Uber Data Science Challenge

Brendan Herger, 13herger@gmail.com
Code, and this write up, are available at https://github.com/bjherger/Uber-DS-Challenge

Part 1

Nota Bene: I've made a few assumptions, such as what time best represents a trip, and that users do not have multiple completed sign ups. My decisions are heuristic, and normally would be confirmed with existing staff or the product owner before releasing analysis.

Additionally, I've put little emphasis in manual query optimization. For example, for question 2 could subset sub-tables for the for the first week of 2016. I assume that the query planner used is smart enough to make these optimizations.

Question 1

Q: 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.

A:

```
SELECT
  PERCENTILE_CONT(.9)
    WITHIN GROUP (ORDER BY trips.actual_eta-predicted_eta)
  AS 90th_percentile
 FROM trips
  LEFT OUTER JOIN cities
   WHERE trips.city_id == cities.city_id
  WHERE cities.city_name IN ('Qarth', 'Meereen')
    AND trips.status == 'completed'
    AND trips.request_at > (CURRENT_TIMESTAMP- INTERVAL '10 days');
```

Question 2

Q: 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.

A:

This query is somewhat contrived. If this is a common data access pattern, I would normally work with the product owner to understand if the sub-tables I generate (*first completed ride timestamp* and *rode in first week*) are valuable enough to be stored on their own, or if there is a more optimal way to store this data in a more accessible way.

```
SELECT signups_enhanced.day_of_week, AVG(rode_in_first_week::int)
  FROM

  -- Create sub-table with one row for every rider who signed up, with rode_in_first_week metric
  ( SELECT events.*
      EXTRACT( DOW FROM _ts) AS day_of_week
      -- Actually compute rode_in_first_week metric
      -- Check if user has a ride
      (MIN(trips.request_at) IS NOT NULL
       -- First ride within 168 hours
      AND MIN(trips.request_at) <= MIN(events._ts) + INTERVAL '168 hours'
      -- No rides before sign up
      AND MIN(trips.request_at) >= MIN(events._ts))
        AS rode_in_first_week
    FROM trips
    LEFT OUTER JOIN

      --   Create sub-table with every rider's first completed trip
      (SELECT DISTINCT ON (trips.client_id) trips.client_id, request_at
        FROM trips
          WHERE trips.status == 'completed'
        ORDER BY trips.request_at ASC
      ) AS first_completed_trips

      WHERE events.rider_id  == first_completed_trips.client_id
      AND event_name == 'sign_up_success'
  ) AS signups_enhanced

  GROUP BY signups_enhanced.day_of_week
  WHERE EXTRACT(WEEK FROM signup_ts) == 1
    AND EXTRACT(YEAR FROM signup  ts) == 2016;
    AND city_name IN ('Qarth', 'Meereen');
```

Part 2

Question 1
Q:
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?

A:
Ideally, during planning of the new release I would work with the team behind the new release to identify their goals and metrics that capture them.

I would propose the following metrics:

- **New feature time:** The amount of time spent in the four new drive app sections (Home, Earnings, Ratings, Account)
- **Driver Productivity:** The difference in total fares seen by drivers who use the new app and to drivers who use the old app
- **Driver help requests:** The difference in driver contacts (e.g. email, phone) to Uber between drivers who use the new app and drivers who use the old app

Additionally, before the trial begins I would work with the product team to choose acceptable thresholds for each metric for a new release. For example, we might use the thresholds below, with a pre-defined statistical siginificance:

Metric	Threshold
New feature time	20 minutes a week or more
Driver productivity	No change or increase
Driver help requests	No change or decrease

Question 2

Q:

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?

A:

Existing protocols

First, I would reach out to the team that is commonly tasked with A/B testing the rider app, and seek their recommendations or general testing framework. This would provide consistency in mobile A/B testing, and reduce redundant work in developing testing frameworks.

Assumptions

Additionally, I would confirm a few assumptions:

- Driver app versions before the new release are not substantially different
- Driver apps are not substantially different based on operating system (for both the current release and the new release)
- The metrics above meet product owner needs

Test design

In the absence of a testing plan from the rider app team, I would proceed with the following trial:

Segmentation: 3 distinct geographic locations, with 25% of drivers in each location forcibly upgraded to new version, and the remaining 75% forcibly frozen in their current version
Length: 1 month
Safety checks: Hourly checks for drop in driver productivity (in case of app bugs or crashes), and monitoring driver help requests (to ease burn in period, watch for bugs or crashes)

Summary

I feel that this trial design would be large and diverse enough to capture meaningful signal, without unduly exposing a large population of drivers to an unproven re-design. Additionally, network effects (e.g. drivers with the old app seeing drivers with the new app) should be minimal, and could be controlled by branding the new release as a 'pre-release' version. Finally, a one month test period should be enough to gather statistically significant results, and quickly iterate on the testing framework.

Question 3

Q:

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.

A:

I would evaluate each of the metrics designed at the onset of the trial, relative to the thresholds designated at the onset of the trial.

I would then identify if the thresholds and statistical significance levels were appropriate, and adjust the thresholds and statistical significance levels as appropriate.

Part 3

Nota Bene: Please see code, located https://github.com/bjherger/Uber-DS-Challenge. Code for this question is located at bin/q3.py

Question 1

Q:

Perform any cleaning, exploratory analysis, and/or visualizations to use the provided data for this analysis (a few sentences/plots describing your approach will suffice). What fraction of the driver signups took a first trip? (2 points)

A:

Please see plots below. Approximately 0.11% of drivers listed in the provided data have a first trip date. I assume any drivers that do not have a first trip date have not completed a trip.

Question 2

Q:

Build a predictive model to help Uber determine whether or not a driver signup will start driving. Discuss why you chose your approach, what alternatives you considered, and any concerns you have. How valid is your model? Include any key indicators of model performance. (2 points)

A:

Note: In order to iterate rapidly, I've limited myself to 2 hours to develop the following models. While more advanced modeling is possible (perhaps even preferable), I believe this is a realistic amount of time for a brief pass at this problem.

Executive Summary: We should focus on getting more potential drivers to have their vehicles inspected, which appears to be a bottleneck in our funnel.

Model choice

As there is a stated preference for interpretable models, I've elected to use logistic regression. Whereas a decision tree would have also lead to an interpretable model, it would be more difficult to discuss variable importance without delving into branching logic. If a more predictive model were necessary, I'd likely move forward with an SVM model, Random Forest model and / or neural network. For further accuracy, I might ensemble these approaches, and look at more advance grid searching / tuning of hyper parameters.

Model validation

Continuing the descriptive emphasis, I've elected to primarily focus on the statistical significance (keeping variables with alpha < .05) of coefficients in tuning the models for this first pass. I've also computed AUC for a 20% holdout, to verify that the models maintain some predictive accuracy.

For more predictive modeling, I would likely use 5 fold cross validation, again optimizing AUC. I might also look at the confusion matrix, as well as the cost associated to true signups who did not drive, true signups who did not drive, type 1 and type 2 errors.

Model results for all drivers

I've run two models. The first covers all observations, and suggests primarily that drivers with a vehicle inspection are most likely drive (suggesting that funnel analysis might be more appropriate, and that effort should be put into increasing the vehicle inspection rate for new drivers). The model also suggests that potential drivers are more likely to complete a ride if they were referred, signed up on a weekday, and / or were from Berton. Raw model output is available at the end of this report.

Model results for drivers with vehicle inspection, background check

The second model I ran was subset to drivers who had completed a background check and vehicle inspection. These drivers were further down the funnel, and already more likely to drive than there peers who had not completed these two steps. However, subsetting the data also allowed for more detailed analysis.

This model showed that signups were more likely to have first rides if they had a newer vehicle, completed their background check earlier, and completed their vehicle check later. Raw model output is available at the end of this report.

Question 3

Q:

Briefly discuss how Uber might leverage the insights gained from the model to generate more first trips (again, a few ideas/sentences will suffice). (1 point)

A:

I would emphasize the value of modeling the full funnel between signup and first drive. Furthermore, I would suggest focusing on getting more potential drivers to have their vehicles inspected, which appears to be a bottleneck in the funnel.

Moreover, I would suggest moving towards predictive modeling, and using those models to prioritize contact with drivers. For example, a model predicting which drivers will not complete a first ride could help allocate background checks, vehicle checks and contact from Uber associates.

Model results

Below is the un-interpreted output from the models run. This output should be read in tandem with reviewing the modeling code, located https://github.com/bjherger/Uber-DS-Challenge. Code for this question is located at bin/q3.py

Model results for all drivers

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                  drove   No. Observations:                43745
Model:                          Logit   Df Residuals:                    43740
Method:                           MLE   Df Model:                            4
Date:                Mon, 01 Aug 2016   Pseudo R-squ.:                 0.4554
Time:                        22:49:15   Log-Likelihood:                -8403.5
converged:                       True   LL-Null:                       -15430.
                                        LLR p-value:                     0.000
=====================================================================================================
                                        coef    std err          z      P>|z|      [95.0% Conf. Int.]
-----------------------------------------------------------------------------------------------------
Intercept                            -5.3724      0.074    -72.934      0.000        -5.517    -5.228
signup_channel_referral[T.True]       0.4938      0.038     12.906      0.000         0.419     0.569
city_Berton[T.True]                   0.1165      0.039      2.966      0.003         0.040     0.193
signup_weekday[T.True]                0.3623      0.042      8.619      0.000         0.280     0.445
vehicle_inspection_known[T.True]      4.6163      0.073     63.663      0.000         4.474     4.758
=====================================================================================================
AUC for 20% holdout: 0.922580207546
```

Model results for drivers with vehicle inspection, background check

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                  drove   No. Observations:                10309
Model:                          Logit   Df Residuals:                    10304
Method:                           MLE   Df Model:                            4
Date:                Mon, 01 Aug 2016   Pseudo R-squ.:                 0.2057
Time:                        22:49:15   Log-Likelihood:                -5647.5
converged:                       True   LL-Null:                       -7110.3
                                        LLR p-value:                     0.000
=====================================================================================================
                                        coef    std err          z      P>|z|      [95.0% Conf. Int.]
-----------------------------------------------------------------------------------------------------
Intercept                             0.6030      0.041     14.599      0.000         0.522     0.684
signup_channel_referral[T.True]       0.4867      0.046     10.673      0.000         0.397     0.576
city_Berton[T.True]                   0.0860      0.047      1.829      0.067        -0.006     0.178
signup_to_vehicle_add                 0.1751      0.019      9.169      0.000         0.138     0.213
signup_to_bgc                        -0.1758      0.005    -38.530      0.000        -0.185    -0.167
=====================================================================================================
AUC for 20% holdout: 0.794798031562
```