Part 1 - SQL Syntax
[2 points]

Given the below subset of Uber's schema, write executable SQL queries to answer the questions below. Please answer in a single query for each question and assume read-only access to the database (i.e. do not use CREATE TABLE).

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 avg(actual_eta-predicted_eta),
        percentile_cont(0.9) WITHIN GROUP (ORDER BY actual_eta-predicted_eta)
  from trips a left join cities b on
      a.city_id=b.city_id
  where b.city_name in ('Qarth','Meereen')
   and DATE_PART('day', CURRENT_TIMESTAMP - request_at) <=30
   and status='completed';
```

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 t2.city_name, t1.day_of_week, (100*sum(t1.completed_trip)/count(*)) as
signup_percentage
from (
    select a.city_id, extract(DOW from a._ts) as day_of_week,
        CASE WHEN DATE_PART('hour', a._ts - b.request_at)<=168 and b.status =
'completed'
            THEN 1
            ELSE 0
        END as completed_trip
    from events as a
    join trips as b
    on a.city_id = b.city_id
    where a.event_name = 'sign_up_success'
    and a._ts between '2016-01-01 00:00:00'::timestamp and '2016-01-08
00:00:00'::timestamp
   ) as t1
join cities as t2
on t1.city_id=t2.city_id
and t2.city_name in ('Qarth','Meereen')
group  by  1, 2;
```

Part 2 - Experiment and metrics design
[3 points]

The Driver Experience team has just finished 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 design the primary success metric of the redesigned app. What are 2-3 additional tracking metrics that will be important to monitor**

   Primary success metric:
   - **Increase in driver engagement** on the new app as opposed to driver engagement on the old app (How frequently and which new features from the redesigned app are being used. Define a threshold which indicates an increase in the engagement (number of times/week))

   Additional tracking metrics:
   - **Increase in driver earnings / trips** (since now the driver has real time access to his trip earnings and can plan for his target earnings for the day and act upon it)
   - **Increase in driver rating** (since now the driver has real time access to his trip rating and can act upon his shortcoming from the trip in real time to improve on his service and prevent his rating from going down further)
   - **Increase in overall Uber customer engagement** as the drivers receive real time notifications from Uber through the app as well as drivers can seek assistance from Uber, thereby improving customer service.

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?**

   Step 1 – Check if an existing testing plan is in place or add to it if possible. Perform A/B tests on ensuring bug free code or monitoring for bugs.

   Step 2 -- Set A/B Tests with the help of Test group and Control group (i.e. Randomly divide the users from the same segment (i.e. same gender, age group, geography, same shift (day/ night), similar car (small vs xl)) into Test group and Control group.

   Check out how many users from the test group continue using the redesigned app and what impact it has on the metrics defined in part 1 for a fixed test duration (decide on a threshold upon discussion with the team. e.g. one month). Insights can be derived on which user segment tends to use the redesigned app more frequently.

   Try a test to determine how frequently the users in the test group use the app and which features do they use vs. how frequently the users in the control group use the

app and which features do they use. If the use of the redesigned app is more than the old app and new features are being used then the result of the test is positive else negative.

Create a table with test results and another table with users.

user table – information about users
Columns –
user_id – id of user. Primary key
sex – gender (Male / Female)
age – age
shift – shift user operates in (day / night)
city – city user operates in
state - state user operates in
country - country user operates in

test table – information about test results
Columns –
user_id – id of user, unique by user. Foreign key (refers to primary key user_id from user table)
date – date the app (old / redesigned) was used the first time since the test started
date_last_used – date the app (old / redesigned) was last used
frequency_of_use – number of times the app is used during the test duration
features_used – the features in the app (old/ redesigned) that are being used
app_language – language selected by the user in the app (in case the test is performed on a limited population from multiple countries)
test_group – 1 / 0. Test = 1 would be users who're in the test group and have been provided access to the redesigned (new app). Test =0 would be users who're in the control group and have access to / use the old app.
conversion – label (target variable) if the user uses the app for a certain number of times during the test period then a label of 1 will be assigned for the user specific test result else a label of 0 will be assigned.

Quick results can be delivered by analyzing the A/B test steps and running one A/B test for monitoring each metric by automating each of the A/B test. An algorithm can be designed to automate the steps.

This trial/test would be performed on a limited scale of the population that would ensure that the entire population is not affected by the app redesign in case of undesirable results but at the same time on a large enough scale to obtain statistically significant results which guide an informed decision on the launch of the new product.

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.**

   If the test set yields better improvement in terms of the metric results as opposed to the metric results within the control group over a period, it should be acceptable to launch the new design

   For example - If for a certain user, the frequency of use over a specified test period is over an agreed upon threshold, then a value of 1 is assigned to the conversion feature for that user else a value of 0 is assigned.

   If the number of conversions in the test table is over a certain threshold agreed upon with the product management team, then the test is considered a success and is a good enough sign for the product to be launched. If on the other hand, the number of conversions are less than the threshold, the product or new redesigned app would be rolled back.

Part 3 - Data analysis
[5 points]

Uber's Driver team is interested in predicting which driver signups are most likely to start driving. To help explore this question, we have provided a sample1 dataset of a cohort of driver signups in January 2015. The data was pulled a few months after they signed up to include the result of whether they actually completed their first trip. It also includes several pieces of background information gather about the driver and their car.

We would like you to use this data set to help understand what factors are best at predicting whether a signup will start to drive, and offer suggestions to operationalize those insights to help Uber.

See below for a detailed description of the dataset. Please include any code you wrote for the analysis and delete the dataset when you have finished with the challenge. Please also call out any data related assumptions or issues that you encounter.

1. **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)***

   Refer to Data cleaning, exploratory analysis in this file - > Signup.ipynb

   During exploratory data analysis, it has been observed that 11% of the driver signups took a first trip (11% of the driver signups had a date value for the 'first_completed_date' feature.

   Exploratory Data Analysis and cleaning–

   - Determining the number of features, instances, null values(NaN) in the dataset.
   - Determining unique values for each of the features.
   - Determining null values for each of the features.
   - Determining the distribution of values for each of the features.
   - Assign value 9999999999 to vehicle_year feature where the value was 0(for four records)
   - Imputing data for null values
     - Converting 'date' columns to datetime format. NaN values would be converted to NaT values.
     - Imputing data for 'vehicle_make', 'vehicle_model' variable null
     - values with 'missing'
     - Impute 9999999999 for NaT and other missing values for all the features except 'vehicle_make', 'vehicle_model'

   Observations-

   - The information about the dataset states 'To help explore this question, we have provided a sample dataset of a cohort of driver signups in January 2015.' but the unique values for the 'signup_date' feature indicates all the signups for the dataset provided happened in January 2016.
   - The name of the feature in the dataset information provide is 'signup_timestamp' which is the timestamp of account creation (local time) whereas in the dataset it is 'signup_date' and doesn't seem to include the time.
   - The date format specified in the dataset information is 'YYYY-MM-DD' whereas in the dataset it is 'MM/DD/YY'
   - The possible values for the 'signup_channel' feature in the dataset information provided are '['offline','paid','organic','referral']' whereas in the dataset the possible values are '['Paid', 'Organic', 'Referral']'.
   - The name of the feature in the dataset information provide is 'city_id' whereas in the dataset it is 'city_name'.

- The possible values for the 'signup_os' feature in the dataset information provided are '['android','ios','website','other']' whereas in the dataset the possible values are '['ios web', 'windows', 'android web', nan, 'mac', 'other']'.
- The name of the feature in the dataset information provide is 'first_trip_date' whereas in the dataset it is 'first_completed_date'.
- Number of null values for the 'first_completed_date' feature = 48544(of 54681) imbalanced classes dataset.
- The vehicle make includes questionable values such as 'Bicycle', 'Bike', 'Walker'.
- Validity of the provided models. (Avenger - used in military, IS, F-150-pickup truck etc.)
- Each year has a period at the end which needs to be cleaned. The dataset description specifies year in the form YYYY
- year 0 exists for this feature which needs to be cleaned
- bcg_date seems to be highly correlated with the target variable. If bcg_date is null then first_completed_date is always null i.e. the driver signup cannot start driving.
- Anomaly - Signup date is later than that of Vehicle added date?

Feature Engineering -

- Create a new column 'driving' as target variable with value 1 for 'driving' and value 0 for 'not_driving' based on whether ‘first_completed_date’ exists.
- Create new features for signup_bgc_dtdiff, signup_veh_added_dtdiff, bgc_veh_added_dtdiff
- Create features for categories based on the percentile distribution of date difference values in days.
- Creating Dummy features for categorical features such as 'city_name','signup_os_mod','signup_channel' and all the new categorical features created which are non-numeric.
- Creating a feature for each of the date features. Assign 0 for missing date and 1 if date exists

Refer to the Data cleaning, exploratory analysis, feature engineering in the attached file **- > Signup.ipynb**

Feature Selection -

- Used Decision tree (Extra tree ensemble) for obtaining the feature importance for each of the features and ranking them in descending order of importance.
- Determined the most valuable variables in descending order of correlation
- Followed the process of iterative addition of features in a logistic regression model using the statsmodel package to obtain the P values and coefficients.
- Followed the process of iterative removal of variables from the model

considering a cutoff of 0.25 for the P value.
- Obtained the final list of features to be used for modeling.
- 18 features were used in the modeling process from the 62 features.

Refer to the Feature selection in the attached file ->
Signup_EDA_FeatureSelection.ipynb

2. **Build a predictive model to help Uber determine whether 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)***

- The task of determining whether a driver signup will start driving is a type of supervised classification problem. The 'Driving' feature has been created during data preparation phase with binary labels (1 and 0 indicating 'driving' and 'not driving' respectively) based on the presence/ absence of 'first_completed_date'.
- The dataset was split randomly into (60% - train set, 20% - holdout set, 20% - test set) in order to train the model on the training set, evaluate the model performance on the test set and determine the model's generalization accuracy on the holdout set.
- The target variable 'Driving' has imbalanced classes. There are significantly more number of zeroes (not driving) than ones (driving) labels for this feature. This could be a matter of concern specifically during model performance evaluation.
- Created the following models, results of which would enable selection of the appropriate model based on the business needs.
    - Logistic Regression with scikitlearn
    - Logistic regression with statsmodel (provides interpretability in terms of the weight and thus the importance of each of the features.)
    - Decision Tree
    - Gaussian Naïve Bayes
    - Support Vector machine
    - K Nearest Neighbors

Modeling Results Summary:

Classifiers	Test Sample				Hold Out Sample			
	Accuracy	Recall	Precision	AUC	Accuracy	Recall	Precision	AUC
Logistic Regression	0.94	0.76	0.74	0.97	0.94	0.76	0.72	0.97
Decision Tree	0.93	0.71	0.73	0.92	0.93	0.68	0.70	0.90
Gaussian Naïve Bayes	0.91	0.93	0.56	0.95	0.90	0.95	0.53	0.96
Support Vector Machine	0.94	0.80	0.74	0.95	0.94	0.79	0.71	0.95
K Nearest Neighbors	0.94	0.94	0.94	0.83	0.94	0.73	0.72	0.96

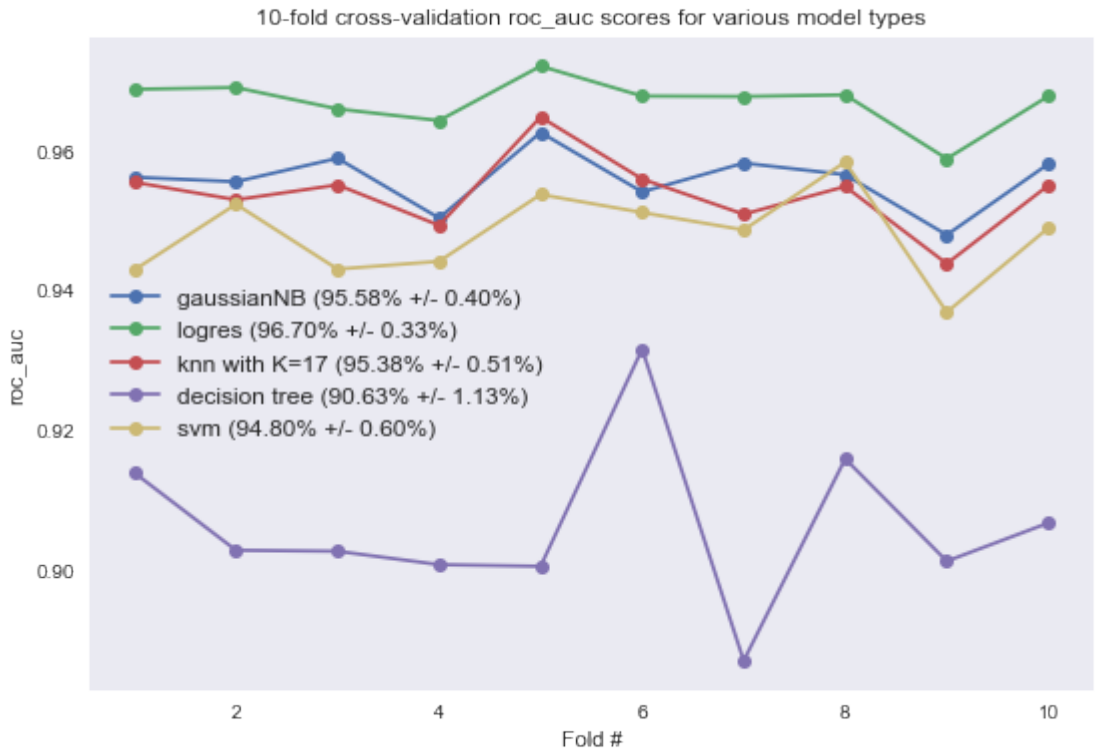
Metric Selection –

As we have a case of imbalanced classes, 'Accuracy' doesn't seem to be an appropriate metric for model evaluation in our case.

It would be appropriate to look at the Recall, Precision and Area under the Curve(AUC) metrics for evaluating model performance. Business objective should be used to decide on an appropriate threshold for precision and recall tradeoff.

Model Selection –

Model selection should be based on Uber's business objectives. For example, if the model was utilized to predict future samples and prioritize marketing investment, it is advisable to understand tradeoff between cost incurred per conversion versus the business benefit.



Based on results of 10-Fold cross validation roc_auc score depicted in the plot above and the recall and precision values from the model results in the table above, Logistic Regression seems to be a better choice for predicting conversion. Given its interpretability and ease of incorporating new training data, it could be a good candidate for this business problem.

Refer to the modeling process in the attached file -> **Signup_modeling.ipynb**

3. **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)**

   After close evaluation of various models and predictive variables, I found following factors are best at predicting whether a signup will start to drive:

- The difference between vehicle added date and signup date to be the most predictive of whether a signup will start to drive. The smaller the difference i.e. the closer the vehicle added date to signup date the more likelihood of a signup to start driving.

- The second most important variable appears to be the difference between background

check consent date and signup date. The closer the background check consent date to the signup date, the more likelihood of a signup to start driving.

- The next two important variables were “Referral” signup channel followed by vehicle model “Prius”.

- Although the signup city “Berton” came up as one of the predictive variables (with lower order of importance), I’ve put less emphasis on this variable to make the machine learning application more generalized (less bias to geography).

With the limited business operating model understanding combined with the learnings from the data modeling exercise, I would offer recommendations in three board operational areas that can help Uber.

1. **Evaluate Process Improvement Opportunities**: Map the entire signup process to explore opportunities that can help drivers add the vehicle information during (close to) the signup process. Also, evaluate the process workflow to understand if any improvements can be done to help expedite the background consent. This assessment will help design the process that will be more efficient and effective at the same time improve the driver conversion rate.

2. **Prioritize Marketing Investments**: Application of the model to future data samples can help prioritize marketing investment towards drivers that are more likely to start driving. For example, a targeted outbound telemarketing approach to facilitate and expedite information gathering on vehicle and driver background is likely to result in higher conversion rate, less time to conversion and better diver signup process experience. Insight from data also revealed “Referral” signup channel being more effective thus suggesting that marketing spend in referral channel can help grow the business.

3. **Grow the Uber Xchange Leasing business**: The insights from data modeling, highlights Vehicle Added date is most predictive to whether a signup will start to drive, however this data field has significant instances with missing or no information. In case vehicle access and availability happens to be an underlying issue with majority of these signups, there may be an opportunity to provide these signups with low-cost vehicle leasing options and thus generate more leads for Uber Xchange.