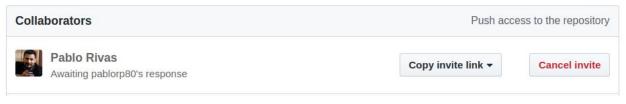
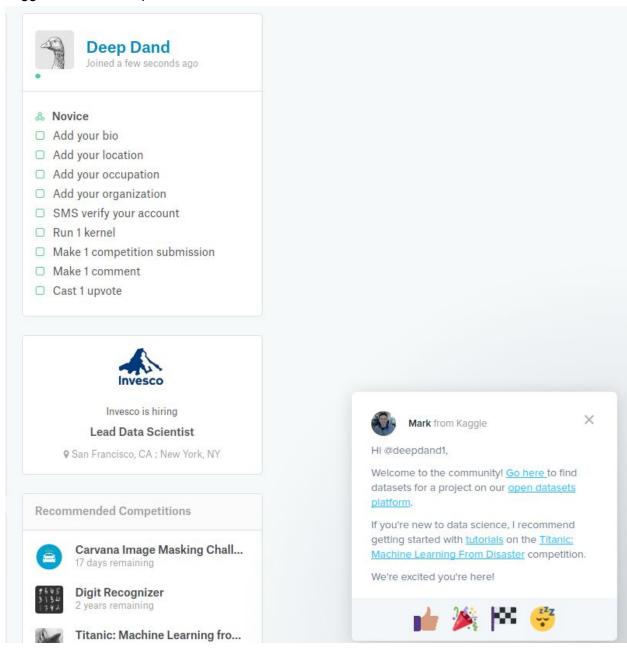
- 1. Single repo name Dand-00
- 2.
- a. Installation Proof

```
🔊 🖨 🗊 deep@acergpu940: ~
deep@acergpu940:~$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> import numpy
>>> import scipy
 >>> import sklearn
 >>> import matplotlib
>>> import pandas
>>> import tensorflow
I tensorflow/stream_executor/dso_loader.cc:111] successfully opened CUDA library libcublas.so locally I tensorflow/stream_executor/dso_loader.cc:111] successfully opened CUDA library libcudnn.so locally I tensorflow/stream_executor/dso_loader.cc:111] successfully opened CUDA library libcufft.so locally I tensorflow/stream_executor/dso_loader.cc:111] successfully opened CUDA library libcuda.so.1 locally I tensorflow/stream_executor/dso_loader.cc:111] successfully opened CUDA library libcurand.so locally
>>> print(sys.version)
2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609]
 >>> print(numpy.__version__)
1.12.0
>>> print(scipy.__version__)
0.17.0
  >>> print(sklearn.__version__)
0.18.1
 >>> print(matplotlib.__version__)
2.0.0
 >>> print(pandas.__version__)
0.19.2
 >>> print(tensorflow.__version__)
0.11.0rc2
```

b. Github class repo - DeepLearningF17.



c. Kaggle account - deepdand1



3. Problem 4.3

Answer a. - The multiplication is not possible since the dimensions doesn't meet matrix multiplication criteria. The criteria states that the number of columns of first matrix should be equal to number of rows of second matrix.

>>> import numpy as np

>>> import tensorflow as tf

>> a = [[1,4,-3],[2,-1,3]]

```
>>> b = [[-2,0,5],[0,1,4]]
>>> np.matmul(a,b)
ValueError: shapes (2,3) and (2,3) not aligned: 3 (dim 1) != 2 (dim 0)
Answer b.
>>> import numpy as np
>>> import tensorflow as tf
>> a = [[1,4,-3],[2,-1,3]]
>>> b = [[-2,0,5],[0,1,4]]
>>> at = np.transpose(a)
>>> at
array([[ 1, 2],
    [4, -1],
    [-3, 3]])
>>> result = np.matmul(at,b)
>>> result
>>> array([[-2, 2, 13],
    [-8, -1, 16],
    [6, 3, -3]])
>>> rank = np.ndim(result)
>>> rank
2
Answer c.
>>> a = [[1,4,-3],[2,-1,3]]
>> b = [[-2,0,5],[0,1,4]]
>>> c =[[1,0],[0,2]]
>>> bt = np.transpose(b)
>>> ci = np.linalg.inv(c)
>>> result = np.add(np.matmul(a,bt),ci)
>>> result
array([[-16., -8.],
```

[11. , 11.5]])