**A picture containing text

Description automatically generated**

**A02 – REPORT ON ADDITIONAL SPARK LIBRARIES.**

Keeping a personal vehicle in big cities is not an easy thing. One needs to worry about arranging money for vehicle costs, insurance, road tax, various toll charges and parking charges. The commuter has to face traffic and, the new vehicle keeps amplifying the pollution problem. Despite struggling with issues, the number of personal vehicles continues to rise in the city. The use of public transport can solve many issues and would be beneficial in protecting the environment. The public transport system has existed for many years but it’s still not a popular choice always.

Some of the factors that make public transport not an obvious choice could be:

- Lack of last miles connectivity

- Time taken for commuting

- Non-availability of service during the need of hours

- Bad user experience due to poor quality of coach

As part of the assignment, I am focusing on the travel timing aspect of public transport using the Dublin bus dataset. There are a defined number of bus stop in the city where a bus stops as per its schedule. The bus has a defined route that mostly aims to cover the maximum area possible to provide affordable service to the masses. A bus following a bus route/line needs to stop at all its stoppage before reaching to the destination, this increases travel time even for travelling short distances as bus don’t take the shortest path and sometimes the points don’t have direct road connectivity.

***The goal here is to find the shortest bus route between two bus stop (source and destination) and plan a new bus for that route using this recommendation.***

It may not be practically feasible to start bus from all bus top to another bus stop at all the times, but this can be considered for the popular area where people travel most.

I am proposing to use Spark Graph Frames here which is based on Spark SQL Data Frames. Graph frames support Java, Scala and Python API and support algorithms, queries, and motif findings.

The concept of the Graph frame is based on the graph which consists of edges and vertices. Graph frame consists of an edge data frame and vertex data frame. All vertexes should have a unique id, and all edges connect two vertexes called source and destination.

Dublin bus dataset row has some static column which is a candidate for the vertex of the graph. All bus stoppage has a unique id and can be used as vertex id. Bus stoppage does have positional value (latitude & longitude) as well which can be a property of vertex. The remaining columns Date, vehicle number, delay, congestion can be used as a property of the graph edges.

The steps involved in doing this task would be:

* Load the dataset and find all the bus station (stoppage) id which is an integer and their latitude and longitude. To do this first filter row where atStop ==1 and then group the row based on closerStopID and take the average of latitude and longitude value. Here doing average is necessary as the dataset has a slight difference in reading for latitude and longitude value for same bus stop id taken at different time. The obtained data frame will be used as a vertex data frame. Need to cache this data frame as it will be used later.
* For edge data frame, need to find the connection between the bus stops. For this need find bus stoppage per bus number per line and add an edge between consecutive bus stop vertex. This is like finding bus schedule which we did for 2nd question of the 1st assignment and call it as schedule data frame. After finding the schedule, need to form data frame which has column ‘src’ and ‘dst’ which is source and destination bus stop id. This can be achieved using lag function of spark sql.
  + Add new column as dest to the source schedule data frame and populate its value and distance from the last bus stop (as we did for 4th question of the 1st assignment).

Call this as edge data frame and cache it for later use.

* Form GraphFrame using vertex and edge dataframe and cache it. Now the Graphframe is ready, can-do querying and find out the shorted bus route path between two bus stop. Here the generated graph is directional and the bus stop where there is maximum incoming connection would be most popular destination for the travel and can be obtained by graph’s indegree value. Alternatively, grouping based on src & dst of graph edge and ordering in decreasing order will give the popular source and destination.

Now pick two bus station and find the shorted bus route between them doing breadth-first search. Obtained route needs to validate if there is direct bus between the station selected. A new bus route can be planned to make better user experience, and this can be done for most bus stop per hours windows (who will wait for more than an hr to travel at bus stop).

As next step, I would apply Spark ML lib to do prediction for bus delay which is famous exercise to do with transport dataset.