

**دانشگاه صنعتی امیرکبیر**  
**( پلی تکنیک تهران )**

Machine Learning Course

Fall 2023

ML-PROJECT

Koorosh Khavari Moghaddam

[koorosh.k@aut.ac.ir](mailto:koorosh.k@aut.ac.ir)

[koorosh.khavari.m@gmail.com](mailto:koorosh.khavari.m@gmail.com)

Student ID: 401131023

## Contents

Model & Simulation .....	3
Problems And Tricks used in Simulation .....	3
Model Implementation and Logic.....	3
Subway Graph .....	3
Simulation Logic in Few Words .....	3
Detailed Explanation of Logic.....	3
Dataset Generation .....	5
Problem set .....	6
Estimation of entering passengers per stage .....	6
Estimation of leaving passengers per stage .....	6
Estimation of total passengers in subway.....	6
Estimation of distribution rate .....	7
Evaluation.....	7
Density Estimation Evaluation.....	7
Rate Estimation Evaluation .....	7

## Model & Simulation

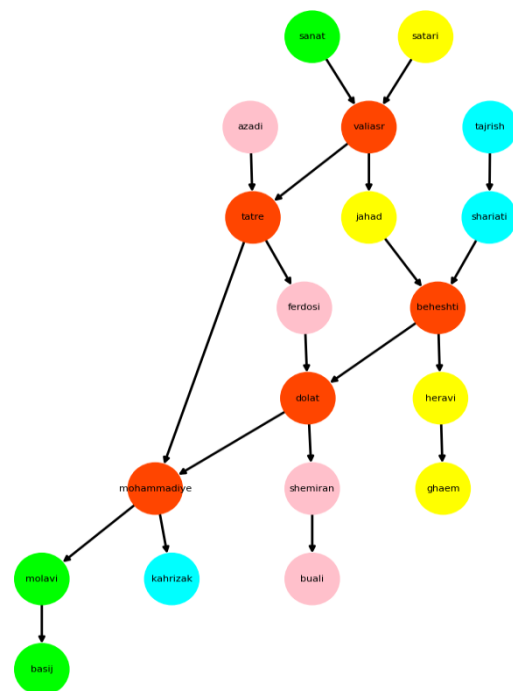
### Problems And Tricks used in Simulation

- Initially, passengers will enter the stations and wait for the train. Therefore, people will come and not exit at the **beginning of the simulation**.
- As suggested, we will handle **joint stations** in the following way: first, passengers will leave the station, and then half of the remaining ones will go in both directions.
- The **exit rate** will be calculated as the minimum value between the entrance-exit and the poisson\_exit. This ensures the correct flow through the stations based on the problem parameters.
- Do not **split people in half**! Instead, we will first disturb the halve values and round it to make sure the last person won't get halved.
- Everyone will leave the metro at the **last stations**. No one will stay there forever!
- There are fewer people in the station than the number of people who are going to exit. Let the **exit value** be the minimum between the people in the station and the exit count.
- If there is no train in a specific station, it means that specific station is not opened yet! People can't enter closed station; the **stations will get open** as soon as the first train reach to them.

### Model Implementation and Logic

#### Subway Graph

Based on the provided description of the subway, its structure gets implemented as a networkx graph as follows.



#### Simulation Logic in Few Words

Following implementation used a very intuitive approach to simulate the subway flow by using the graph incident matrix, given the rules we set for the simulation, we can simply simulate the flow by multiplying the nodes weights and the incident matrix together, resulting vector will be the updated state. Before each update, we inject passengers at appropriate stations and after each update, we will deport them!

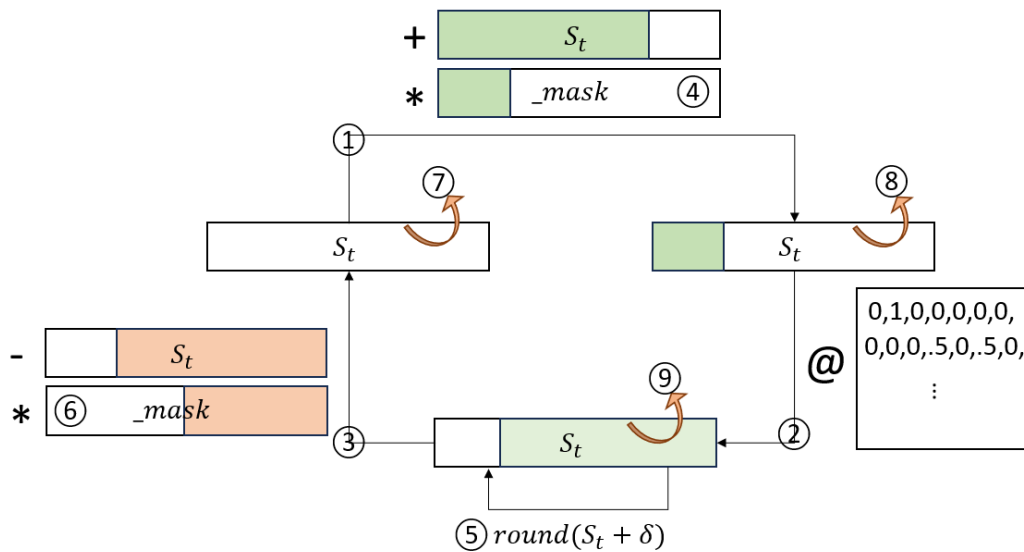
#### Detailed Explanation of Logic

In this section, we explained the simulation logic in detail by providing a simple description of logic behind state, state transformations, validators and logger.

##### State Vector

At any time, step  $t$ ,  $S_t$  will show the number of passengers in each station, it has the following order: ['tjrish', 'sanat', 'satari', 'azadi', 'valiasr', 'tatre', 'beheshti',

## State Transformations



①  $S_t \leftarrow enter(S_t, mask)$ : this function is used to inject passengers in the subway, the `_mask` parameter is a validator described in the validator section later in this text.

③  $S_t \leftarrow \text{departure}(S_t, \text{mask})$ : this function is used to deport passengers from the subway, the `_mask` parameter is a validator described in the validator section later in this text.

## Validators

④ enter `_mask`: as mentioned in the trick's sections, all the states are not going to be open at the same time, they are going to gradually get open as the first train and passengers reach to them. This natural progression is modeled using the depth variable which is simply defined as the max distance between each node and the set of the initial nodes, intuitively, this means the latest time one station can be opened. This mechanism is needed in order to handle the joint station properly, otherwise some complications will arise.

⑤ transformation disturbance: this operation is done in order to handle the transportation of odd number of the people. The incident matrix will naturally divide the population in joint stations in half, but if there are odd number of them, someone might end up being halved! Therefore by adding  $\pm 1$  to each

station afterward and round them up, we will transport that person from one station to other and correct the overall transportation.

⑥ exit \_mask: this validator exists with the same reason the enter one exists, to ensure natural flow of the closing of the station. As the night time reaches, the first stations gradually get closed. The first closing time for first stations (depth 0) is exactly 22:00-00:30, afterward the next depth stations will be closed gradually. This order is implemented using the exit \_mask variable.

### Loggers

As can be seen in the figure, we can log the process in any stage we like! We can capture the flow before, during or after of any transformation, inside the main code though, we log the entrance⑧, number of people inside⑨ and departure⑦ of the passengers. Number of passengers inside the subway can be logged either before entering or after exiting or after transportation, we used the last option to capture the number of people in each station given specific time after each transportation inside the subway.

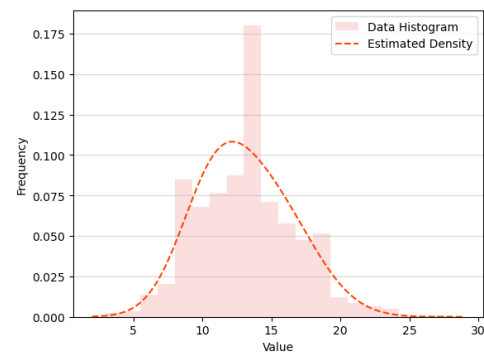
### Dataset Generation

This problem is handled using the function stage\_population\_sample inside the code. We can export desired dataset by providing appropriate data type argument for this function. This argument has the general form "[type]\_[transformation]\_by\_[location]", the "type" section can get either "re" for regression or "de" used in density estimation. "re" datatypes will have both input data and labels but the "de" datasets will only have input data. The "transformation" section can get either "enter", "exit" or "inside". Depending on this section, dataset will contain number of the entered people in each date and time, number of people who departure in each date and time or the number of people inside each station in each date and time. The third section "location" is simply the name of specific station, use the "total" to get all of the stations at once.

## Problem set

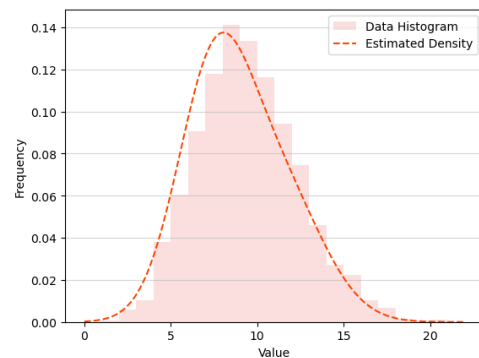
### Estimation of entering passengers per stage

We simply fitted a bimodal Gaussian mixture model to the distribution of the passengers entering per-stage. For example, the histogram of the entering passengers to the “molavi” station is shown in the plot. The best bimodal Gaussian distribution model pdf is showed using the dotted line.



### Estimation of leaving passengers per stage

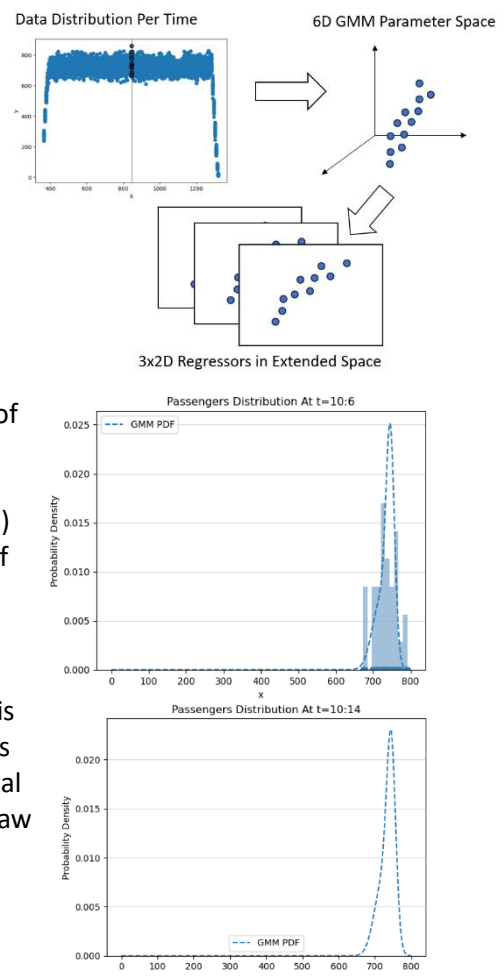
We simply fitted a bimodal Gaussian mixture model to the distribution of the passengers leaving per-stage. For example, the histogram of the leaving passengers from the “molavi” station is shown in the plot. The best bimodal Gaussian distribution model pdf is showed using the dotted line.



### Estimation of total passengers in subway

As can be seen in the shape, the scatter plot of the total passengers in the subway in each time (which must be a multiply of 6 min starting from 360) is a unimodal distribution. After fitting a GMM to each distribution for each time, we transformed the problem to the parameter space of the GMMs, after that by defining 3 polynomial regressors, we estimated each parameter of the GMMs (which are weights, means and covariances) we used these models to predict the parameters of the GMM governing the distribution at each time step.

For example, the estimated GMM for the total passengers in the whole subwas at time step 10:6 is shown. One interesting property of this model is its ability to generalize, for example it predicts the total passengers at time step 10:14 would be like the draw dotted line.



## Estimation of distribution rate

We used a GMM to fit a distribution to the histogram of the passengers, then the rate of the passengers would be simply the average of the modes of the distribution. This is not the best estimation but it is correct non the less.

For example, as shown below, the model predicts the rate of 37 for the “azadi” station passenger entrance, which is not exact but good enough.

rate=38.11077729101322

37.0

## Evaluation

### Density Estimation Evaluation

We used two density distance measure in order to calculate the difference between the “histogram” of the real value and estimated “pdf” of the estimated models. One of them is Wasserstein distance, which is independent of the domain of the distributions and measures the overall similarity of them, the other one is cross entropy distance.

### Rate Estimation Evaluation

We used norm 2 error in this case.