Opendoor Data Science Take Home Problem Set

The questions below are meant to give candidates a sense of the problems that we tackle at Opendoor. We expect solutions in the form of a readme + working code. The problem set should take around 3 hours to complete.

Part 1. Simple housing model

Your task is to implement a simple model to predict home prices for a small real estate transactions dataset.

Instructions

1. Download the sample dataset here. (deprecated)
2. Predict the close price as of list date given the other attributes.
3. Build separate models with and without ListPrice.
4. Feel free to join the dataset to any other data sources, so long as they are not leaky.

Questions

1. Describe your methodology. Why did you pick it?
2. How well would you do if you excluded the list price?
3. What is the performance of your model? What error metrics did you choose?
4. How would you improve your model?
5. How would you host your model in a production environment to predict values of homes in real-time?

Part 2. Simple *n*-grams

Generate the top 10 *3*-grams for the article http://en.wikipedia.org/wiki/N-gram

Part 3. Memoizer

Write a function that accepts a single-argument function *f*, and an integer *k*, and returns a function that behaves the same as *f* except it caches the last *k* distinct accessed results of *f*.

For instance, if *memoize* is the function we're after, and let *mem_f = memoize(f, 2)*, then

- *mem_f(arg1) → f(arg1)* is computed and cached
- *mem_f(arg1) → f(arg1)* is returned from cache
- *mem_f(arg2) → f(arg2)* is computed and cached
- *mem_f(arg3) → f(arg3)* is computed and cached, and *f(arg1)* is evicted

Additional questions:

- Can you describe the efficiency of the memoizer?

- How does your memoizer handle concurrent access?