1. Description of the dataset and what's challenging about it

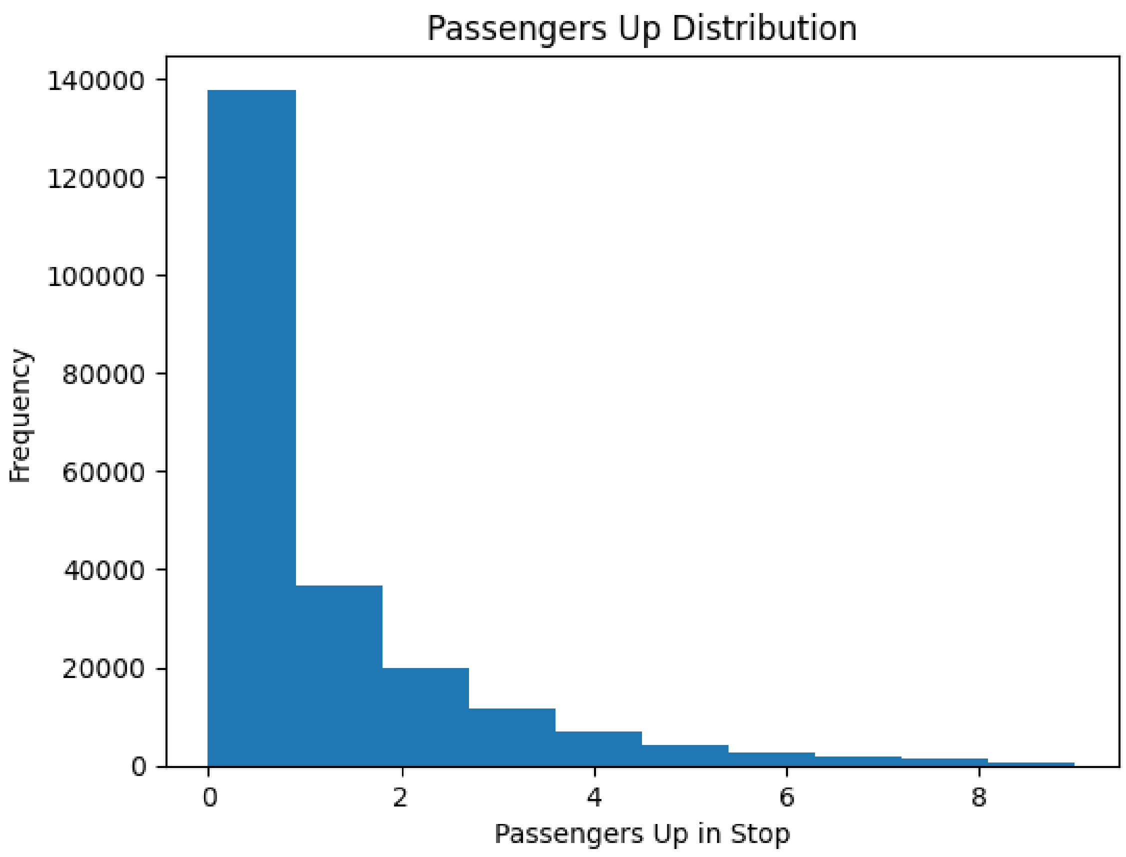
The dataset includes bus line trips across their route.

It includes identification information such as trip_id, line_id, station_id, etc

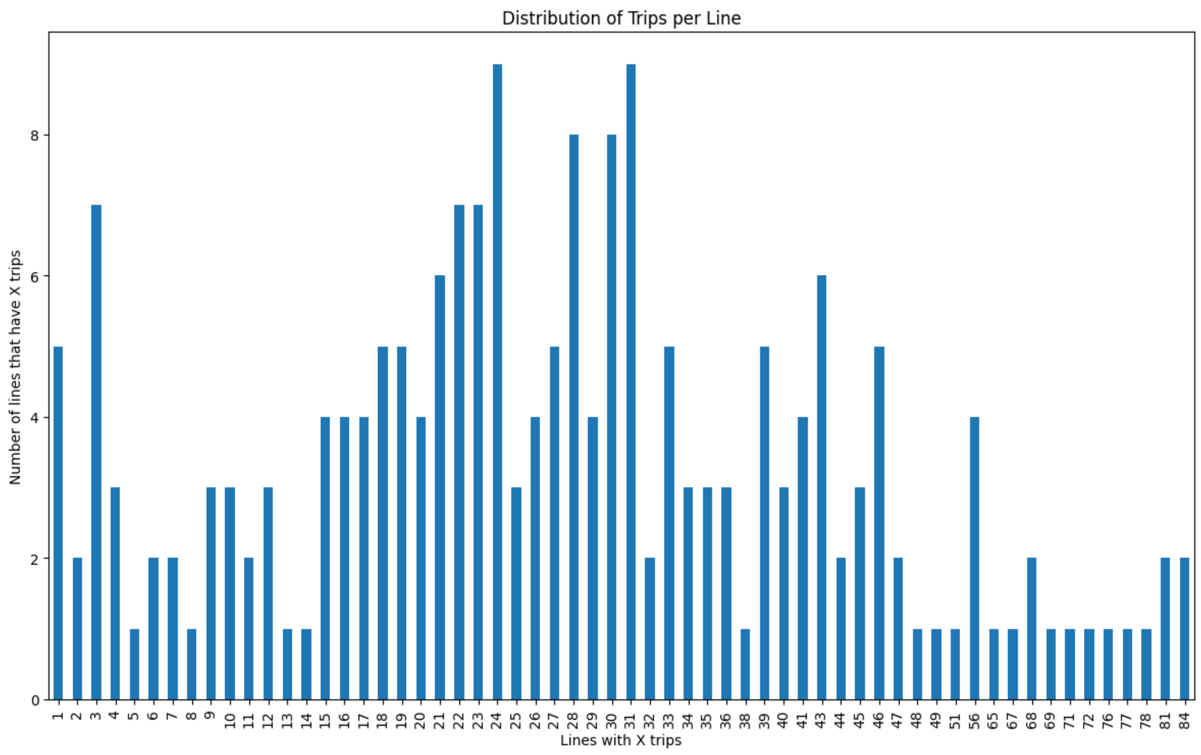
In addition, it has data we are trying to predict, such as time of arrival, and number of passengers that go on in the stop, log, lat and more.

There are a few difficult issues when predicting on the data:

- High skew on 0 values, which can cause the model to overfit and just predict ~0 all the time:



- Some lines are more represented than others, some lines will appear in 80 trips, while others in 1, which will give a bias on the lines that appear more:



In general, there we a lot of different features and combinations that could be correlated, and knowing which one is worth the time was hard to determine at first.

Also, some features were hard to understand and interpret, like nipuah, alternative and had to be researched by the values themselves sometimes, plus some data like x,y position is useless when raw, and needs to be processed heavily.

Data cleaning and preprocessing:

We cleaned the data from outlier trip durations.

In the *passengers_continue* column, if there were negative values, we deduced it's a human error and chose to put instead the abs value.

We filtered for outliers in the trip_duration that was greater than 2 standard deviation units from the median, and as well in passengers_continue .

we dropped the following columns:

mekadem_nipuach_continue, *mekadem_nipuach_luz* and all ID columns.

We believed that the *mekadem nipuch* was not useful enough to calculate the data, the nipuah_continue column was also highly correlated with continue column, thus not needed.

Features:

in the exploration of the model predicting passengers up, we decided to add the following columns:

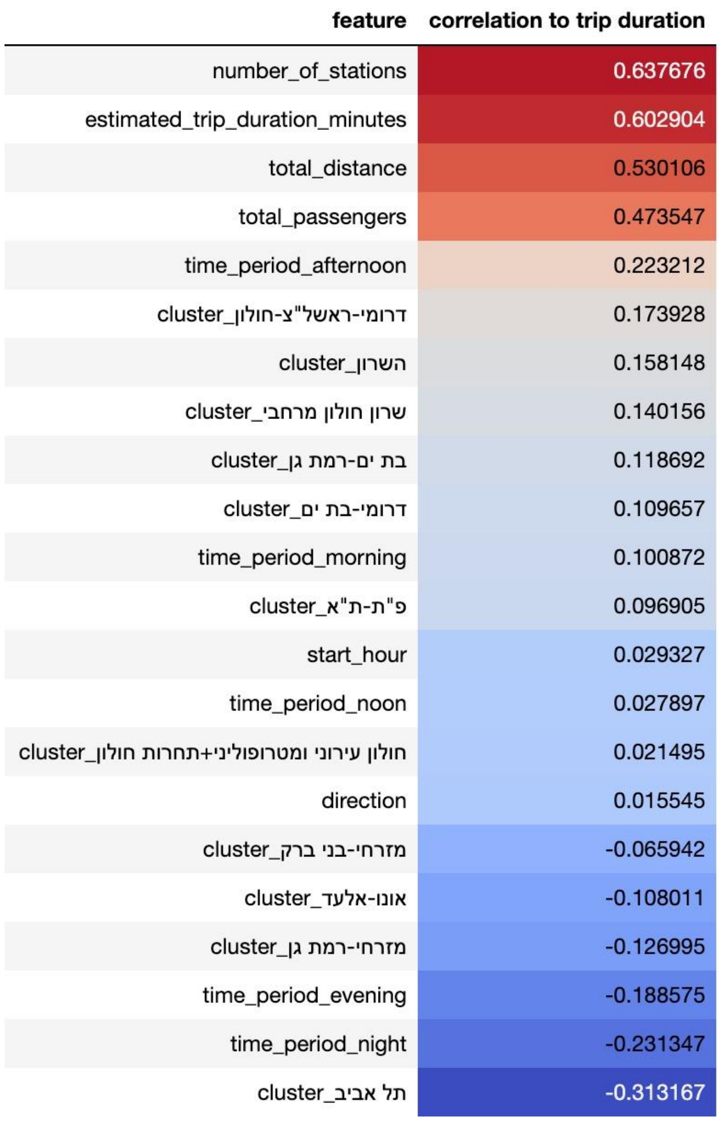
feature	Explanation	Passengers up prediction	Trip duration prediction
Total distance	from its first station to the last	🟩	🟩
Line count	(how many other bus lines pass through this current station)	🟩	🟥
Max line count	(saving the max line_count it passes through in it's route) : we believed that it can affect how many people might board the bus	🟩	🟥
density	how many other stations are around the current one according to gaussian kde algorithm: we used the kde algo that calculates distance from one station to next station.	🟩	🟥
Time period	morning being 7-11, Noon 11-15, Afternoon 15-19, Evening 19-23, Night 23-7	🟩	🟩
Start hour	what hour the bus started	🟩	🟩
Drive fraction	how much of the trip was completed	🟩	🟥
Max density	similar to max line_count - the maximum density of a station the line passes through	🟩	🟥
Number of stations		🟥	🟩
Estimated trip duration	Heuristic calculation according to the average time in each cluster and adjusted accordingly to the distance of each individual trip.	🟥	🟩
Total passengers		🟥	🟩
Direction		🟥	🟩
Cluster	categorical one hot encode	🟥	🟩

3. The considerations that guided our design of learning systems.

We started with quick and dirty baseline to see our starting point (linear regression with one feature) and an mse 3.67 for task 1 and mse = 20000 for task 2.

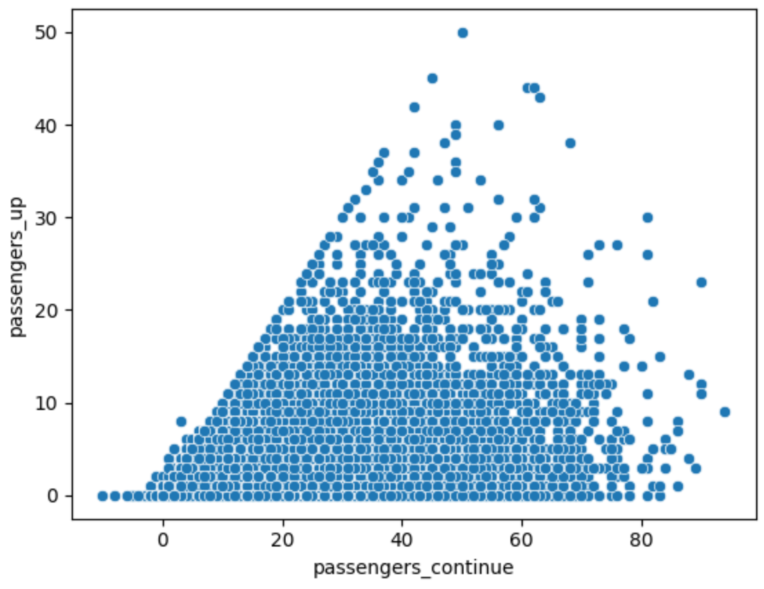
Then we iterated and added features as we went. We split the data to train/dev/test and evaluated via kfold cross validation.

For feature engineering we used features based on intuition and verified them using correlation to the y column:



feature	correlation to trip duration
number_of_stations	0.637676
estimated_trip_duration_minutes	0.602904
total_distance	0.530106
total_passengers	0.473547
time_period_afternoon	0.223212
cluster_דרומי-ראשל"צ-חולון	0.173928
cluster_השרון	0.158148
cluster_שרון חולון מרחבי	0.140156
cluster_בת ים-רמת גן	0.118692
cluster_דרומי-בת ים	0.109657
time_period_morning	0.100872
cluster_פ"ת-ת"א	0.096905
start_hour	0.029327
time_period_noon	0.027897
cluster_חולון עירוני ומטרופוליני+תחרות חולון	0.021495
direction	0.015545
cluster_מזרחי-בני ברק	-0.065942
cluster_אונו-אלעד	-0.108011
cluster_מזרחי-רמת גן	-0.126995
time_period_evening	-0.188575
time_period_night	-0.231347
cluster_תל אביב	-0.313167

We saw using the correlation heatmap between columns how the different features might affect the prediction. Notably the *passengers_continue* was the most correlated one to passengers up prediction:



Note that it seems the *passenger_continue* field includes the passenger up data (no value is above y=x line).

4. The various methods we tried and ended up not using them, and the results we obtained with them.

We tried using linear regression, however gradient boosting worked a lot better which indicated that the data is not linearly separable.

We tried creating features like *rush_hour* or things that have to do with the *door_close* column, but we didn't see enough impact on our model.

We also tried getting more complicated features, like elevation of ride. However, due to time constraints we could not manage to finish them.

5. The test error we expect our system's predictions would have is:

2.5 on task 1 and **80** on task 2.

Based on our splitting of the data, we kept a test set rather separate, and made sure it looks like how a real test set would be created.

We decided to sample entire trips instead of random rows based on how the test data looked, which affects how the model will train and hopefully make it more aligned with the test set.