Q: How map reduce can be applied to find the happiness quotient of a community of people. Explain with parameters involved and sample dataset.

Ans:

The dataset I have taken contains the below information

```
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Country          782 non-null    object
 1   Happiness rank   782 non-null    int64
 2   Happiness Score  782 non-null    float64
 3   GDP per capita   777 non-null    float64
 4   Social support   777 non-null    float64
 5   Healthy life     777 non-null    float64
 6   Freedom          777 non-null    float64
 7   Generosity       777 non-null    float64
 8   Corruption       775 non-null    float64
 9   Year             782 non-null    int64
```

Dataset Shape: (782, 10)

The dataset is attached in the github.

The MapReduce algorithm is a framework for processing and analyzing large datasets in a distributed manner. It consists of two main phases: the map phase and the reduce phase. Let's break down how the MapReduce algorithm can be applied to determine the happiness score using the provided dataset.

Step 1: Initialization

The dataset containing information about various countries and their attributes is loaded into the MapReduce framework. Each row in the dataset represents a country, and each column represents a different attribute such as GDP per capita, social support, healthy life expectancy, freedom, generosity and corruption.

Step 2: Map Phase

In the map phase, the dataset is transformed into intermediate key-value pairs. Each row in the dataset is processed independently by multiple mappers, which can run on different nodes in a distributed system. Here's how the map phase works:

a. Filtering

Before processing, the mapper filters out the 'Happiness Rank' and 'Happiness Score' columns since they are not needed for calculating the happiness score.

b. Key-Value Transformation

For each row in the dataset, the mapper extracts the country name as the key and calculates the happiness score based on the remaining attributes. This could involve a simple averaging of the values in each row or applying a specific formula to derive the happiness score.

c. Emitting Intermediate Key-Value Pairs

The mapper emits intermediate key-value pairs, where the key is the country name, and the value is the calculated happiness score.

d. Parallel Execution

Multiple mappers process different parts of the dataset simultaneously, distributing the workload across the cluster to achieve parallelism and faster processing.

Step 3: Shuffle and Sort

After the map phase, the intermediate key-value pairs are shuffled and sorted based on their keys. This ensures that all values associated with the same key are grouped together and ready for the reduce phase.

Step 4: Reduce Phase

In the reduce phase, the shuffled and sorted intermediate key-value pairs are processed to produce the final output. Each unique key (country name) is processed by one or more reducers, which can also run on different nodes in a distributed system. Here's how the reduce phase works:

a. Grouping by Key

The reducer receives a key along with a list of values associated with that key. In this case, the key is the country name, and the values are the happiness scores calculated by different mappers.

b. Aggregation

The reducer aggregates the happiness scores for each country by calculating the mean, median, or applying any other aggregation function based on the received values.

c. Emitting Final Key-Value Pairs

After aggregation, the reducer emits the final key-value pairs, where the key is the country name, and the value is the aggregated happiness score.

d. Writing to Output

The final key-value pairs are written to the output, which can be stored in a distributed file system or any other storage system.

Step 5: Output

The MapReduce framework produces the final output containing the happiness scores for all countries. This output can be further analyzed, visualized, or used for other purposes.

Conclusion

The MapReduce algorithm efficiently processes large datasets by dividing the computation into smaller tasks and distributing them across a cluster of nodes. By following the steps outlined above, we can determine the happiness score for each country in the dataset using the MapReduce paradigm, achieving scalability and fault tolerance in the process.