MapReduce Implementation

Ninjas

Agenda

1.Dataset Description

2.The problem And Solution

3.Data cleaning

4.Implementing MapReduce

5.MapReduce results

Dataset Describtion:

#	Column	Non-Null Count	Dtype
0	USMER	1048575 non-null	int64
1	MEDICAL_UNIT	1048575 non-null	int64
2	SEX	1048575 non-null	int64
3	PATIENT_TYPE	1048575 non-null	int64
4	DATE_DIED	1048575 non-null	object
5	INTUBED	1048575 non-null	int64
6	PNEUMONIA	1048575 non-null	int64
7	AGE	1048575 non-null	int64
8	PREGNANT	1048575 non-null	int64
9	DIABETES	1048575 non-null	int64
10	COPD	1048575 non-null	int64
11	ASTHMA	1048575 non-null	int64
12	INMSUPR	1048575 non-null	int64
13	HIPERTENSION	1048575 non-null	int64
14	OTHER_DISEASE	1048575 non-null	int64
15	CARDIOVASCULAR	1048575 non-null	int64
16	OBESITY	1048575 non-null	int64
17	RENAL_CHRONIC	1048575 non-null	int64
18	TOBACCO	1048575 non-null	int64
19	CLASIFFICATION_FINAL	1048575 non-null	int64
20	ICU	1048575 non-null	int64

COVID-19 Dataset

This dataset contains an enormous number of anonymized patient-related information including pre-conditions.

The raw dataset consists of 21 unique features and 1,048,576 unique patients.
They are :

The problem And Solution

- Problem:
  to find out (number of deaths)

- Solution:
  use the 'DATE_DIED' column

Data cleaning

```
# drop missing data
covid.drop(covid[(covid['USMER']==97) |(covid['USMER']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['MEDICAL_UNIT']==97) |(covid['MEDICAL_UNIT']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['SEX']==97) |(covid['SEX']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['PATIENT_TYPE']==97) |(covid['PATIENT_TYPE']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['INTUBED']==97) |(covid['INTUBED']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['PNEUMONIA']==97) |(covid['PNEUMONIA']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['PREGNANT']==97) |(covid['PREGNANT']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['DIABETES']==97) |(covid['DIABETES']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['COPD']==97) |(covid['COPD']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['ASTHMA']==97) |(covid['ASTHMA']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['HIPERTENSION']==97) |(covid['HIPERTENSION']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['INMSUPR']==97) |(covid['INMSUPR']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['CARDIOVASCULAR']==97) |(covid['CARDIOVASCULAR']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['OBESITY']==97) |(covid['OBESITY']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['RENAL_CHRONIC']==97) |(covid['RENAL_CHRONIC']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['TOBACCO']==97) |(covid['TOBACCO']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['CLASIFFICATION_FINAL']==97) |(covid['CLASIFFICATION_FINAL']== 99)].index, axis=0,inplace=True)
covid.drop(covid[(covid['ICU']==97) |(covid['ICU']== 99)].index, axis=0,inplace=True)
covid
```

Implementing MapReduce functions

```
np.savetxt(r'C:/Users/DELL/OneDrive/Desktop/Lina/Big data & AI bootcamp/file.txt', covid.DATE_DIED, fmt='%s')
```

Map reduce

```
%%file wordcount.py

from mrjob.job import MRJob

class MRdateCount(MRJob):


    def mapper(self, _, date):

        yield (date, 1)


    def reducer(self, key, values):
        yield (key, sum(values))

if __name__ == "__main__":
    MRdateCount.run()
```

```
Overwriting wordcount.py
```

MapReduce results

```
# run the code as a terminal command
!python wordcount.py file.txt
```

```
"29\/06\/2020"  299
"29\/07\/2020"  48
"29\/09\/2020"  1
"30\/01\/2020"  2
"30\/03\/2020"  17
"30\/03\/2021"  1
"30\/04\/2020"  141
"30\/04\/2021"  10
"30\/05\/2020"  235
"30\/06\/2020"  295
"30\/07\/2020"  64
"30\/08\/2020"  1
"30\/10\/2020"  1
"31\/01\/2021"  1
"31\/03\/2020"  23
"31\/05\/2020"  242
"31\/07\/2020"  43
"31\/08\/2020"  2
"9999-99-99"    53871
```

```
count_alive = len(covid[covid['DATE_DIED'].map(lambda x : x=='9999-99-99')])
count_dead = len(covid[covid['DATE_DIED'].map(lambda x : x!='9999-99-99')])
print(count_alive)
print(count_dead)
```

```
53871
24304
```

Thank you!