Goals for next week 3/17:

1. Speeding:
    a. Connect the speeding alerts to Slack. For now, the speed limit is set to be 79 miles and if the car goes over 80 miles for more than 2 minutes, we send a message with the car license, driver's name, current speed, location.
    b. Sort the information by driver
        i. Match the # of minutes speeding with driver per trip, per day, per month and all historical data
2. Battery Charge Rate
    a. Use the maximum battery charge rate (throw away outliers) as the standard charge rate. We are ok if the rate is within 10%.
    b. Also connect it to Slack. If we detect abnormally low charge rate, push the notification with car location, license, current charge rate as a message.

AS -- Speeding Spec

Step 1: Query the database and obtain the speed, saved as a variable named current_speed together with the time stamp current_time. Every time we update our speed variable, we store the speed and time from last period in last_speed and last_time.

Step 2: Determine whether the car is speeding.

Variables created:

- Thresholds used: limit1, limit2, limit3
- Num_violations: store number of times the car speeds
- Accu_time_speeding: total number of seconds the car speeds during the entire trip
    - If we also want to know a more exact breakdown, we can create variables accu_time_speeding_level1 for speed between limit1 and limit2, accu_time_speeding_level2 for speed between limit2 and limit3, etc.
- Accu_miles_speeding: total miles driven while speeding

First cut: Fixed threshold = 75 mph, so we set variable limit = threshold.

- If we detect a speed that is above our limit, we consider the car to be speeding the entire period until we query the car again.
- If current_speed > limit, num_violations += 1, accu_time_speeding = accu_time_speeding + (current_time - last_time).

Second cut: Adjust speed limit based on location.

- Google map Roads API: Match Latitude and Longitude with road segment and posted speed limit.
- Additional information we need to obtain from each query: longitude, latitude.

- We use the longitude and latitude information of the car as input, and query the Google map Roads API to get the corresponding speed limit.
    - If we have a return value from the API, set limit = returnVal
    - If the APT returns NULL, set limit = threshold.
- Similarly, if current_speed > limit, num_violations += 1, accu_time_speeding = accu_time_speeding + (current_time - last_time).

Battery Charge Rate (Shaocong, Wenjing)

- **Main Idea**

What we want to do here is to check if the battery is being charged at a rate as we expect. We first analyze the historical Tesla data and find out the factors which may influence battery charge rate and construct models for battery charge time according to different combination of conditions. Then if there is significant difference between the actual charging rate and our expected one, we should check.

- **Historical Data Analysis:**

In this part, we analyze the relationship between charge rate and various physical features using historical data. We proceed by making corresponding regression models (Linear regression, ridge regression, least-square regression) for different factors.

Below are the factors we should take into consideration:

- Model (Specification of Car and battery).
- The specification of the charger.
- The frequency of battery charging.
- Age of battery and Car.
- Voltage of the electrical source.
- Weather (Humidity and temperature).
- Latitude, longitude.
- Height above sea level;

We first discard features which don't have any influence on the battery charge rate. Then we make models for predicting values such as battery level and time to full charge, for each combination of features. (Machine learning tools such as Decision Trees will be useful here).

- **Variables**:

We have below variables:

- Variables for each physical feature above.
- charging_Time: The length of time battery has been charged.
- actual_battery_Level: The quantity of electricity of the battery.
- predicted_battery_Level: The quantity of electricity of the battery predicted by our model.
- threshold_difference_in_battery_level
- required_Battery_Level: The quantity of electricity required for next trip.

    - time_To_Full_Charge

- **Monitoring**:
  By the data analysis part, we are able to predict how long will the battery take to finish charging under each real conditions. We follow below steps to monitor every 5 minute:
    - We calculate the predicted_battery_Level according to our prediction model using charging_Time and physical features.
    - If the difference predicted_battery_Level and actual_battery_Level is beyond our threshold, there may be malfunctions.
    - Try to detect the reason of malfunction automatically.
    - If the third step doesn’t work, we require manual check.

Don’t DO: To Be Covered by Battery charge team

- Is there possibility to run out of battery
    - Calculate the estimated miles the car can drive at current battery level L1.
    - Calculate the remaining miles for the current trip L2.
    - If 0 < L2 - L1 < 5, alert orange; if 5 < L2 – L1, alert red.
    - Should also push this information to driver and if they perceive it as a problem, they should be able to contact operation for solutions. (Ex. The location of the nearest charger)

Issue: UX Specification Needed - Slack to start, then something more visual and cumulative.

- Are cars braking suddenly or too jerky
    - Check whether the speed difference decreases by a huge amount within a short amount of time.
        - If the difference is more than 40 mph within 2 seconds, should show orange, if more than 50 within 2 seconds, should show red. The alert should be pushed to the driver’s ends and if multiple yellow alerts are triggered or any red alert, should notify the operation team.
        - Requires a check for speed every 2 seconds.
    - Change direction very fast.
        - Check the speed. If the wheel turns more than 180 degrees or the heading shift more than 80 degrees while speed is higher than 25 mph, mark yellow, higher than 30 mph, mark orange and mark red if more.

(Normal turning speed should be 10mph for U-turn and 15mph on most roads.)

- Required parameters: “speed”, “heading”, “shift_state”