Analysis of US Road Accident Data using MapReduce

TASK 1:

ind the number of accidents occurring per hour that satisfy a set of conditions and display them in sorted fashion.

Mapper 1:

For every line in the standard input, the following operations are performed:

1) Read line in dict format using json.loads()

2) Use if condition to check if record satisfies conditions specified (refer to docs for conditions).

- If conditions are satisfied, obtain time of accident from "Start_time" key.
  Split on '.' to handling edge case's where time is present in format "Y-m-d Hr-min-sec.00000" to get time in required format
- Output using print in the following format: hour of accident, 1

Modules Used: sys, json, math, datetime

Reducer 1:

1) Read hour and accident count (i.e - 1) from standard input
2) Convert hour to int using try except block to handle value errors.
3) A dict is maintained of all hours the value of which is incremented each time accident at that hour is observer

4)This dict is sorted by key and output is printed in the format: hour, count

In-built Modules used: sys

TASK 2:

Find record count per city and state

Mapper 2:

1) Read the input Latitude, Longitude, Distance (D) from command line
2) For each line in standard input , read line as dict using json.loads()
3) For each record obtain the latitude and longitude
4) Calculate Euclidean distance of accident from the obtained latitude and longitude with that inputted.
5) If calculated distance is within D, create a payload in format specified in document and do a post to the url given to obtain city and state where accident occurred from the response.
6) Print the output in the format: state, city, 1

In-built modules used: sys, json, math

Reducer 2:

1) State, city and count are read in the same order from standard input .
2) A dict of dicts in format {state : {city : count…}} is maintained to keep track of state's and citie's count of the number of accidents happening.
3) Dict is check if record for state and city exists,
    a If it exists the count for state is incremented and updated .
    b Else new key value pair is added to dict with the a count of 1 for the state.
4) Loop thought the dict and print output in the format :

```
State
city, count
state, total count
```

5)Total count is obtained for each state by taking sum of count for each city values

In-built modules used: sys