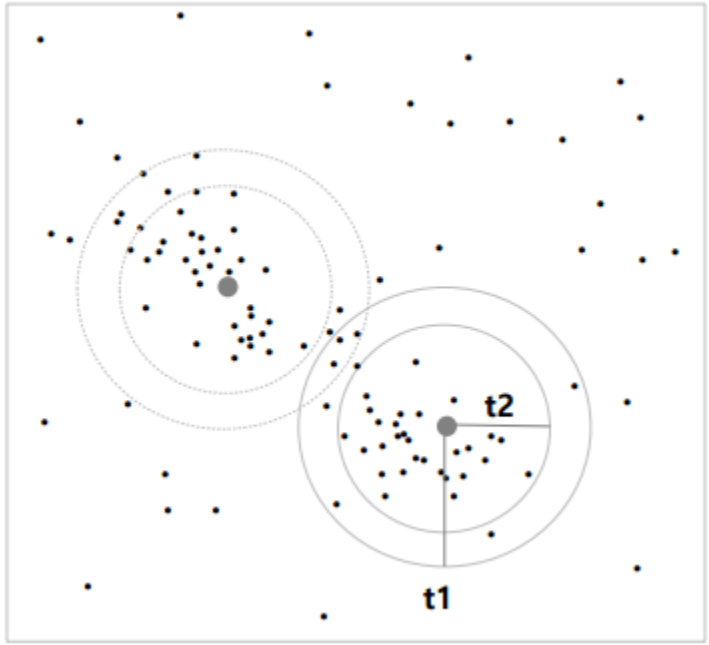
- i: 벡터 파일 경로
- o: 출력 결과 경로
- dm: 군집 거리 측정 알고리즘
- t1: 거리값 1
- t2: 거리값 2

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

Canopy 군집분석이 정상적으로 수행되면 HDFS의 /pilot-pjt/mahout/clustering/canopy/out/ 경로에 clusters-xx-final이라는 디렉터리가 만들어지고 그 안에 결과 파일이 생성돼 있을 것이다.

첫 번째 실행에서 "t1=50, t2=45"로 설정하고, 유사도 거리 측정을 위해 SquaredEuclideanDistanceMeasure를 사용했다. 참고로 Canopy 군집분석에서는 t1, t2의 길이가 그림 7.89처럼 "t1 〉 t2"이어야 하고, 중심점으로부터 "t2"의 반경 안에 있는 데이터는 해당 군집의 데이터로 확정되며, "t2"와 "t1" 사이의 데이터는 다른 군집 영역에 다시 포함되어 다른 군집의 데이터로 취급될 수 있다.



7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

10. Canopy 군집분석 결과를 다음 명령어로 확인한다.

```
$ mahout clusterdump -i /pilot-pjt/mahout/clustering/canopy/out/clusters-*-final
```

```
, 0.012, 0.014, 0.008, 0.009, 0.010, 0.010, 0.101, 0.103, 0.105, 0.099, 0.098, 0.010, 0.007
10, 0.007, 0.013, 0.009, 0.008, 0.008, 0.010, 0.010, 0.010, 0.009, 0.008, 0.008, 0.007, 0.0
.009, 0.006, 0.008, 0.012, 0.010, 0.008, 0.012, 0.010, 0.013, 0.010, 0.011, 0.011, 0.010, 0
 0.007, 0.006, 0.010, 0.008, 0.010, 0.008, 0.007, 0.010, 0.007, 0.011, 0.006, 0.012, 0.008,
0, 0.011, 0.012, 0.013, 0.015, 0.015, 0.011, 0.010, 0.008, 0.015, 0.018, 0.013, 0.013, 0.01
013, 0.010, 0.013, 0.008, 0.014, 0.007, 0.013, 0.011, 0.012, 0.009, 0.016, 0.012, 0.008, 0.
0.010, 0.015, 0.011, 0.010, 0.015, 0.014, 0.007, 0.015, 0.014, 0.011, 0.010, 0.013] r=[]}
16/10/09 21:01:42 INFO clustering.ClusterDumper: Wrote 1 clusters
16/10/09 21:01:42 INFO driver.MahoutDriver: Program took 4042 ms (Minutes: 0.06736666666666
```

그림 7.90 Canopy 군집분석 결과 1

그림 7.90에서 보면 Canopy 군집분석 결과 1개의 군집이 만들어졌다. 2,600명의 스마트카 사용자 마스터 데이터가 1개의 군집으로 분석됐다면 t1, t2의 각 초기 거리값을 너무 크게 잡은 것이다.

표 7.6 Canopy 군집분석 설정 1

t1	t2	클러스터 개수
50	45	1

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

11. t1, t2의 값을 각각 10, 8로 설정한 후 실행한다.

```
$ mahout canopy -i /pilot-pjt/mahout/clustering/output/vec/tf-vectors/ -o /
pilot-pjt/mahout/clustering/canopy/out -dm org.apache.mahout.common.distance.
SquaredEuclideanDistanceMeasure -t1 10 -t2 8 -ow
```

Canopy 군집분석에 대한 실행 결과를 다음 명령어로 확인해 보자.

```
$ mahout clusterdump -i /pilot-pjt/mahout/clustering/canopy/out/clusters-*-final
```

```
C-818{n=1 c=[11:1.000, 23:0.333, 25:0.667, 28:1.000, 37:1.000, 38:1.000, 40:0.333, 41:1.000, 44:0.333, 51:0.3
:0.333, 204:0.667, 235:1                clustering.ClusterDumper: Wrote 819 clusters                333] r=[]}
16/10/10 22:29:39 INFO
16/10/10 22:29:39 INFO driver.MahoutDriver: Program took 6921 ms (Minutes: 0.11535)
```

그림 7.91 Canopy 군집분석 결과 2

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

그림 7.91을 보면 이번에는 819개의 클러스터가 만들어진 것을 확인할 수 있다. 전체 스마트카 사용자 2,600명을 대상으로 819개의 군집으로 생성했다는 것은 t1, t2의 각 거리값을 너무 작게 잡았다고 볼 수 있다.

표 7.7 Canopy 군집분석 설정 2

t1	t2	클러스터 개수
50	45	1
10	8	819

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

12. 마지막으로 t1, t2의 값을 이번에는 각각 12, 10으로 설정하고 Canopy 군집분석을 실행한다.

```
$ mahout canopy -i /pilot-pjt/mahout/clustering/output/vec/tf-vectors/ -o /
pilot-pjt/mahout/clustering/canopy/out -dm org.apache.mahout.common.distance.
SquaredEuclideanDistanceMeasure -t1 12 -t2 10 -ow
```

Canopy 분석 실행이 끝나면 다음 명령어로 군집 결과를 확인해 본다.

```
$ mahout clusterdump -i /pilot-pjt/mahout/clustering/canopy/out/clusters-*-final
```

```
62, 33:0.332, 34:0.390, 36:0.089, 37:0.060, 38:0.019, 39:0.019, 40:0.102, 41:0.329, 42:0.070, 44:0.0
, 47:0.105, 49:0.322, 50:0.019, 51:0.278, 52:0.057, 53:0.044, 55:0.057, 57:0.291, 60:0.044, 65:0.392
67:0.056, 68:0.044, 70:0.057, 71:0.221, 73:0.348, 81:0.044, 83:0.263, 84:0.302, 85:0.044, 86:0.229,
:0.100, 92:0.130, 93:0.019, 95:0.070, 96:0.267, 98:0.056, 99:0.019, 100:0.130, 101:0.038, 102:0.019,
104:0.019, 105:0.019, 107:0.070, 108:0.038, 110:0.068, 112:0.019, 113:0.038, 114:0.067, 116:0.019, 1
2:0.302, 123:0.044, 124:0.044, 129:0.331, 130:0.056, 132:0.300, 135:0.044, 137:0.044, 140:0.089, 142
0.100, 146:0.044, 148:0.365, 149:0.441, 150:0.060, 151:0.051, 152:0.271, 154:0.189, 155:0.333, 156:0
249, 158:0.305, 159:0.057, 160:0.057, 161:0.291, 164:0.056, 175:0.019, 181:0.200, 182:0.130, 185:0.0
9, 189:0.044, 200:0.057, 206:0.057, 207:0.057, 208:0.067, 209:0.171, 214:0.019]}
16/10/12 01:49:14 INFO clustering.ClusterDumper: Wrote 148 clusters
16/10/12 01:49:14 INFO driver.MahoutDriver: Program took 6311 ms (Minutes: 0.10518333333333334)
```

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

그림 7.92를 보면 148개의 군집이 생성됐다. 2,600명의 고객을 대상으로 148개의 군집은 적절해 보인다. 적절하다는 기준은 다소 주관적일 수 있으나 분석 요건과 데이터의 성격에 따라 적절한 군집의 개수를 판단한다.

표 7.8 Canopy 군집분석 설정 3

t1	t2	Clusters 개수
50	45	1
10	8	819
12	10	148

7.7 분석 파일럿 실행 5단계 – 머하웃&스파크 ML

스마트카 고객정보 군집 분석1

실습