Code Review - Ankita-Ushang-Xia

Ayushi-Balaraj-Shikha

October 23, 2017

Code Review:

- Would be better if the code is divided into packages for readability.

- The last commit was after 2:00pm. There was a change in the jar files and the Makefile. However, it was a minor change.

```
Commit Id:   B9b9ae11bae35cd99fbb01f1ae434061ec52646d
```

- The code is well written and properly documented.

- It would have been better if they had consistency in both SingleHopFlights and FlightPrediction

Mapper and reducer for SinglehopFlights are

```
SingleHopFlights.SingleHopFlightsMapper.class
SingleHopFlights.SingleHopFlightsReducer.class
```

However, the mapper and reducer for job : FlightPrediction:

```
FlightPrediction.FlightPredictionMapper.class
FlightPrediction.FlightPredictionMapper.FlightPredictionReducer.class
```

This should have been

```
FlightPrediction.FlightPredictionMapper.class
FlightPrediction.FlightPredictionReducer.class
```

For the sake of consistency and standards.

- FlightPredictionReducer should have to be in FlightPrediction or in a separate file.

- Not sure why they have restricted the number of reducers

```
j.setNumReduceTasks(12);
```

And why they chose 12. No explanation is given in this regards.

If conditions inside IsRouteRequired in UtilityHelper could have been written this way

```
public static boolean isRouteRequired(String[] record, String[] input){
     if(record[4].equals(input[1]) == false     ||
   (record[0].equals(input[3]) && record[1].equals(input[4]))
             return false;

        if(record[0].equals(input[3]) || record[1].equals(input[4]))
          return true;

     return false;
   }
```

- The authors have designed their system in such a way they have to hardcode flight list to check for predictions. Perhaps changing to different model for prediction could have helped. And also not sufficient explanation is given as to why it was not possible to test for all combinations.

```
Ideally we should check all predictions, however due to the fact that this job is incomplete, meaning we
have to hard-code the flight-list that we want to test, it's nearly impossible to test all. The randomly
picked result gives limited insight.
```

- Value of K is defined in the README.md as the neighborhood. We assume this is from assignment 1,2,3.

```
Where `k` is the k to be passed for the K-neighborhood
```

- We were able to generate the results but we were unable to compare it with our results.