Python and Machine Learning Development Skills Assessment

This test is being administered to identify your ability to learn new concepts on the fly from online documentation. It is assumed that you have the following basic skills:

- Familiarity with Python 3.5+
- Familiarity with pip and/or Conda (for installing python packages).
- Comfort level to install various python packages on your computer.
- Use of an Python IDE (we use Jetbrains PyCharm - PyCharm CE can be downloaded free of charge).
- Github account with public access

Task 1 (2-3 hours): Simple web server and file manipulation:

Setup a web server using python and Flask. The web server will do two simple things:

- Write a phrase to the file, as sent in an http request.
- Read all phrases from the file, and return them in the body of an http response.

Task Expectations:

1. Setup webserver to run on http://localhost:8080.
2. Create file called ‘storage.txt’ which holds a file that has one phrase per line.
3. Create two endpoints on the server that take http requests:
    a. /write: The write endpoint will accept POST requests and take a single JSON parameter, ‘phrase’ and write it at the end of the file. The post body would be of the form:
    {phrase: “<phrase>”}
    b. /read: The read endpoint will accept GET requests and return each phrase in the list as a single JSON object of the form:
    {phrases: [“<phrase1>”, “<phrase2>”, … “<phrase*n*>”]}

Bonus Points (2 hours):

- Install PostGres locally.
- Create a database called ‘phrase_holder’ with a table called ‘phrases’
- Create a table called ‘phrases’ with ‘id:integer’ and ‘phrase:text’ columns.
- Read and write to the database as well as the file, but have code to catch errors if database is not connected.

Submission:

- Submit link to github project for review.

Task 2 (2-3 hours): Evaluate a skip-thought model:

The skip-thought outlined in Kiros paper (https://arxiv.org/pdf/1506.06726.pdf) allows for phrase vectors to be used to identify phrases with semantic relatedness and paraphrase detection. The Tensorflow folks have implemented the underpinnings of his paper in Tensorflow (https://github.com/tensorflow/models/tree/master/research/skip_thoughts).

Task Expectations:

Go to the Tensorflow link and follow instructions:

- Download the pretrained book corpus model.
- Download the SICK and MSRP datasets.
- Run the SICK and MSRP.
- Save the results to a text file.

Bonus Points (8 hours):

- Download wikipedia corpus: https://en.wikipedia.org/wiki/Wikipedia:Database_download.
- Extract about 30MB of sentences.
- Train up a 400 dimension vector model.
- Run the SICK and MSRP evaluation tests.
- Save the results to a text file.

Submission:

- Submit link to github project for review.