**Preprocessing steps**

1. Create tall datasets with bike dropoff / pickup information
   1. 1 row from the raw dataset is divided into 2 rows
   2. Act: **+1** means bike drop off, **-1** means bike pickup
2. Identify reshuffled trips
   1. Reshuffle occurred when a bike was picked up from station A, but was last seen to arrive in station B
      1. the bike was reshuffled from station B -> A
   2. The time for reshuffle happened exactly midway
      1. Station B drop off **(01:00AM)**, Station A pick up **(02:00AM)**, then reshuffle happened at **01:30AM**
3. Add starting number of bikes for each stations
   1. Since we are using entire 2019 year data, we used 2018 December to calculate this
      1. For each station, add up the total number of bikes that finished its trip on that station. That number will be the starting number for 2019.
4. Get total capacity for each station using API
   1. Drop stations that have no data on total capacity
5. Drop stations that have abnormally low (smaller than -10) average availability during 2019.
   1. Negative availability means that there were additional reshuffles that helped the inflow of bikes in that station, but the data cannot explain that. We chose to exclude these outliers so that we could focus on the reliability and accuracy of our model
6. Cap the availability at **[0, total capacity]** for all stations
   1. This makes sense because we want to know the overall movement of the availability in stations over time. We are **not** concerned with the exact availability at a given time.
7. Get the proportional availability by calculating **availability / total capacity**
   1. Proportional availability < 0.2 -> NO
   2. Proportional availability > 0.2 -> YES