YouTube data analysis using MapReduce
The dataset is a reference of YouTube data.
Dataset Description:
Column 1: Video id of 11 characters.
Column 2: uploader of the video
Column 3: Interval between the day of establishment of YouTube and the date of uploading of the video.
Column 4: Category of the video.
Column 5: Length of the video.
Column 6: Number of views for the video.
Column 7: Rating on the video.
Column 8: Number of ratings given for the video
Column 9: Number of comments done on the videos.
Column 10: Related video ids with the uploaded video.

Problem Statements:
1) What are the top-5 categories with maximum number of videos uploaded?
2) Find the top-10 rated videos?
3) What is the most viewed video in the given dataset?

1) What are the top-5 categories with maximum number of videos uploaded?
Answer :
you_mapper.py

```
#!/usr/bin/python

import sys

for line in sys.stdin:
 splits = line.split('\t')
 if (len(splits) > 7):
  splits[3] = str(splits[3])
  print'{0},{1}'.format(splits[3],1)
```

you_reducer.py

```
#!/usr/bin/python

import sys
a_dict={}
counter = 0
for line in sys.stdin:
```

```
 category,value = line.split(',')
 category = str(category)
 value = int(value)
 if(counter == 0):
  a_dict[category] = value
  counter = counter + 1
 else:
  list_keys = [key for key in a_dict]
  if (category in list_keys):
   a_dict[category] = a_dict[category] + value
  else:
   a_dict[category] = value


'''for i in a_dict:
 print i, a_dict[i]'''

desc_order_list=sorted(a_dict,key=a_dict.get,reverse=True)

for i in range(0,5):
 print(desc_order_list[i],a_dict[desc_order_list[i]])
```

Command

```
cat youtubedata.txt | sort | ./you_mapper.py | ./you_reducer.py
```

Cloudxlab command

```
hadoop jar /usr/hdp/2.6.5.0-292/hadoop-mapreduce/hadoopstreaming.jar -input
/user/support1161/youtubedata.txt -output
/user/support1161/you_cloudxlab_output -file
/home/support1161/you_mapper.py -file /home/support1161/you_reducer.py
-mapper /home/support1161/you_mapper.py -reducer
/home/support1161/you_reducer.py
```

2) Find the top-10 rated videos?

Answer :

you_mapper.py

```
#!/usr/bin/python

import sys
```

```
for line in sys.stdin:
 splits = line.split('\t')
 if(len(splits) > 7):
  splits[0] = str(splits[0])
  splits[6] = float(splits[6])
  print'{0},{1}'.format(splits[0],splits[6])
```

you_reducer.py

```
#!/usr/bin/python

import sys
from collections import defaultdict


example_dict = defaultdict(list)

for line in sys.stdin:
 video,rating = line.split(',')
 video = str(video)
 rating = float(rating)
 example_dict[video].append(rating)


for a in example_dict:
 example_dict[a] = sum(example_dict[a])/float(len(example_dict[a]))

desc_order_list=sorted(example_dict,key=example_dict.get,reverse=True)

for i in range(0,10):
 print(desc_order_list[i],example_dict[desc_order_list[i]])
```

Command

```
cat youtubedata.txt | sort | ./you_mapper.py | ./you_reducer.py
```

Cloudxlab command

```
hadoop jar /usr/hdp/2.6.5.0-292/hadoop-mapreduce/hadoopstreaming.jar -input
/user/support1161/youtubedata.txt -output
/user/support1161/you_cloudxlab_output -file
```

/home/support1161/you_mapper.py -file /home/support1161/you_reducer.py
-mapper /home/support1161/you_mapper.py -reducer
/home/support1161/you_reducer.py

3) What is the most viewed video in the given dataset?
Answer :
you_mapper.py

```
#!/usr/bin/python

import sys

for line in sys.stdin:
 splits = line.split('\t')
 if(len(splits) > 7):
  splits[0] = str(splits[0])
  splits[5] = int(splits[5])
  print'{0},{1}'.format(splits[0],splits[5])
```

you_reducer.py

```
#!/usr/bin/python

import sys
a_dict={}
counter = 0
for line in sys.stdin:
 video,views = line.split(',')
 video = str(video)
 views = int(views)
 if(counter == 0):
  a_dict[video] = views
  counter = counter + 1
 else:
  list_keys = [key for key in a_dict]
  if (video in list_keys):
   a_dict[video] = a_dict[category] + views
  else:
   a_dict[video] = views
```

```
'''for i in a_dict:
 print i, a_dict[i]'''

desc_order_list=sorted(a_dict,key=a_dict.get,reverse=True)

for i in range(0,1):
 print(desc_order_list[i],a_dict[desc_order_list[i]])
```

Command

```
cat youtubedata.txt | sort | ./you_mapper.py | ./you_reducer.py
```

Cloudxlab command

```
hadoop jar /usr/hdp/2.6.5.0-292/hadoop-mapreduce/hadoopstreaming.jar -input
/user/support1161/youtubedata.txt -output
/user/support1161/you_cloudxlab_output -file
/home/support1161/you_mapper.py -file /home/support1161/you_reducer.py
-mapper /home/support1161/you_mapper.py -reducer
/home/support1161/you_reducer.py
```