Part 1 SQL Syntax

Comments:

- I have given myself about 20 min for this part of the challenge
- If I were to be given more time I would have come up with more optimal queries

1. For each of the cities 'Qarth' and 'Meereen', calculate 90th percentile difference between Actual and Predicted ETA for all completed trips within the last 30 days.

```
select percentile_cont(0.9) within group (order by t.actual_eta - t.predicted_eta) as
ninetiethPercentileDiff
from trips as t
left join cities as c on t.city_id = c.city_id
where c.city_name in ('Qarth','Meereen') and t.status = "completed" and t.request_at >
(current_date - interval '30 days');
```

2. A signup is defined as an event labeled 'sign_up_success' within the events table. For each city ('Qarth' and 'Meereen') and each day of the week, determine the percentage of signups in the first week of 2016 that resulted in completed a trip within 168 hours of the sign-up date.

```
select q1.city_name,q1.day_of_week,sum(q2.num_riders)*100/sum(q1.num_riders) as
percentage_of_drivers
from (
    select c.city_name as city_name,extract(dow from e._ts) day_of_week,count(distinct
e.rider_id) num_riders
    from events e
    left join cities c on e.city_id=c.city_id
    where e.event_name='sign_up_success' and c.city_name in ('Quarth','Meeren') and
extract(week from e._ts) = 1 and extract(year from e._ts) = 2016
    group by c.city_name,extract(dow from e._ts)
) q1
left join (
    select c.city_name as city_name,extract(dow from e._ts) day_of_week,count(distinct
e.rider_id) num_riders
    from events e
    left join cities c on e.city_id=c.city_id
    left join trips t on e.rider_id=t.driver_id

    where e.event_name='sign_up_success' and c.city_name in ('Quarth','Meeren') and
extract(week from e._ts) = 1 and extract(year from e._ts) = 2016 and t.status='completed'
and date_part('day',t.request_at-e._ts)*24+date_part('hour',t.request_at-e._ts)<168

    group by c.city_name,extract(dow from e._ts)

) q2 on q1.city_name=q2.city_name and q1.day_of_week=q2.day_of_week

group by q1.city_name,q1.day_of_week
```

Part 2 Experiment and metrics design

Comments:

- I have given myself about 30 min for this part of the challenge
- If I were to be given more time I would have come up with more optimal and robust testing strategies
- I would have also experimented with Bayesian A/B testing strategies if they are feasible in such a setup
- Besides the metrics that I have mentioned below I would also recommend logging the driver-partner activity in the new user app to understand their behavior and help identify any underlying patterns

The Driver Experience team has just finished redesigning the redesigning the Uber Partner app. The new version expands the purpose of the app beyond just driving. It includes additional information on earnings, ratings, and provides a unified platform for Uber to communicate with its partners.

1. Propose and define the primary success metric of the redesigned app. What are 2--3 additional tracking metrics that will be important to monitor in addition to the success metric defined above?

"Driving with Uber means more than just being on the road". I would want drivers to better engage with the app. ***Driver engagement*** is the key here. What is the purpose of all the new/improved features if the driver is not actively using them.

Engagement can be quantified in several ways one of which could be in terms of amount of time spent on the new app.

Across all the sections, this could be time spent in each of the sections, time spent overall, time spent looking at ads (if any) etc.

We can also quantify engagement with how frequently the driver checks the new app.

We can log the metrics at the lowest level possible (for each second) and aggregate the results at daily/weekly level(s) to get a bigger picture.

<u>Each section in the new partner app can have individual metrics:</u>

- If we can correlate new/improved "Heat map" in "Home" section to reduced turn-around time between rides and reduced cancellation of rides due to "driver is far away" reason by the people requesting the ride
- If we can correlate new/improved "Earnings" section to higher earnings for the driver partner, it would be another metric that can quantify engagement and ultimate success of the new partner app. Higher driver earnings mean higher share for Uber and a happy driver partner.
- If we can correlate new/improved "Ratings" section to the variation in the suggestions the driver is receiving (if there is a new suggestion then it could mean that he had improved the old suggestion) or reduced number of suggestions he received (if he followed all suggestions and there is nothing new to suggest for him) then it can quantify engagement with the new app
- If we can correlate new/improved "Account" section to the more accurate/relevant/up-to-date information of a driver being available then it can quantify engagement with the new app

Each of these metrics give a peep-hole view of the app. To quantify/understand the overall success/failure of the app I need to model, aggregate and summarize the results from each of the metrics.

2. Outline a testing plan to evaluate if redesigned app performs better (according to the metrics you outlined). How would you balance the need to deliver quick results, with statistical rigor, and while still monitoring for risks?

If this is not the first time that such an experiment is being conducted by the team then I would try to leverage any of the existing A/B testing framework(s) at Uber for the task.

If this is the first time then I would first cluster drivers based on their historical statistical data of their engagement with old partner-app.

I would then select a cluster of drivers (who have similar usage statistics) and upgrade 50% of the drivers to new app while the other 50% continue to use the old app.

If the metrics are logged at each second then 1 month would be sufficient to capture intra-minute, intra-hour, intra-day, intra-week and intra-month seasonalities.

I would leverage Michelangelo platform to identify "patterns" in the test group and look for "anomalies" and "break-points" in their usage statistics. For a quick and dirty hypothesis testing I would use a paired t-test to check if the difference in the usage statistics is significant or not and with what level of confidence. These results can be delivered extremely quickly with statistical rigor.

As mentioned earlier we can monitor for anomalies in the above-mentioned metrics at each second for each driver to identify any potential risks and prevent a bigger impact before the end of the hour/day. The thresholding for detecting anomalies can be set by the business team. The anomalies can be differentiated as anomalies at driver end or at server(s) hosting the app and the appropriate action can be taken accordingly

3. Explain how you would translate the results from the testing plan into a decision on whether to launch the new design or roll it back.

I would make sure I am comparing apples to apples by nullifying all other effects like increase/decrease due to holidays, increase/decrease due to emergencies etc. I would want to make sure that the effect (if any) is due to the improvements alone in the new-app.

I would identify that the increase/decrease in the metrics is consistent and not temporary.

I would also make sure that there are appropriate trend, seasonality and other patterns in the observations.

I would also take into consideration the results of the metrics and construct a forecast for the next 18 months. I would then compart this forecast with the forecast for the existing app. If the net profit (including all the costs of development, testing, deployment and maintenance) (in $) is substantial I would go ahead and launch the new design. If the numbers don’t look good I would like to see if this may have a positive long-term impact. If that is the case then I would go ahead with the launch else I would try to incorporate the insights obtained during the first trial run and try to come up with a better plan the next time