**Assumptions:**

1. Read the input data and look for ArrDelay or field index 14 only negative values. Filter out the 0/NA or +ve values
2. Filter out origin and destination (index 16 and 17 if either of them are empty.
3. These are the only 3 field used

**Build Command (maven) :**

1. Download the project and go to folder Swami-UnityChallenge
2. Run ***mvn clean package***
3. ***Using hadoop 2.6, Java 1.8***

**Code:**

1. **Main Driver:** com.unity.challenge.mapreduce.driver.MapreduceDriver
2. Will execute AirlineDelayAverageMR and then TopNAirlineDelayMR (please read forceExecution below)
3. **AirlineDelayAverageMR/ AverageCSVMapper** will filter out the following records from the input in the mapper. ArrDelay if it is ‘NA’ or 0 or >0 or origin or destination is blank
4. **AirlineDelayAverageMR/ AverageCSVReduce** will have 100 records per reducer file, using PriorityQueue
5. **TopNAirlineDelayMR** will run only if the no of reducer files are more than 1 and the –f is set to true

**Run**:

1. Go to HADOOP\_HOME:

2. run command:

***./bin/hadoop jar data-pipeline-challenge/Swami-UnityChallenge /target/unity-mr.jar com.unity.challenge.mapreduce.driver.MapreduceDriver -i input -o output -n 100***

1. Arguments

|  |  |  |  |
| --- | --- | --- | --- |
| **Argument** | **Description** | **Default Value /Required** | **Notes** |
| -i | input | Required | Input Local Folder |
| -o | outputfolder | Required | Output Local Folder |
| -n | Number of Top Records | 100 | 100/1000 |
| -f | forceExecution | true | False - Will execute the only when MR1 is successful and only 1 reducer output file is generated from MR1.  True- will execute MR2 even if MR1 generates 1 reducer part file |

**Test:**

1. Create test cases using mrunit and other test cases using TestNG
2. Also tested the output with data from 2006 and 2007.. The data I validated was matching