DATA INTUITION: Candidates answer all questions in this section

Question 1:

Assume that you have a communication graph of large number of users u. Denote all users by set U. Interaction at time t between two users u1, u2 is represented by r12(t)=r(u1,u2,t). For each user u, you have information such as:

- The main location of that user (e.g. district)
- The time series of phone balance and top up

For interaction r12(t)=r(u1,u2,t)

- [A01] If users interactions are via phone calls, there is information such as call time, call duration, rough locations for both two users. The content of the phone call is not available though.
- [A02] In the case of Facebook data, there is information such as: the public message user u1 post in u2's home, message time, number of likes, etc.

There are some labeled data where the income of users in set P is observed. Note that set P is much smaller than set U. P is a subet of U. Income of user u1 at time is recorded and denoted by q(u1,t)

Assume each distinct user only has one income observation at a specific time

a/ Please consider the case [A01] and derive the features which are important to predict user income

b/ Please do the same thing for the case [A02]

DATA FRAME CODING AND METHODOLOGY UNDERSTANDING:

Question 2:

Please read Ping Li, Trevor Hastie, and Kenneth Church, Very sparse random projections http://ww.web.stanford.edu/~hastie/Papers/Ping/KDD06_rp.pdf

a. Lemmas 1 to 3 summarize the main results and contributions of this work. Please explain these results in words. How does this work extend what is known from previous work by Achlioptas on constructing Johnson-Lindenstrauss embeddings? What are the pros and cons of using big values of parameter s in very sparse random projection methods?
b. Please consider the following scenario. You have a location summary data frame of a large number of users, denoted df01. Denote all users by set U. This data frame summarizes which site, at a specific time, each user appears at. The schema of the data frame is as follow.

```
root
|-- user_id: long
|-- date_time: string (format YYYY-MM-dd:hh-mm-ss)
|-- site_id: long
```

Denote the set of all site ids S. Let A denote large matrix of size |U|x|S|, where components A(u,s) denote the number of times user u appears at site s. Consider projecting A to a lower dimension |U|xd (d << |S|), where the resulting matrix B can be used as machine-generated location-related features for users in set U.

b1. Please explain why we might want to consider using very sparse random projections for this embedding. What parameter s you may want to use?

b2. Write pseudocode to transform df01 to df02 with schema as follow.

```
root
|--  user_id: long
|--  ft_01: float
|--  ft_02: float
...
|--  ft_20: float
```

where values df02(u,ft_i) (i = 1, 2,..., 20) are projected values of matrix A on space |U|x20, using very sparse random projections.

Question 3:

a. What are the fundamental differences between mean and median? Consider the mean and median of housing prices in District 4, Ho Chi Minh City. Among the two values (mean and median), which do you think would be higher and why?
b. Assume you are measuring the temperature of a laboratory room. There is a special thermometer installed in the room, which logs data to a database with hundreds of data points every second. Assume that you store the temperature data in a data frame df01, whose schema is as follow.

```
root
|-- date: string (format YYYY-MM-dd)
|-- time: string (format hh-mm-ss)
|-- temperature: integer (measured in Celsius)
```

Assume, also, that you know the range of all values in the temperature column a priori (because, for example, you know the thermometer can only withstand a certain range of temperatures.) Suppose the range is [-1000,1000].

The goal is to find the median temperature of the room each day. Suppose that for each date, there is a large number of data points that you can’t sort the temperatures. Your task [T] in this exercise is to transform df01 into df02 with the following schema

```
root
|-- date: string (format YYYY-MM-dd)
|-- median_temperature: integer (measured in Celsius)
```

Where df02 indicates the median temperature of the lab room each day.
b1. Write pseudocode to perform [T]
b2. Write pseudocode to perform [T] considering that the schema of df01 is as follow.

```
root
|-- date: string (format YYYY-MM-dd)
|-- time: string (format hh-mm-ss)
|-- temperature: float (measured in Celsius)
```

Question 4:
Please consider the following scenario. You have two data frames with the following schemas

df01

```
root
|-- usr_id: string
|-- feature_1: long
|-- feature_2: long
|-- feature_3: long
```

df02

```
root
|-- usr_id: string
|-- feature_4: long
|-- feature_5: long
```

Assuming there is no duplications on usr_id in the two data frame, i.e., df01.count() = df01.select(“usr_id”).distinct().count() and df02.count() = df02.select(“usr_id”).distinct().count(), and assuming that df01 and df02 both have N rows.
a/ What is the computational complexity of operation df01.join(df02,”usr_id”) and why?
b/ Write a program that executes the above join operation more efficiently

Question 5:
Please consider the following scenario. You have a data frame df01 with the following schema.

```
root
|-- usr_id: string
|-- app: string
|-- frequency: float
```

Each row of dataframe df01 indicates the percentage of time each user spends on a particular application. Suppose you want to get a matrix representation Users x Apps, where each entry indicates the frequency. In other words, you want to calculate df02 with the following schema.

```
root
|--  usr_id: string
|--  frequency_app_1: float
|--  frequency_app_2: float
…
|--  frequency_app_n: float
```

where n is the total number of distinct applications in df01, i.e.,
df01.select(“app”).distinct().count() = n.

How would you perform this task assuming:

a. Scenario [A01]: n < 1,000

b. Scenario [A02]: n > 10,000