Big Data Report - 1

MAPPER 1

The data is read one row at a time from the standard input(Terminal) using json.loads() and sys.stdin (get_input function). The values for the given attributes is checked for nan, if the record is nan the record is ignored and the next record is brought in to the program.If these attribute for record does not contain any nan values then the given conditions for Severity, Visibility, Precipitation, Sunrise_sunset attributes is checked. If it is found to be true then the Weather_Condition value is compared with given weather condition keywords. Upon finding a match for one of the keys of weather condition given, the keys of description given is checked to see if they are present in the Description value. If the record passes all the above conditions then the Start_Time attribute for the record is processed using strptime() of datetime module, split(), join() functions and also using the slice operations to obtain the hour of occurrence of the accident. The hour of the accident is then printed on to the standard output(Terminal) in the format (hour,1).

REDUCER 1

A dictionary with keys ranging from 0 to 23 and values as 0 is created.
The data is read from standard input using sys.stdin
The hour is extracted from the data read using strip() and split() functions. The appropriate hour in the dictionary is incremented by 1
All the non zero values of the dictionary is printed in the format(hour, count)

MAPPER 2

The latitude, longitude and the Euclidean distance is read from the command line arguments. The data is read one row at a time from the standard input(Terminal) using json.loads() and sys.stdin (get_input function). The value for attributes Start_Lat and Start_Lng is checked for nan, if the record has nan the record is ignored and the next record is brought in to the program. If the attributes of record does not contain any nan values then the Euclidean distance is calculated using the above attributes as one point and the latitude and longitude read from the command line arguments as another point. The calculated Euclidean distance using the given formula(implemented in the Euclidean function) is checked to see if it is greater than the distance read from the command line argument. If the calculated distance is lesser than the given distance then it calls the post function to obtain city and state of the occurrence of the accident. The post function uses the requests.post function to do POST request http://20.185.44.219:5000/ with latitude and longitude in json format as the payload and processes the json by the post response to return city and state as string. If the city and state are not none then it is printed in the format (state,city,1)

REDUCER 2

4 variables one each to keep track of the current city(current_city) ,current state(current_state), no of accidents in the current city(count), no of accidents in the current state(total_count) are assigned to empty string, empty sting, 0, 0 respectively. The data is read from standard input using sys.stdin. The state and city are extracted from the data read using the strip() and split() functions as state and city variables. The logic below is repeated for each line in the standard input. If the current_state is an empty string then state is printed. current_state, current_city, count, total_count are assigned to state, city, 1, 1 respectively. If state is not equal to current_state the following are printed in the mentioned format (current_city,count),(current_state, total_count), (state) each of them in a new line. Current_state, current_city, count, total_count are assigned to state, city, 1, 1 respectively. Else total count is incremented by one. If the current_city is same as city the (current_city,count) is printed in that format, current_city and count are assigned to city and 1 respectively. Else count is incremented by 1.After all the iterations (current_city,count), (current_state, total_count) are printed in a new line in that format.