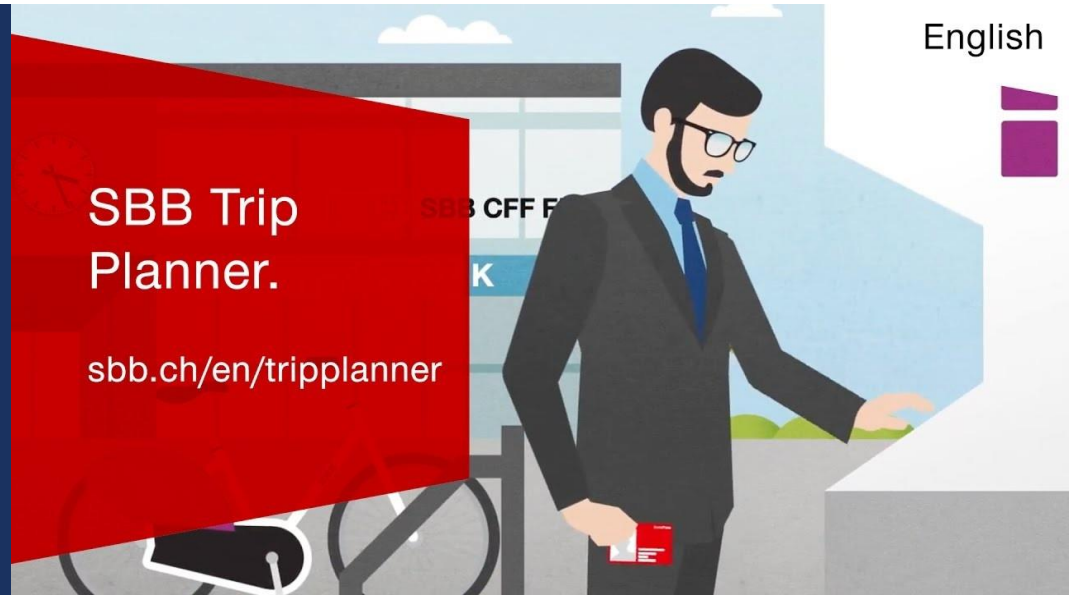
DS LAB - FINAL ASSIGNMENT: ROBUST JOURNEY PLANNING

LAB IN DATA SCIENCE - EE-490(H)

CLAUDIO LOUREIRO – CEM MUSLUOGLU – QIANQIAN QIAO - PEGGY WANG



SUMMARY

- Defining problem
- Modelization of our system
- Implementation and results
- Shortcomings and conclusion

DEFINING THE PROBLEM

- Build a public transport route planner given the SBB dataset
- Compute the fastest path Q% of the time
- Define an interface to visualize the paths

MODELIZATION

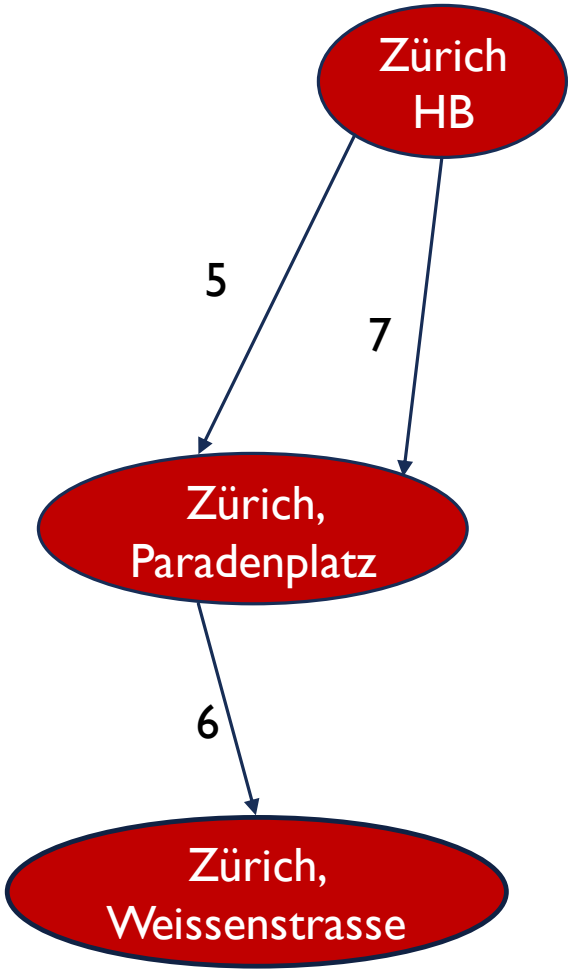
- Probability distribution modeling
- Generation of a graph
- Route planning
- Visualization

IMPLEMENTATION (1)

- Probability distribution modeling
  - Calculation of mean and standard deviation for trains/bus/trams over several days for arrival and departure times
  - Statistics for a given transportation at a given time and station
  - Assumption of a log normal law for the random variables representing the delay of arrivals
  - $T \sim Log - \mathcal{N}(\mu, \sigma^2)$
  - $\Pr(T < Scheduled\ departure\ time\ - Scheduled\ arrival\ time)$

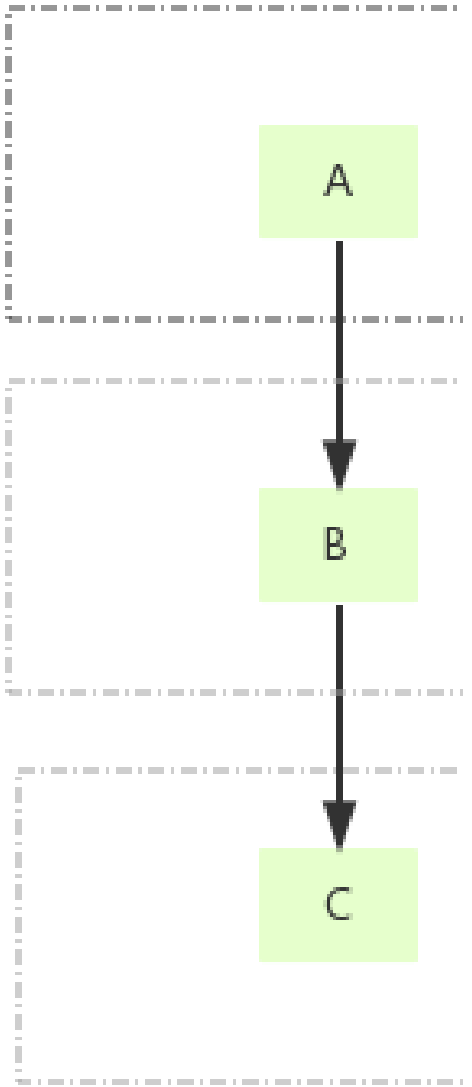
IMPLEMENTATION (2)

- Graph generation
  - A node is generated for every station in the range of 10km from Zürich
  - An edge is generated between nodes when there is a connection between these two stations by a transport
  - Edges are also generated between all the train stations



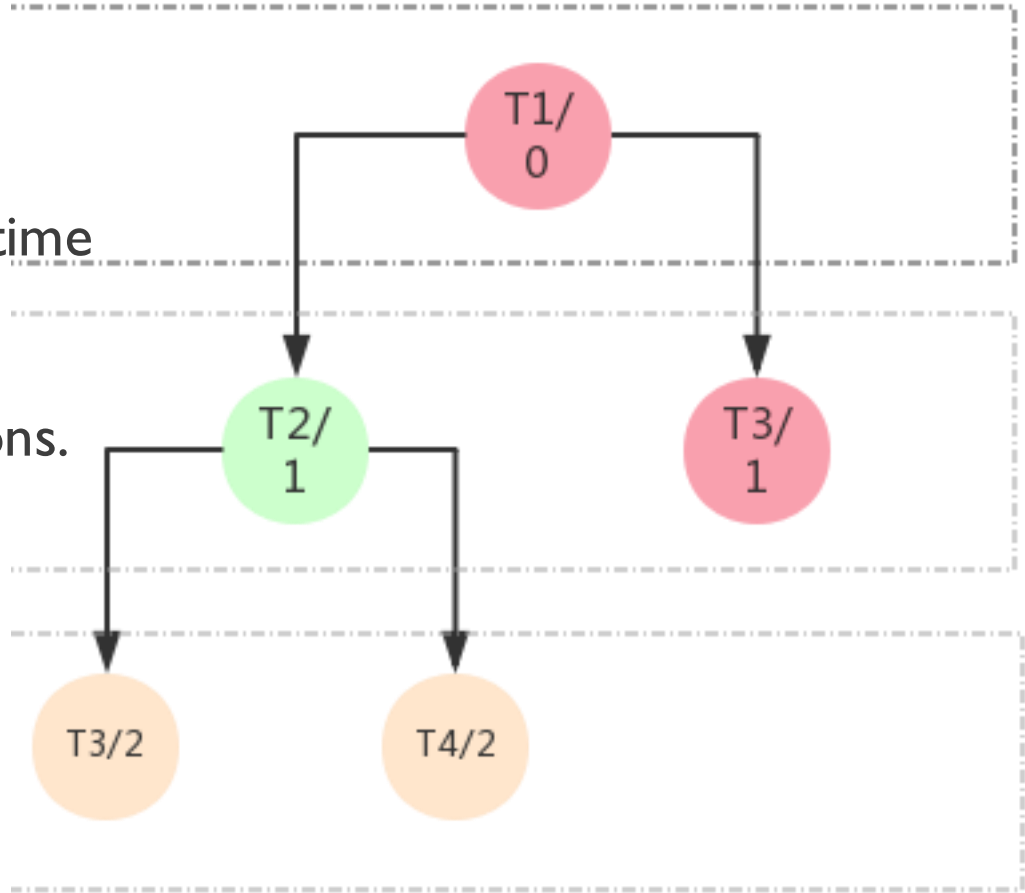
IMPLEMENTATION (3)

- Route planning
  - MultiDiGraph & DiGraph
  - DiGraph to get 1 / 2 shortest path of stations



IMPLEMENTATION (3)

- Route planning
  - Check all connections from MultiDiGraph
  - Get valid routes according to present and departure time
  - Get a new Graph_route using the time as nodes.
  - DFS to find the path from T1 to all possible destinations.



IMPLEMENTATION : UNCERTAINTY

- We assume that there is a 100% chance of arriving on time by walking
- If the we change a transportation in a same station (without walking) and the arrival of the current bus is the same as the departure of the next bus we compute the probability of being late no more than 30 seconds
- If we do not change transports, we obviously have a 100% of making the connection
- We assume that these random variables are independent
- $\prod_{i \in connections} p_i > \gamma_t, \gamma_t$ is the risk factor, then the path is valid

IMPLEMENTATION: VISUALISATION

- bokeh + Google map API
- Multi-ways to show:
  - Start point: red point
  - End point: yellow point
  - Different means of transportation：different color
  - Information: hovertools

- Example:

  From 'Zürich, Wollishofen’

  To 'Zürich, Museum Rietberg’

  From '05:10:00'

Station_name:	Transportation:	Arr_time:	Dep_time:
['Zürich, Wollishofen',	['walk',	['05:10:00',	[’05:10:00',
'Zürich, Butzenstrasse',	'walk',	'05:15:00',	'05:15:00',
'Zürich, Morgental',	'85:3849:107424-03005-1',	'05:20:00',	’05:21:00',
'Zürich, Post Wollishofen',	'85:3849:107424-03005-1',	'05:22:00',	'05:22:00',
'Zürich, Billoweg',	'85:3849:107424-03005-1',	'05:24:00',	'05:25:00',
'Zürich, Brunaustrasse',	'85:3849:107461-03005-1']	'05:26:00',	'05:41:00',
'Zürich, Museum Rietberg'],		'05:42:00'],	‘’],

IMPLEMENTATION: VISUALISATION

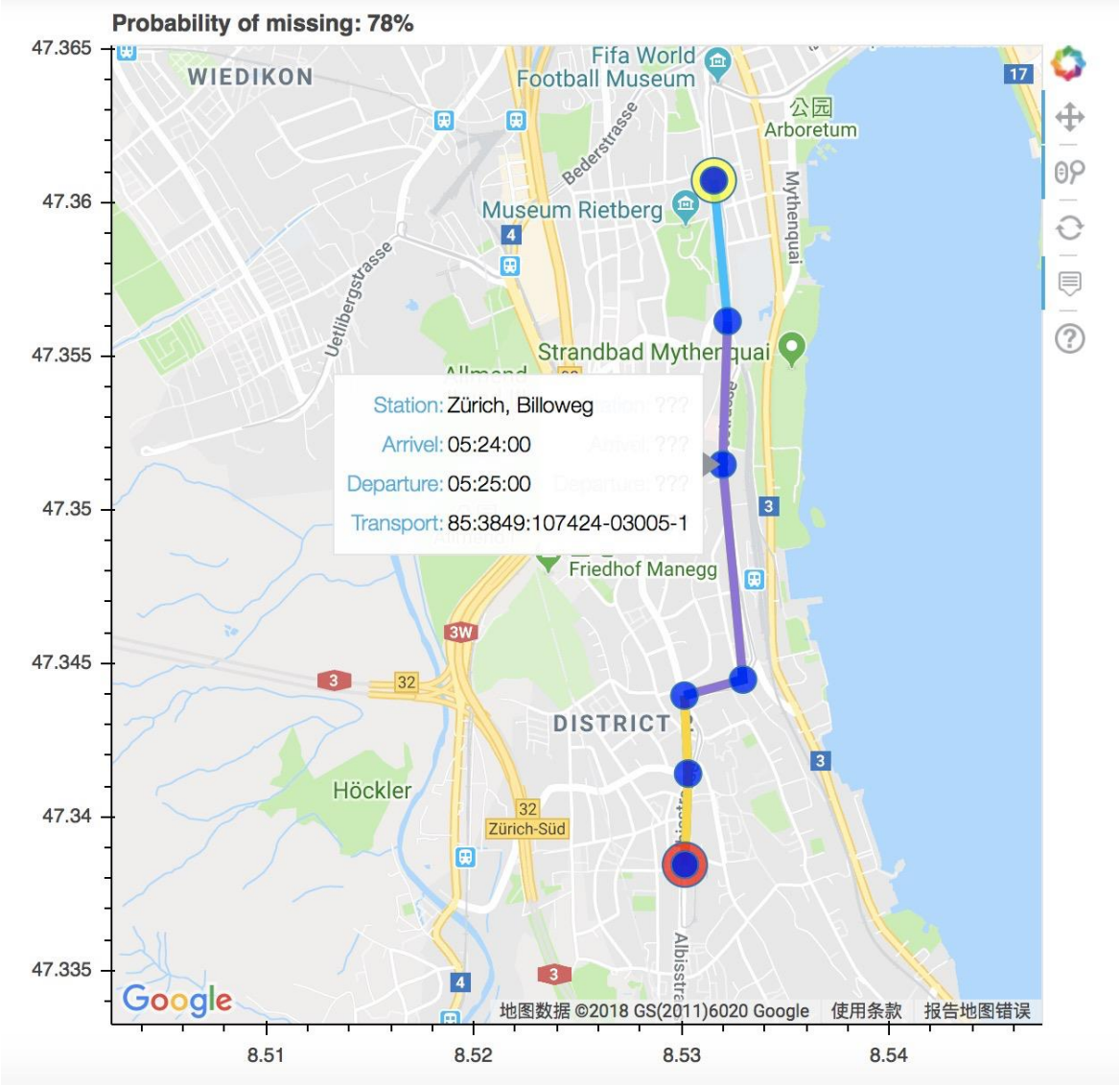
- bokeh + Google map API
- Multi-ways to show:
  - Start point: red point
  - End point: yellow point
  - Different means of transportation：different color
  - Information: hovertools

- Example:

  From 'Zürich, Wollishofen’

  To 'Zürich, Museum Rietberg’

  From '05:10:00'



SHORTCOMINGS

- Many assumptions on the probabilities
- Algorithm could be improved → high computation time
- In most cases departure and arrivals is the same when computing our path