Behavioral based route planner

Name: Emil Karlsson

Class: MU22

Background

I'm always in a rush, and I'm fast. So when using a route planner app I'll have to make multiple searches and puzzle it together to "my route". Route planner tends to return response where some routes are excluded because the app assumes I wouldn't make it. But I would. So my conclusion is that there's room for optimization not necessarily in the operator side of public transportation. Since there's physical limitations such as infrastructure etc. But to optimize/personalize routes in the end user side based on the end users historical and location data (lat, lon, time, speed).

Purpose

I believe that to achieve more streamlined public transportation the end user behavior must be in consideration. Make the route planning more personalized and aim to get the smoothest route based on end user Geo data and stuff. Build behavior profiles.

Target group:

1. Operators route end users to a specific route to reduce crowding and time spent in transit. It can serve as a compliment to surveys, to get better data about user behavior continuously which helps out with planning traffic.
2. End users could be target group to optimize their routes.

Goals

- Utilize location data to identify if end user takes a alternate route. ex: end user travels from T-Centralen to Slussen. Match user location with transportation location, speed. Decide what route and transport the user actually use to their destination.
- Analyze user location behavior in transit for route optimization.
- Investigate possibility of more individual end user routing.
- Target is to make the end user spend as little time as possible commuting but enough time.
- Find the most "effective" route based on geo-location, average speed, route history etc.
- Analyze if new route is more effective for the user. Ex: User moves in different phase so they have different opportunities in switches.
- Measure end users time to make a switch.
- Machine learning to individually predict more effective routes in future transit.

Plan

I decided to stay within SL. Because I want to use a system where changes, crowding and many different transport types are available.

1. Get API-keys.
2. Collect General Transit Feed Specification (GTFS) data.
3. Set up DevEnv.
4. Java back end to communicate with specified api:s and machine learning model.
5. Write ML script in python. Utilizing Round-Based Public Transit Routing Algorithm (RAPTOR) algorithm.
6. Write java backend.
7. Run in individual docker containers

Technologies

- Python. Machine Learning, Analyze geo-location data och traffic data. Jupyter-Notebook for ML model.
- Java. Backend to handle data between external API:s and Python ML model.
- Anaconda 3. Virtual Env, package management for ML.
- JSON.
- Git
- Docker. Run ML and Java backend in containers.
- GeoLocation, GeoPandas, sklearn, matplot.
- Spring Boot. WebServices, REST, JWT, Sockets.
- Kanban and docs. Jira, Confluence.
- API:s and data:
  SL Transport | Trafiklab
  SL Deviations | Trafiklab
  SL Nearby stops 2 | Trafiklab

Repo

lbrp/gtfs at main · pablo-chacon/lbrp