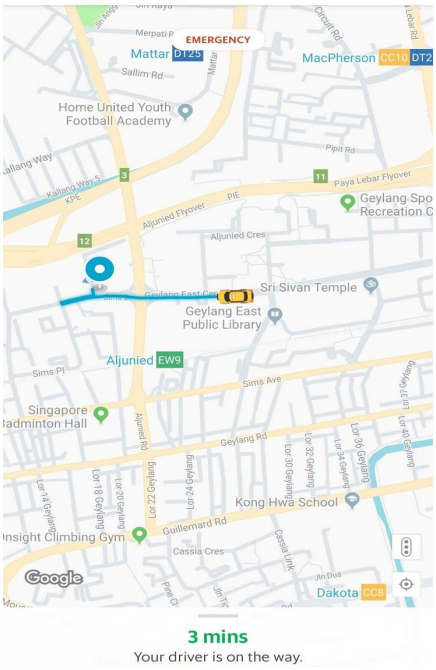
Microsoft Azure Virtual Hackathon
Problem Statement

Grab is Southeast Asia's leading ride-sharing company. Millions of passengers arrive at their destinations safely everyday via Grab's services. Estimated time of arrival (ETA) on the way towards pickup or destination plays a crucial role in order to provide excellent travel experience to our customers. An accurate prediction of travel times, reduces passenger anxiety before the ride helps them in managing their time better. Additionally, Grab can leverage accurate ETAs to maximize efficiency of allocation as well as make fair pricing for each trip.

In this hackathon, we would like to pose the problem of predicting estimated times of arrival given GPS trails from thousands of trips fulfilled by Grab. This can be achieved by developing a machine learning model either dependent on the route or otherwise.



The above figure shows ETA displayed on the Grab app during the pickup phase of a ride.

Problem Definition:

Given attributes of a trip represented by a tuple

- latitude_origin
- longitude_origin
- latitude_destination
- longitude_destination
- hour_of_day
- day_of_week

Build an ML model / algorithm to predict Expected Time of Arrival (ETA) in seconds.

Evaluation:

1. **PHASE 1:**
    - Ideation
    - Solution design
    - Architecture
    - Model / Algorithm

    For design and architecture, please consider feasible solutions for data collection pipeline, model training pipeline, model deployment and scalability.

    For model / algorithm, encourage to try data exploration, data pre-processing, feature engineering, experiments to help model / algorithm selection.

    No prototype creation / trained model will be required during this phase.

2. **PHASE 2:**

    The proposed model / algorithm will be evaluated by a hidden test dataset.

    You can choose testing your model on one of three categories of trip data,
    (1) Singapore - Car
    (2) Jakarta - Car
    (3) Jakarta - Motorcycle

    The inputs to the evaluated model will be
    - latitude_origin,
    - longitude_origin,
    - latitude_destination,
    - longitude_destination,

- hour_of_day,
- day_of_week

RMSE will be used as a metric to evaluate the error of prediction from actual travel time. A model with the lowest RMSE value is considered as performing the best.

Details of model submission and evaluation will be explained in phase 2.

Dataset:

Grab-Posisi [1], is the first GPS trajectory dataset of Southeast Asia consisting of GPS traces from both Singapore and Jakarta, Indonesia. The data were collected in April 2019 with a 1 second sampling rate, which is the highest amongst all the existing open source datasets. It also has richer contextual information, including the accuracy level, bearing, speed and labels trajectories by data acquisition source (Android or iOS phones) and driving mode (Car or Motorcycle). The dataset contains more than 88 million pings and covers more than 1 million kms. Experiments on the dataset demonstrate new challenges for various geographical applications. The dataset is of great value and a significant resource for the community to benchmark and revisit existing algorithms. Table 1 shows trajectory category, and Table 2 shows attributes of GPS Pings.

Table 1. Trajectory Category

City	Mode	Device	Total Trajectories
Singapore (SIN)	Car	iOS	14K
Singapore (SIN)	Car	Android	14K
Jakarta (JKT)	Car	iOS	14K
Jakarta (JKT)	Car	Android	14K
Jakarta (JKT)	Motorcycle	iOS	14K
Jakarta (JKT)	Motorcycle	Android	14K

Table 2. Attributes of GPS Pings

Attribute	Data Type	Remark/Format
Trajectory_ID	string	identifier for the trajectory
Latitude	float	WGS84
Longitude	float	WGS84
Timestamp	bigint	UTC
Accuracy Level	float	circle radius, in meter
Bearing	float	degrees relative to true north
Speed	float	in meters/second

Dataset link:
[1] https://engineering.grab.com/grab-posisi

How to Request for Dataset:

Grab-posisi is a GPS trajectory dataset Grab publishes to external parties for research usage. Dataset will only be shared upon request and review of the requestor. The data recipient should undertake the Non-Disclosure Agreement (NDA) and Data Processing Addendum (DPA) before the dataset can be shared.

To request for the data:

- Each team can send an Email request to Grab by a team representative.
- Please use Microsoft Azure Virtual Hackathon + representative's name as Email subject.
- Please indicate the representative's name and mobile phone number with country code in Email.
- Please sign Non-Disclosure Agreement (NDA) and Data Processing Addendum (DPA) and attach to Email.
- Please send request to grab.posisi@grabtaxi.com

For NDA, you will need to fill in the required information on page 1 and page 14, 19.1.(b) and then sign on page 16 at the bottom.

For DPA, you will need to replace the highlighted information on page 1 and the last page.

If you have any questions regarding the documents, please let us know. In addition, please confirm that the mobile number in your previous email is correct. We will share the dataset with you via securezapp. You should receive an email containing the download link to the dataset and you will be asked to enter a one-time password (OTP) sent to your mobile number. Thank you.

FAQ

1. **What are some Python packages that would be helpful for this challenge?**
    - For working with geospatial data: Geopandas, Shapely
    - For processing of data: QGIS
    - For analysis of road networks: OSMNX, NetworkX
    - For visualisation of GPS pings: KeplerGl

2. **Some pings have the value 0 for in the speed column - what does that mean?**

For iPhone devices, speeds = 0 means that it is invalid or the vehicle could be at a stop. For Android devices, speeds= 0 means that the vehicle is at a stop.

3. **The GPS pings are not matched, how can i process them?**

Some of the resources that perform map matching based on the Hidden Markov algorithm include GraphHopper or Barefoot. However, if you simply want to obtain the nearest edge to the GPS ping (might not be the most accurate method), this can be done using the 'get nearest edge' function in OSMNX or spatial join to the closest edge using GeoPandas or qGIS.

4. **How can I properly evaluate the model before submission?**

Method of evaluation on the participant's end would depend on the approach you have taken. If you have taken a direct machine learning approach, then a simple split into training and testing sets would suffice. Otherwise, if the element of time was considered when building your model, then you should set aside some days for testing to ensure no data has been leaked.

However, do note that whichever approach you have taken, the final evaluation method would be the same for all submissions, and you have to ensure that your model would be able to accept the data.

5. **What does the column 'trj_id' refer to?**

'trj_id' can be taken as a trip or booking ID, where each trip within each country would have a unique ID associated with it.

6. **How do I interpret the ‘pingtimestamp’ column?**

‘pingtimestamp’ refers to the UTC timestamp. Participants can choose to convert it into a readable time format using Python’s datetime package, or any other suitable methods.

7. **Can I use external data?**

Yes, external data is allowed as long as it is relevant to the problem, for example, weather data during the period.

8. **How can I do experiments?**

Based on the data provided, you may need to set up your own experiments to test models including ground-truth labelling, training / test data preparation, experiment analysis etc..