**Introduction to Big Data**

**Project Report**

Madhura Yerawar

Alfred Zane Rajan (ar2257@njit.edu)

**Oozie Workflow:**

**Algorithm:**

**Program for Flights On Schedule:**

We create one Job with a Mapper of Text Key and LongWritable Value output; and a Reducer with Text Key and DoubleWritable Value output.

**Mapper:**

* Split the comma separated records and store them in an Array of Strings.
* Use an If Condition to skip Header and empty records.
* The Unique Carrier code and the arrival delay are obtained from the ninth and the fifteenth fields respectively.
* NULL valued records are excuded.
* The carrier code prefixed with “All: ” is written as Key with Value 1 inorder to obtain Total Flight Count.
* **IF** the arrival delay is more than 10 minutes, the carrier code is again prefixed with “Delayed: ” and passed as Key with Value 1 inorder to obtain Delayed Flight Count.

**Reducer:**

* Initialize a new Hashmap<String, Long>
* Initialize 3 Min variables with maximum double value and 3 Max variables with 0s. Initialize Text classes for respective Carrier names.
* In reduce function, **IF** the Key contains “All: “, remove it to get Carrier Code and sum up all the values to get Sumtotal.
* Put the Carrier Code and Sumtotal as Key Value pair into the Hashmap.
* **ELSE** **IF** the Key contains “Delayed: “, remove it to get Carrier Code and sum up the values to get delayed Sum.
* Get the SumTotal for the Carrier Code from the Hashmap.
* Divide the Sum by the SumTotal to get the probability of Delayed Flights. Subtract this value from 1 to get the probability of flights being On Schedule.
* Compare the obtained probability with each of the current Min and Max values and modify the list accordingly
* Write the three maximum and minimum On Schedule with their respective carrier codes.

**Program for Taxi Times:**

We create one Job with a Mapper of Text Key and LongWritable Value output; and a Reducer with Text Key and DoubleWritable Value output.

**Mapper:**

* Split the comma separated records and store them in an Array of Strings.
* The Origin, Destination, Taxi In and Taxi Out values are obtained from the Seventeenth, Eighteenth, Twentieth and Twenty First fields respectively.
* Use an If Condition to skip Header and records with “NA”.
* Write the Destination and Taxi In time as Key Value output.
* Write the Origin and Taxi Out time as Key Value output.

**Reducer:**

* Initialize 3 Min variables with maximum double value and 3 Max variables with 0s. Initialize Text classes for respective Carrier names.
* In the reduce function sum up all the values to the total of the taxi time and also keep a count of the iterations.
* Find the average Taxi time by dividing the total by the count.
* Compare the obtained average with each of the current Min and Max values and modify the list accordingly
* Write the three maximum and minimum average Taxi times with their respective airport codes.
* **IF** any of the Min values show maximum double value or if any of the Max values show 0, then write output that no further Taxi times available in the given data.

**Program for Flight Cancellation:**

We create one Job with a Mapper of Text Key and LongWritable Value output; and a Reducer with Text Key and LongWritable Value output.

**Mapper:**

* Split the comma separated records and store them in an Array of Strings.
* Use an If Condition to skip Header, empty records and records with “NA”.
* The Cancellation code is obtained from the Twenty third field.
* Write the Cancellation code as Output Key with Value as 1.

**Reducer:**

* Initialize a Max variable to 0 and a Text class to store respective Cancellation Code.
* In the reduce function sum up all the Values to get the number of times the cancellation code in the Key appeared.
* If this Sum is greater than the current Max value set the current sum as the Max value and also its respective Cancellation Code.
* **IF** the final Max value is 0, then write output that no cancellation codes are available in the given input data.
* **ELSE** write the maximum Cancellation Code as output.