INTERVIEW PRACTICE (DATA ANALYST)
UDACITY'S DATA ANALYST NANODEGREE
LEKHRAJ SHARMA, JAN 2017

Project: Answer the Following Interview Questions (6 total)

For each of the questions below, answer as if you were in an interview, explaining and justifying your answer with two to three paragraphs as you see fit. For coding answers, explain the relevant choices you made writing the code.

1. *Describe a data project you worked on recently.*

A data project which helped to improve my skills in data science was the final project on AB testing as a part of finishing Data Analyst Nanodegree from Udacity. In AB testing we test a change with two groups called "Control" (let us call A) and "Experiment" (let us call B) at the same time. We only offer change to experiment group and then measure the change with respect to control group (which do not get the change) to help us make a call whether this change is worth launching (i.e. change met business goals).

I had to analyze results of an experiment called "Free Trial Screener" and make a recommendation if organization (in this case Udacity like organization offering courses online) should go ahead with the change. Normally when user signs up for a free trial to a course, they are directly enrolled and automatically charged after the trial period. In this experiment, user was asked if they would be willing to devote 5 or more hours per week for the course. If they responded in affirmative, they would be enrolled as usual but for those users who may not be willing to commit to these minimum hours, a message would appear suggesting that the course usually requires larger time commitment and they might want to consider accessing course material for free. They still had the option to continue with enrolling if they like. Hypothesis was that setting up a clearer expectation upfront may mean less frustrated students down the line and improving capacity to support committed students. So possibly happier and better engaged customer!

Based on business goals, we clearly defined our metrics along with key evaluation metrics and the ones we do not expect to change (invariants). We also clearly defined expectation we had on these metrics to help us make the final call. We first estimated deviation in key evaluation metrics based on baseline data. This along with given practical significance level desired was used to figure out the sample size (our funnel size), duration (how long) and exposure (what fraction of traffic should be exposed to this change) for this experiment. Now it was time to look at the data from the experiment to look at changes observed between control and experiment groups. Used statistical techniques to figure out 95% confidence interval to help making a call if change being observed is statistically and practically significant. Also used sanity/sign tests to check if these calls match. Interesting part was that although the evaluation metrics significance levels matched the expectation at the surface but still did not recommend with going ahead with the launch because of uncertainty around one of the key metric (net conversion). Since our business could not afford to take risk about net conversion going down significantly. These kind of hard calls are what all data scientist need to make.

What I liked about this project is that this presented a good business situation where one gets to apply rigor of data science to make the right call. Being part of software industry I was generally aware of AB testing at top level but this project allowed me to go through all data science behind making these calls. It also helped me to improve the ways goals can be measured and outcome analyzed in a formal way.

2. *You are given a* ***ten piece*** *box of chocolate truffles. You know based on the label that six of the pieces have an orange cream filling and four of the pieces have a coconut filling. If you were to eat four pieces in a row, what is the probability that the* ***first two*** *pieces you eat have an orange cream filling and the* ***last two*** *have a coconut filling?*

Answer:

Let us denote "O" for picking orange cream filling chocolate, "C" for coconut filling chocolate, and P as probability for given event. So our problem becomes:

P(Picking "OOCC" in order)

= P(First as "O") * P(Second as "O") * P(Third as "C") * P(Fourth as "C")

P(First as "O") = 6/10 (since there are 6 orange ones out of total 10)

P(Second as "O") = 5/9 (since there are now only 5 orange ones out of remaining total 9)

P(Third as "C") = 4/8 (since we have 4 coconut ones out of remaining total 8)

P(Fourth as "C") = 3/7 (since there are now only 3 coconut ones out of remaining total 7)

Therefore,

P(Picking "OOCC" in order) = 6/10 * 5/9 * 4/8 * 3/7 = 1/14

*Follow-up question: If you were given an identical box of chocolates and again eat four pieces in a row, what is the probability that exactly **two** contain coconut filling?*

Answer:

Unlike the first part, here out of four chocolates we just need two orange and two coconut cream filling chocolates in any order, which means we have more possibilities to achieve our goal.

First Method

Let us denote "O" for picking orange cream filling chocolate, and "C" for coconut filling chocolate. There are 6 ways (4C2) of picking exactly two coconut filling chocolates out of 4. These are: OOCC, OCCO, CCOO, COCO, OCOC, COOC

So, our probability becomes (see answer to first part for details on how these individual probabilities are calculated):

P(Exactly two "CC")

= P("OOCC") + P("OCCO") + P("CCOO") + P("COCO") + P("OCOC") + P("COOC")

= 6/10*5/9*4/8*3/7 + 6/10*4/9*3/8*5/7 + 4/10*3/9*6/8*5/7 + 4/10*6/9*3/8*5/7

+ 6/10*4/9*5/8*3/7 + 4/10*6/9*5/8*3/7

= 1/14 + 1/14 + 1/14 + 1/14 + 1/14 + 1/14

= 6/14 = 3/7

Second Method (to verify our answer)

Total possible ways to pick 4 chocolates of 10 = 10C4 = (10*9*8*7) / (4*3*2) = 210

Ways to pick 2 orange chocolates out of 6 = 6C2 = (6*5) / 2 = 15

Ways to pick 2 coconut chocolates out of 4 = 4C2 = (4*3) / 2 = 6

Since we want to pick exactly 2 coconut chocolates out of 4, it means we need to also pick 2 orange chocolates as well. Therefore,

P(Picking exactly two coconut chocolates out of 4)

= P(Picking two coconut chocolates AND Picking two orange chocolates)

= (6 * 15)/ 210

= 3/7

3. *Given the table `users`:*

```
Table "users"

+-------------+-----------+

| Column      | Type      |

+-------------+-----------+

| id          | integer   |

| username    | character |

| email       | character |

| city        | character |

| state       | character |

| zip         | integer   |

| active      | boolean   |

+-------------+-----------+
```

Construct a query to find the top 5 states with the highest number of active users. Include the number for each state in the query result. Example result:

```
+------------+-----------------+

| state      | num_active_users |

+------------+-----------------+

| New Mexico | 502             |

| Alabama    | 495             |

| California | 300             |

| Maine      | 201             |

| Texas      | 189             |

+------------+-----------------+
```

Answer:

SQL query to list top 5 states with highest number of active user

We can use aggregation (count) query (group by state) over active users. We list our results (order by) in descending order of active users count and in case of same count, list them in alphabetical order of state names. Final result is limited to 5 (limit) to list only top five states with active users.

```
SELECT state, COUNT(*) AS num_active_users

FROM users

WHERE active IS TRUE

GROUP BY state

ORDER BY num_active_users DESC, state

LIMIT 5;
```

4. *Define a function first_unique that takes a string as input and returns the first non-repeated (unique) character in the input string. If there are no unique characters return None.* Note: Your code should be in Python.

Answer:

Initial version I had, just stored the count of characters in an ordered counter but it required a scan of the ordered counter to return first unique character with count of 1. So in the next version (given below), I just used two different sets to track duplicate and unique characters. Used ordered dictionary to act as ordered set (since sets usually do not have any order) for unique character case so that I can just return its first character without needing to scan it. Used sets over just plain list as operation cost to delete or check for existence is just O(1) in sets. Time complexity of this code is O(n), where "n" is length of input string since we just make one pass on whole string and each set operations (add/delete) take constant time O(1) . Space complexity is of order O(n) as at most we may need to store total of "n" characters in our sets (worst case is when all characters are unique).

```
def first_unique(string):

  # Your code here

  ‘’’

  Returns first unique character (non-repeating) across whole

  of input “string”.

  ‘’’

  from collections import OrderedDict


  dupSet = set()

  uniqueSet = OrderedDict()


  # Scan the given string in one pass, keeping track of duplicate

  # and unique characters set. Chose “ordered” set for unique

  # characters so that we can just return its first element.


  for c in string:

    if c not in dupSet:

      if c in uniqueSet:

        del uniqueSet[c]

        dupSet.add(c)

      else:

        uniqueSet[c] = None


  # Return first unique character or None if no unique character

  # is found.

  return next(iter(uniqueSet), None)
```

```
# Test code

> first_unique('aabbcdd123')

> c

> first_unique('a')

> a

> first_unique('112233')

> None
```

5. *What are underfitting and overfitting in the context of Machine Learning? How might you balance them?*

   Using machine learning we build a model based on seen data (specialization) to predict the outcome based on unseen data (generalization). Our model consists of chosen (or learned) set of features (based on our understanding of the problem) and learned weights on them by training our model on training dataset (seen data). Real test of model accuracy is when we use it on unseen data (tested via our test dataset). Art of building a good model is to find that right balance between building a model that learns from seen data but does well on unseen data as well. Both underfitting (where we do not learn much from seen data) and overfitting (where we tailor too much to seen data) lead to poor performance of our model in real world when it encounters unseen data.

   Underfitting and overfitting is quite similar to classical problem of tradefoff between bias and variance. In case of underfitting, we end up applying high bias towards generalization where our model (possibly oversimplified with too few features) pays little attention to training data, producing results similar to a pure algorithmic approach. An underfitted model is likely to give high errors on training dataset and may not give good results on test dataset as well. On the other hand, in case of overfitting we end up picking high variance of our training dataset causing our model to overfit (possibly made too specific with too many features) to training data so much that it does not generalize well (i.e. does poorly on our unseen data). An overfitted model will show much higher error on test dataset than training dataset.

   Finding that balance between these two extremes is the art of finding that sweet spot where we just make our model as simple as possible (but no simpler) that provides good fit. We need to pick optimal number of high quality features. We could use regularization technique (where we put penalty for extra features) in case of regression, reduce feature set using PCA, discard few branches of decision tree based model to reduce overfitting. Overfitting is a problem when our evaluation of model on training dataset is not similar to how it gets evaluated against unseen data. While theoretically we could iterate over both training dataset and test dataset to co-relate their performance and find that sweet spot where error on training set is small both on training and test dataset, it is not practical due to limitation on availability of the data as well as danger of leaking knowledge of test dataset to our learning model which renders it less effective for real world unseen data.

   A better approach to avoid overfitting is to introduce a validation phase where we use sampling techniques like k-fold cross validation to indicate when further training is not resulting in better generalization. In cross validation, we come up with a samples of training dataset that can be used to test the generalization performance (similar to when we run it against test dataset) and avoid the problem of overfitting. In k-fold cross-validation, we randomly partition original training into k equal sized subsamples. From these k subsamples, we keep a single subsample as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data. This cross-validation iteration of train and test is repeated k times. The averaged results from these iterations are better indicator of performance with our real test dataset (i.e. of our model in real world).

6. *If you were to start your data analyst position today, what would be your goals a year from now?*

   My goals a year from now would be to seen as a person whom organization can count on for making insightful calls based on data which move business forward significantly. I am particularly passionate about working on problems which make human life better, make our planet a better place to live. In the process of getting there, I would like to get good understanding of different data sources and their quality along with kind of questions different stakeholders in organization have. I would also work towards developing deeper understanding of business domain and its customers to better answer questions coming my way as well develop my own insights. Based on best practices in industry and my own learnings, I will develop a repertoire of building blocks to solve data science problems which work best with data sources and business context organization has. I would use this set to not only solve problems coming my way but also evangelize them across organization. I will work towards building credibility in organization by not only delivering on work assigned but also picking at least one project pro-actively. On learning front, I would like to deepen practical exposure to machine learning to solve real problems such as building virtual assistants, harnessing insights from big data and creating visual dashboard that not only present trends but can answer intelligent queries.