Tech used:
Kafka
Hadoop
Plotly

Queries:

RAJ

1. How many distinct buses just crossed/will cross very soon stop_id 572 in time 0:00 - 01:00(Can be found using lat,lng of buses in vehicle_feed and lat,lng of stops in stops table)
2. Report the buses closest(the one(s) at least distance) to Govind Puri stop with stop id 469 at 00:00:05 .

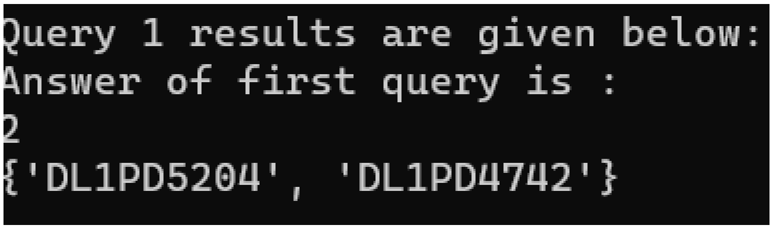
3. Print top 5 stops where most of the distinct vehicles just passed/will pass very soon in time 00:00- 01:00

Utkrisht

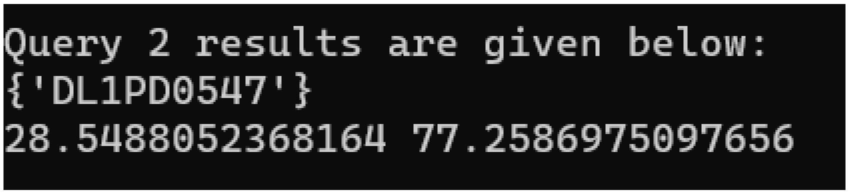
4. From 0:00 to 01:00, how many vehicles are within 1km,2km and 3km of their destination?(make a graph on XY plane where vehicles within 1km are shown with blue dot, 2km->green, 3km->red)

5. Calculate total fare as the stream comes in. (use fare_attributes and fare_rules to find them. Also plot line graph of fare value)

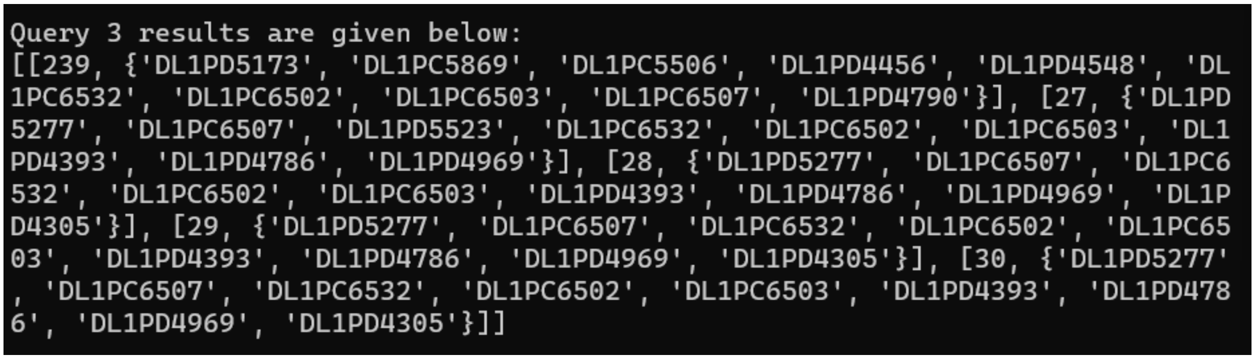
6. Report how many distinct vehicles are there whose first stop is 4054.



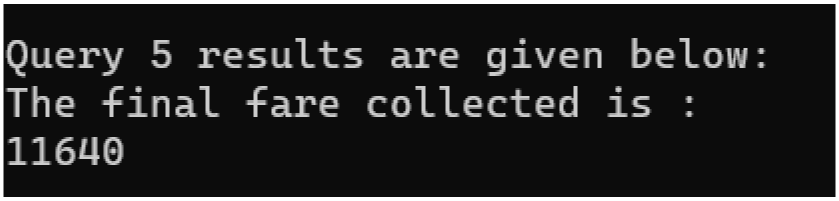
```
Query 1 results are given below:
Answer of first query is :
2
{'DL1PD5204', 'DL1PD4742'}
```



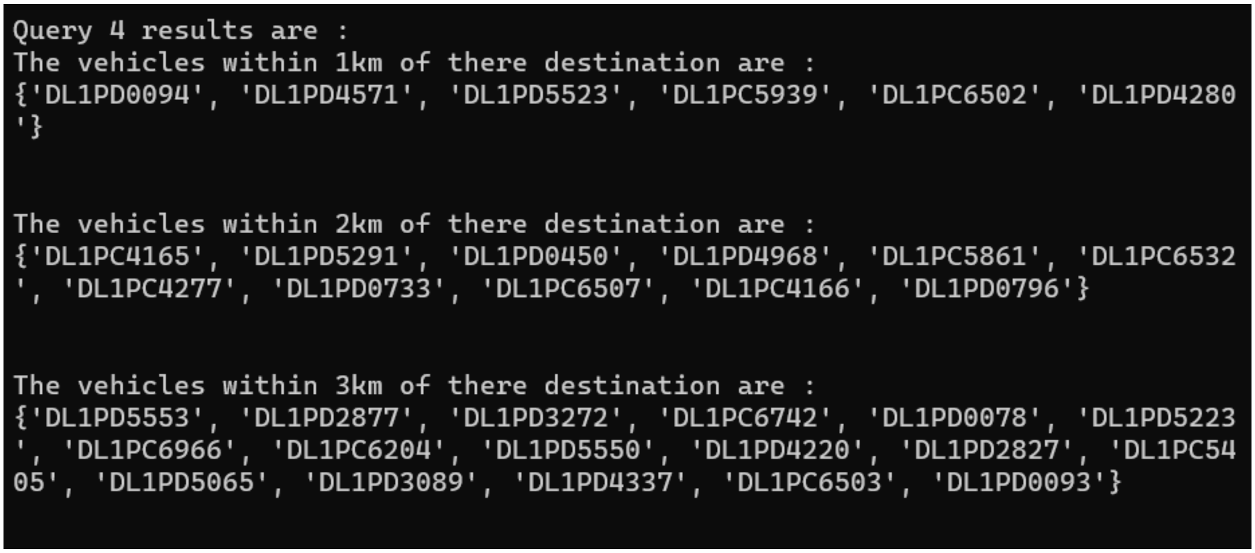
```
Query 2 results are given below:
{'DL1PD0547'}
28.5488052368164 77.2586975097656
```



```
Query 3 results are given below:
[[239, {'DL1PD5173', 'DL1PC5869', 'DL1PC5506', 'DL1PD4456', 'DL1PD4548', 'DL
1PC6532', 'DL1PC6502', 'DL1PC6503', 'DL1PC6507', 'DL1PD4790'}], [27, {'DL1PD
5277', 'DL1PC6507', 'DL1PD5523', 'DL1PC6532', 'DL1PC6502', 'DL1PC6503', 'DL1
PD4393', 'DL1PD4786', 'DL1PD4969'}], [28, {'DL1PD5277', 'DL1PC6507', 'DL1PC6
532', 'DL1PC6502', 'DL1PC6503', 'DL1PD4393', 'DL1PD4786', 'DL1PD4969', 'DL1P
D4305'}], [29, {'DL1PD5277', 'DL1PC6507', 'DL1PC6532', 'DL1PC6502', 'DL1PC65
03', 'DL1PD4393', 'DL1PD4786', 'DL1PD4969', 'DL1PD4305'}], [30, {'DL1PD5277'
, 'DL1PC6507', 'DL1PC6532', 'DL1PC6502', 'DL1PC6503', 'DL1PD4393', 'DL1PD478
6', 'DL1PD4969', 'DL1PD4305'}]]
```



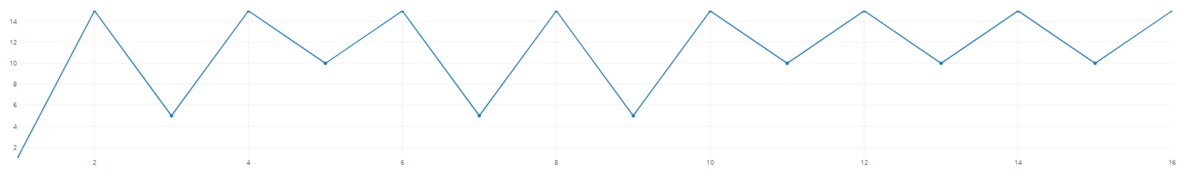
```
Query 5 results are given below:
The final fare collected is :
11640
```



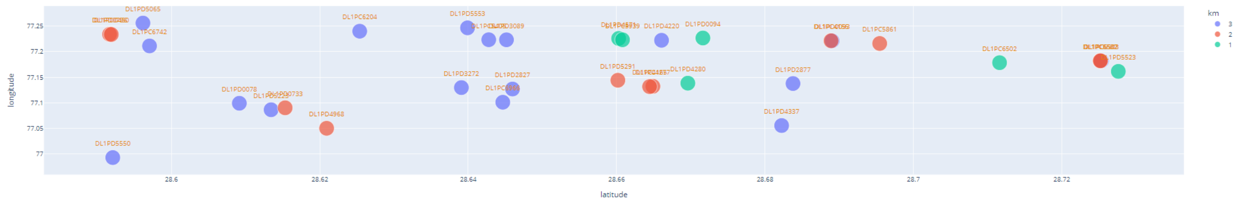
```
Query 4 results are :
The vehicles within 1km of there destination are :
{'DL1PD0094', 'DL1PD4571', 'DL1PD5523', 'DL1PC5939', 'DL1PC6502', 'DL1PD4280
'}


The vehicles within 2km of there destination are :
{'DL1PC4165', 'DL1PD5291', 'DL1PD0450', 'DL1PD4968', 'DL1PC5861', 'DL1PC6532
', 'DL1PC4277', 'DL1PD0733', 'DL1PC6507', 'DL1PC4166', 'DL1PD0796'}


The vehicles within 3km of there destination are :
{'DL1PD5553', 'DL1PD2877', 'DL1PD3272', 'DL1PC6742', 'DL1PD0078', 'DL1PD5223
', 'DL1PC6966', 'DL1PC6204', 'DL1PD5550', 'DL1PD4220', 'DL1PD2827', 'DL1PC54
05', 'DL1PD5065', 'DL1PD3089', 'DL1PD4337', 'DL1PC6503', 'DL1PD0093'}
```

WEB APP

Interactive graph of Query5. Y-axis is fare value. X-axis is corresponding element of the incoming stream. Rightmost value is latest



Interactive graph of Query4



Then, run preprocess_A3.py in spark(can be omitted if sav. files are already present)
Then, start the kafka server.
Then, consumer5.py in vagrant in a new terminal.
Then, visualizeq_final.py in D:\classroom\sem6\bda.
Then run producer5.py in vagrant in a new terminal.