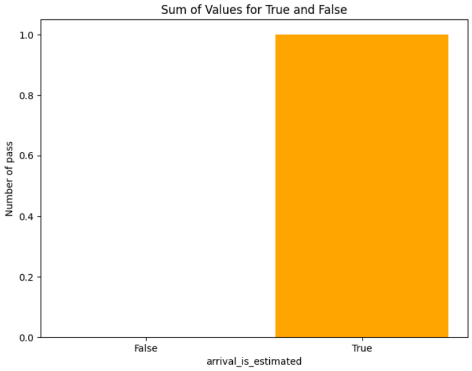
IML Hackathon

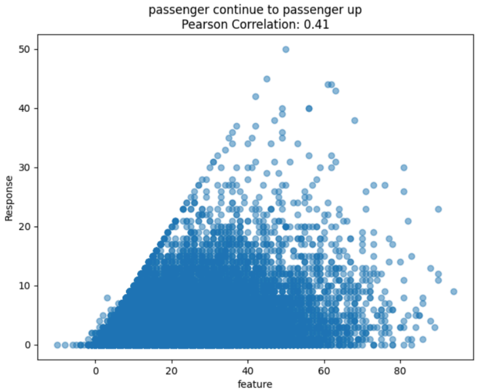
Understanding the data:

We started by simply looking at the data and understanding the meaning of each column. We also used Pearson correlations and scatter plots to better understand which features are relevant for the passengers_up and duration predictions. Here's what we found:

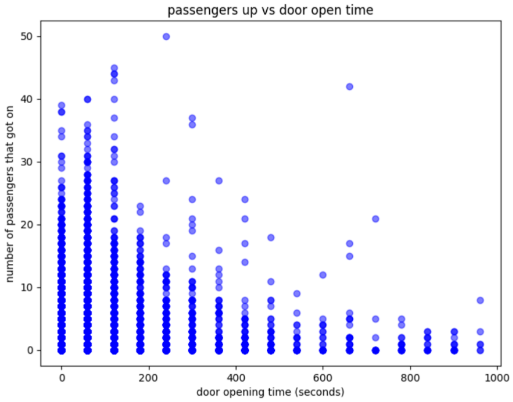
1. The 'arrival_is_estimated' column always aligns with zero values in the 'passanger_up'. This led us to manually predict that if the 'arrival_is_estimated' is TRUE, to predict that zero passengers got on. We believe this means that the bus didn't stop at the station.



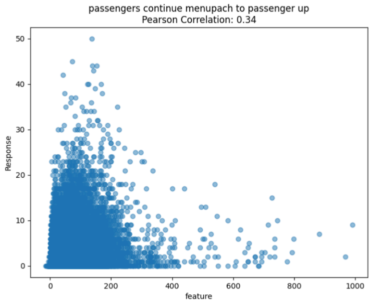
2. The 'passengers_continue' column is highly correlated with the 'passanger_up' column which makes sense.



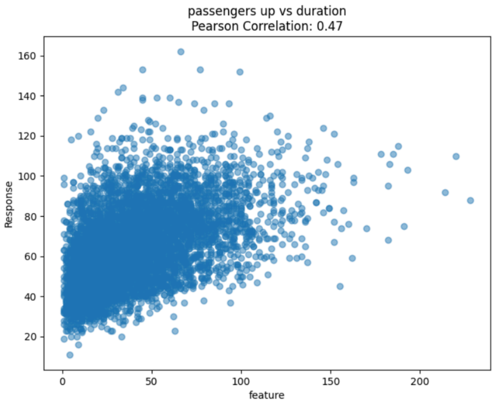
3. At first we thought that the amount of time the door of the bus was open would be highly correlated with the number of passengers that got on but strangely we saw the opposite effect (the shorter the time the door is open the less passengers got on). We are not sure what to deduce from this but because there still is an effect that can be observed this data is beneficial to our model.

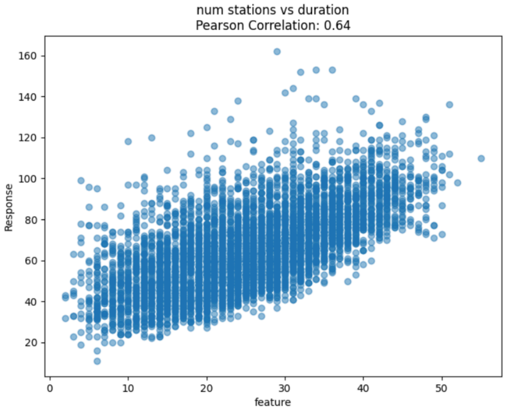


4. We noticed that ‘mekadem_nipuch_luz’ is not correlated with the number of passengers so we deleted the column in preprocess for part 1.
5. More interesting pearson correlations that help predict passenger up can be seen in the following plots:



6. More interesting pearson correlations that help predict duration can be seen in the following plots:





Preprocessing data:

We used a basic preprocess function to do an initial filter of data for both parts one and two. This included removing columns with non numeric values or changing them to categorical values. We also dropped Nan values and duplicate rows and values that were unreasonably high.

Part 1 - Passenger Boardings at Bus Stops :

In order to make the data usable for numerical models, we did the following preprocessing actions:

- Numerically labeled the “part”, “alternative”, “cluster” and “arrival_is_estimated” columns.
- Created a new column of the total time the door was open at the station.
- Removed outlier data.
- Removed empty rows and duplicate rows.

Next, we have applied several methods in order to increase the model performance:

- Added a binary feature indicating whether the stop time was during peak hours (morning: 7-10 AM, evening: 4-7 PM).
- Created interaction terms such as station_index * direction and latitude * longitude to capture complex relationships.
- Added features for the previous stop's passengers_continue and passengers_continue_menupach to capture temporal dependencies.
- Calculated the time differences between consecutive stops within a trip for both arrival and closing times.
- Applied one-hot encoding to categorical features line_id, station_id, and cluster to capture their unique identifiers.
- Scaled numerical features for better model performance
- Dropped the mekadem_nipuach_luz, trip_id_unique, trip_id_unique_station and station_name columns.

Part 2 - Predicting Trip Duration:

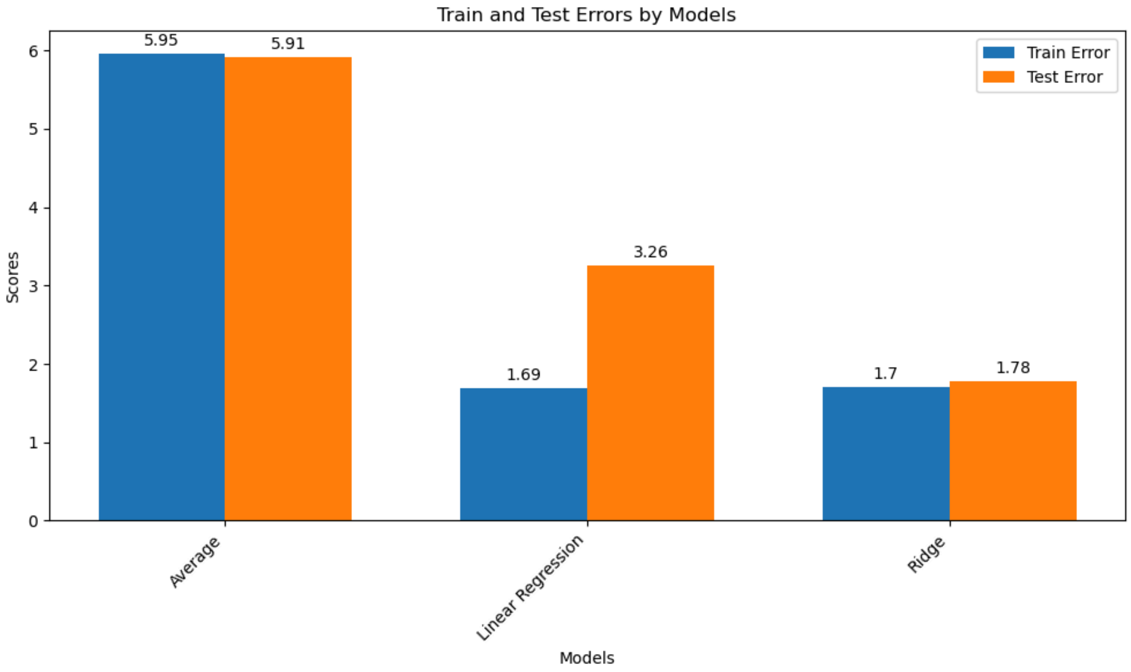
This included changing the entire structure of the input data where each row indicates a bus stop to a new table where each row is a single bus trip and with columns that may be correlated to duration. Because the exact label of duration did not exist we calculated it by taking the max arrival time minus the min arrival time for each bus ride. We also created columns from the categorical data such as trip_id and line_id. Moreover, we created new columns such as total number of stations of the line, and taking the mean passengers up ect.

Model selection

The models that we evaluated are Linear regression, Regression with ridge and predicting the average number (as reference to being better than a constant function). We split the data ⅔ for train and ⅓ for test and evaluated for each task.

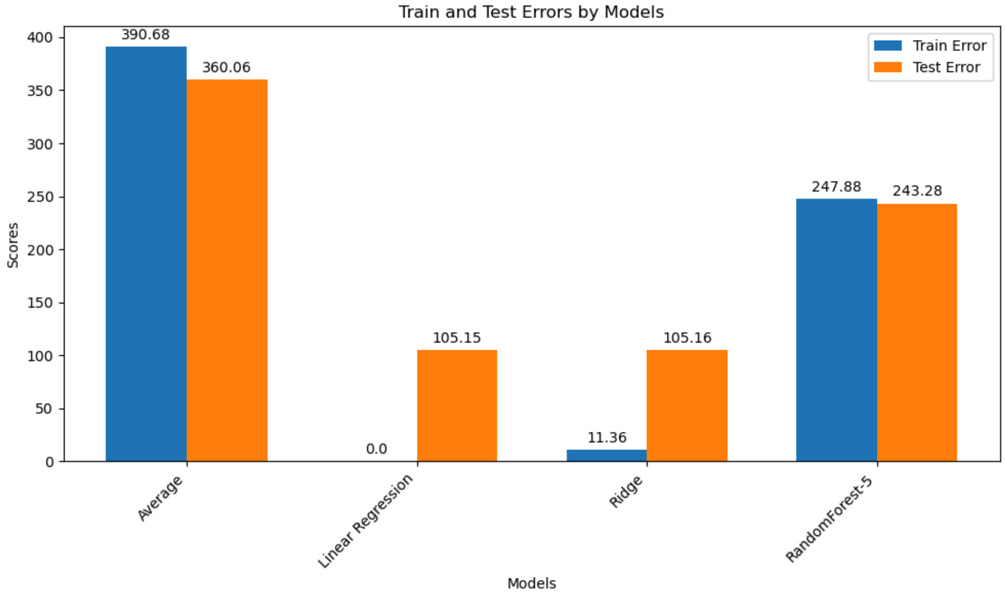
Part 1 - Passenger Boardings at Bus Stops :

The model that was the best fit for this task was: Ridge Linear Regression



Part 2 - Predicting Trip Duration:

The model that was the best fit for this task was Linear Regression but it achieved a high bias which can be observed by the difference between the train and test performance.



In the end we used a different model of Random Forest Regressor for last minute improvements