

Introduction to Machine Learning (67577)

Hackathon 2024 - Challenge 1:

HU.BER– Optimizing Public Transportation Routes


Teachers: Dr. Gabriel Stanovsky, Dr. Roy Schwartz

TAs: Eitan Wagner, Gili Lior, Amit Ben Artzy, Reshef Mintz



Tzars: Itai Alon, Qusay Muzaffar

July 2024

1 Motivation: Better Public Transportation

You've always wondered why you can wait for $\frac{1}{2}$ hour for the 9 bus from Rehavia, only to find that three of them arrive together one after the other, why some buses are packed while others are constantly empty, why getting to Mount Scopus can take a whole day, why the bus driver always closes the back door before the last passenger boarded, or in general whether [public transportation in Israel can be improved](#). After 8 weeks of the IML course, you decide that machine learning can help with (at least some) of these problems! So you start the  HU.BER company, which will use machine learning skills learned at the HU to improve various aspects of public transportation.

2 Dataset

To jump-start  HU.BER you search online, and find `train_bus_schedule.csv`, this seems to be a record of different bus routes and relevant features collected along their stops, containing 226,112 rows. Each row in the data was collected when a specific bus stopped at a specific station. For example, recording how many people boarded the 68 bus at Givat Ram station in a particular trip. You decide to use this data to train your first  HU.BER algorithms. Features description available here: [bus_column_description.md](#). As is often the case, some of the features are directly obvious (e.g., `door_closing_time`), while others are more cryptic (e.g., what's Menupach?), you can research online to try to better understand them and decide if they're useful for your purposes. To access the training and test sets, see Section 4.

3 Tasks

Please note that the tasks are independent. Partial submission will grant you a partial grade. Please read the submission guides carefully - if your output doesn't exactly match these definitions, we will not be able to grade it.

3.1 Predicting Passenger Boardings at Bus Stops

Given data for a single bus stop, your task is to predict the number of passengers boarding the bus at that stop.

Input.

A [csv](#) where each row holds information of a single bus stop within some bus route – all columns except *passengers_up*.

Output.

A csv file named “passengers_up_predictions.csv”, including two columns: *trip_id_unique_station* and *passengers_up*. An example of the output is provided in Table 1 and in [y_passengers_up_example.csv](#).

trip_id_unique_station	passengers_up
111a1	0
222b5	12
333c1	3

Table 1: An example of passenger boarding prediction output.

Evaluation.

We will evaluate your predictions according to their mean squared error (MSE) metric.

3.2 Predicting Trip Duration

Next, you want to predict how long is a single bus trip, from its first station to its last station. For this task, we treat each *trip_unique_id* as a single sample indicating a complete bus trip (see Figure 1). Based on all information of the stops in this trip, we want to predict the arrival time to the last stop.

Input.

A [csv](#) where each row holds information of a single bus stop within some bus trip. The test set excludes the *arrival_time* to all stops within the trip, and only provides that of the first station (i.e., the time the bus left its first stop). The predictions should include the trip duration in minutes.

Output.

A csv file named “trip_duration_predictions.csv”, including two columns: *trip_id_unique* and *trip_duration_in_minutes*. An example of the output is shown in Table 2 and in [y_trip_duration_example.csv](#).

trip_id_unique	trip_duration_in_minutes
111a	78.2
222c	122.0
333b	61.6

Table 2: Trip duration prediction output

Evaluation.

We will evaluate your predictions according to their mean squared error (MSE) metric.

3.3 Using your results to improve public transportation

Finally, you want to summarize the 🇮🇱 H.U.B.E.R conclusions from these two models.

First, outline several key conclusions from your data exploration. These could include insights such as identifying peak rush hours for public transportation, differences in bus usage between various regions of Israel, consistency in public transportation usage, and other interesting patterns or anomalies.

Next, propose practical ways to improve the public transit system. For instance, specify which bus lines should have increased frequency, which lines could have reduced frequency, and identify areas or routes where new bus lines could be introduced. Be as specific and practical as possible, leveraging the insights gained from your data analysis.

Remember—there is no correct answer for this task. We encourage you to be creative with your conclusions and ideas. Use the data insights to develop innovative solutions for enhancing the public transportation system in Israel.

The input The training data set.

The output A pdf file named “conclusions_and_suggestions.pdf”. Provide at most 2 written pages describing your conclusions and suggestions. Add between 3-4 figures to reinforce your claims (not included in the 2 pages of text).

4 Provided files

In the IML.hackathon.2024 github [repo](#) we provide the following files:

- Training data for all sub-tasks:
 - `train_bus_schedule.csv`
- Test sets:
 - Passengers boarding sub-task (3.1): `X_passengers_up.csv`
 - Trip duration sub-task (3.2): `X_trip_duration.csv`
- Example output files, provided for your convenience to ensure that your output is in the expected format.
 - Passengers boarding sub-task (3.1): running your trained model on `X_passengers_up.csv` should output a file with a similar format to `y_passengers_up_example.csv`.
 - Trip duration sub-task (3.2): running your trained model on `X_trip_duration.csv` should output a file with a similar format to `y_trip_duration_example.csv`.
- Evaluation scripts. We will run these scripts to test the quality of your predictions. Make sure that your output csv files are in the expected format for these scripts.
 - `eval_passengers_up.py`
 - `eval_trip_duration.py`

5 Tips

- When loading the train and test csv, and also when saving the predictions csv, use the ‘encoding="ISO-8859-8"’ option within `pd.load_csv` and `pd.to_csv`, to correctly parse the Hebrew fields.

- This data contains time-series events. Even though we haven't learned it in class, you might find it helpful to look up and explore how to approach time-series data.

Figure 1: How to figure out the trip duration for 3.2. Below is an example of a single trip, identified by a unique “trip_id_unique”. The trip duration is the difference between the arrival time to its first station (departure), to its last station.

trip_id	trip_id_unique_station	trip_id_unique	line_id	...	station_index	station_id	station_name	arrival_time	door_closing_time	...
111818	111818c1	111818c	11266	...	1	36667	בית רבקה	19:18:00	19:21:00	...
111818	111818c2	111818c	11266	...	2	32554	תיכון תלם/שפרינץ	19:22:00	19:22:00	...
111818	111818c3	111818c	11266	...	3	32556	שפרינצק / פינס	19:23:00	19:23:00	...
111818	111818c4	111818c	11266	...	4	32471	שמואל סלנט/יחיאל	19:24:17		...
111818	111818c5	111818c	11266	...	5	32469	שמואל סלנט/ביאלי	19:25:58		...
111818	111818c6	111818c	11266	...	6	32467	שמואל סלנט / אחז	19:28:00	19:28:00	...
...
111818	111818c25	111818c	11266	...	25	21828	מטה מרחב ירקון/	19:58:42		...
111818	111818c26	111818c	11266	...	26	21829	ראול ולנברג/הנחוש	19:59:46		...
111818	111818c27	111818c	11266	...	27	21830	ראול ולנברג/הארד	20:01:27		...
111818	111818c28	111818c	11266	...	28	28668	ראול ולנברג/דבור	20:02:00	20:02:00	...
111818	111818c29	111818c	11266	...	29	26807	עתידיס	20:03:00	20:03:00	...

(a) An example for a single trip from `train_bus_schedule.csv`. The total trip duration for the above trip is **45 minutes**, which is the difference between the **arrival time at the first stop** in the trip, and the **arrival time at the last stop**, for this specific trip (marked in red). We omit several rows (station index 7-24) and several columns (cluster, direction, etc.) from this trip only for readability.

trip_id_unique	...	trip_duration_in_minutes	...
111818c	...	45.0	...

(b) Representing trip 111818c for the duration prediction sub-task. The 29 rows from the original training set will be represented in a single row, providing the `trip_id_unique` and `trip_duration_in_minutes`, with any additional features you find useful for this task (such as line id, number of stations in trip, etc.)