Case study on Titanic Data Set: Map reduce
The text document consists details of passengers who were onboard during titanic tragedy.
It contains 12 columns
Column 1: Passenger ID
Column 2: Survived (survived=0 & died=1)
Column 3: Pclass(In which class passenger was travelling)
Column 4: Name
Column 5: Sex
Column 6: Age
Column 7: SibSp
Column 8: Parch
Column 9: Ticket
Column 10: Fare
Column 11: Cabin
Column 12: Embarked

Problem Statements:
1) Average age of the people (both male and female) who died in the tragedy
2) How many people survived travelling class wise?

1) Average age of the people (both male and female) who died in the tragedy
Answer :
titanic_*mapper.py*

```
#!/usr/bin/python
import sys

for line in sys.stdin:
 splits=line.split(',') #splitting each record
 if len(splits) > 6:   #check whether each record has minimum of 7 columns
  if int(splits[1]) == 1: #checking the condition for passengers who died
   if len(splits[5]):  #check whether if some value is present or not in age column
    print '{0},{1}'.format(splits[4],float(splits[5])) #print the gender and age of who died in the
tragedy
```

titanic_*reducer.py*

```
#!/usr/bin/python
import sys
count_f=0 #initialize count of female passengers to 0
```

```
age_f=0 #initialize sum of ages of females to 0
count_m=0 #initialize count of male passengers to 0
age_m=0 #initialize sum of ages of males to 0

for line in sys.stdin:
 gender,age = line.split(',')
 if gender[0]=='f':
  age_f = age_f + float(age) #if the passenger was female add her age
  count_f = count_f+ 1 #increment the count
 if gender[0]=='m':
  age_m= age_m + float(age) #if the passenger was male add his age
  count_m = count_m + 1 #increment the count

print('female',float(age_f/count_f)) #print average age of females
print('male',float(age_m/count_m)) #print average age of males
```

Command

```
cat TitanicData.txt | sort | ./titanic_mapper.py | ./titanic_reducer.py
```

Cloudxlab command

```
hadoop jar /usr/hdp/2.6.5.0-292/hadoop-mapreduce/hadoopstreaming.jar -input
/user/support1161/TitanicData.txt -output
/user/support1161/titanic_cloudxlab_output -file
/home/support1161/titanic_mapper.py -file /home/support1161/titanic_reducer.py
-mapper /home/support1161/titanic_mapper.py -reducer
/home/support1161/titanic_reducer.py
```

2) How many people survived travelling class wise?

Answer :

titanic_*mapper.py*

```
#!/usr/bin/python

import sys

for line in sys.stdin:
 splits=line.split(',') #splitting each record
 if len(splits) > 6: #check whether each record has minimum of 7 columns
  if int(splits[1]) == 0: #check whether the passenger survived or died
   print '{0},{1}'.format(int(splits[2]),1) #print class and 1 for each passenger who survived
```

titanic_*reducer.py*

```
#!/usr/bin/python
import sys
counter=0
pclass_dict={} #empty dictionary to add elements in the form of key value pairs

for line in sys.stdin:
 pclass,count=line.split(',') #take key as passenger class and value as count
 if(counter==0):   #to add first key value pair in the dictionary
  pclass_dict[pclass]=int(count)
  counter=counter+1
 else:
  nh=[key for key in pclass_dict]  #check whether the key already exists or not
  if(pclass in nh):
   pclass_dict[pclass]=pclass_dict[pclass]+int(count) #if exists then add the count to see how many people of that class survived
  else:
    pclass_dict[pclass]=int(count) #if they doesnot exist add the key value pair
print(pclass_dict)
```

Command

```
cat TitanicData.txt | sort | ./titanic_mapper.py | ./titanic_reducer.py
```

Cloudxlab command

```
hadoop jar /usr/hdp/2.6.5.0-292/hadoop-mapreduce/hadoopstreaming.jar -input
/user/support1161/TitanicData.txt -output
/user/support1161/titanic_cloudxlab_output -file
/home/support1161/titanic_mapper.py -file /home/support1161/titanic_reducer.py
-mapper /home/support1161/titanic_mapper.py -reducer
/home/support1161/titanic_reducer.py
```